Barriers and Enablers for Traceability and Federation in Model-Based Digital Twin Engineering

Luca Berardinelli[a], Vittoriano Muttillo[b], Hugo Bruneliere[c], Romina Eramo[b], Alessio Bucaioni[d], Benoit Combemale[e], Sashko Ristov[f], Philipp Gritsch[f], Djamel Eddine Khelladi[g], Dao Khanh Duy[d], Rahmanu Hermawan[d], Andrey Sadovykh[h], Uğur Yayan[i]

[a] *Johannes Kepler University Linz*
luca.berardinelli@jku.at
[b] *University of Teramo*
vmuttillo@unite.it, reramo@unite.it
[c] *IMT Atlantique, LS2N (CNRS)*
hugo.bruneliere@imt-atlantique.fr
[d] *Mälardalen University*
alessio.bucaioni@mdu.se, khanh.duy.dao@mdu.se rahmanu.hermawan@mdu.se
[e] *University of Rennes*
benoit.combemale@irisa.fr
[f] *University of Innsbruck*
sashko.ristov@uibk.ac.at, philipp.gritsch@uibk.ac.at
[g] *CNRS, University of Rennes*
djamel-eddine.khelladi@irisa.fr
[h] *Softeam, France*
andrey.sadovykh@softeam.fr
[i] *Eskişehir Osmangazi University*
ugur.yayan@ogu.edu.tr

Abstract

Context: Digital Twins (DTs) have gained significant momentum over the past decade, reshaping industrial practices across various sectors, including manufacturing, construction, and infrastructure management. While this rapid adoption has driven innovation in cyber-physical integration, the scientific foundations and engineering principles of DTs are still being actively developed.
Objective: Although foundational contributions to model-based engineering (MBE) for digital twins exist, mainly in the form of conceptual frameworks, critical dimensions, such as traceability and federation, remain underexplored. This study aims to provide a comprehensive overview of the state of the art in model-based digital twin engineering, with a focused investigation of these two pivotal concerns.
Method: We conducted a multivocal literature review including 20 peer-reviewed publications and 56 gray literature sources, complemented by a survey of 51 expert stakeholders. The survey was administered during an open session of the MATISSE project plenary meeting, which included both consortium members and external participants from academia and industry. This mixed-method approach was designed to capture both academic and practitioner perspectives on the enablers and barriers to traceability and federation in DTs
Results: Our study reveals how traceability and federation are currently conceptualized, supported, and challenged within model-based DT engineering. We synthesize a landscape of relevant modeling languages, tools, application domains, and technical enablers, and identify key limitations and research gaps. Furthermore, we evaluate existing MBE solutions in light of industrial needs, distinguishing between areas of maturity and those requiring further innovation. The survey results should be interpreted in light of the sampling context, as the respondent pool mainly reflects researchers and early adopters.
Conclusion: This work offers evidence-informed guidance for researchers and practitioners by identifying concrete opportunities for improving traceability and federation in DTs. Our findings point to standardized modeling approaches, cross-domain interoperability, and AI-supported traceability mechanisms as promising directions warranting further investigation. Future research priorities include the development of protocols for federated DT systems and the adoption of a system-of-systems perspective to address complex industrial and societal challenges.

1. Introduction

In the contemporary technological landscape, digital transformation is progressively reshaping various industries. Within this context, Digital Twins (DTs) have emerged as a pivotal technology instrumental in enhancing digitization, sustainability, and resilience in industrial and societal applications. A DT is defined as a virtual representation of a physical system capable of real-time interaction and synchronization with its environment [1]. DTs are increasingly leveraged across multiple life-cycle stages, such as the design, development, analysis, simulation, and operation of complex cyber-physical systems [1].

DTs enable organizations to better understand and predict the behavior of physical systems, allowing for safer and more flexible verification, testing, and monitoring based on real-time data. This capability facilitates risk-free validation of critical infrastructures, such as satellites, trains, and grids. Gartner predicts that by 2027, over 40% of large companies will employ DTs [2].

Despite their transformative potential, DTs present significant challenges across the entire lifecycle, from design and development to deployment and maintenance. Key among these are issues of *traceability* and *federation*.

Traceability refers to the capability of representing, understanding, and analysing relationships between software artefacts. In model-based DT engineering (DTE), traceability consists of a logical relationship between models belonging to different DTs, which can be leveraged in digital thread scenarios [3, 4] These difficulties are further magnified when transitioning from standalone DTs to *federated DTs*.

In the context of DTE, federation refers to the integration and coordination of multiple DTs to create a unified system. This facilitates the sharing of data and resources across different DTs and their digital counterparts, enabling a more comprehensive and holistic representation of the system being simulated or controlled. According to Gartner [5], although 61% of surveyed companies currently deploy multiple DTs, achieving true federation remains elusive due to the high complexity of integration and coordination.

Recognizing these challenges, the Electronic Components and Systems (ECS) Strategic Research and Innovation Agenda (SRIA) outlines priorities to enhance digitization, sustainability, and resilience in industrial and societal applications. Among the thematic priorities identified by ECS SRIA, particular emphasis is placed on advancing federated DTs, developing robust traceability frameworks, and promoting standardized approaches for seamless integration across diverse industrial and societal domains.

Model-Based Engineering (MBE) [6] encompasses engineering methodologies leveraging machine-readable models as purposeful abstractions to deal with complex systems and their lifecycle. It encompasses key concepts such as models, metamodels, and model transformations, aiming for maximum automation, productivity, and system quality [7]. As such, MBE poses itself as a potential enabler for effective DTE processes. In particular, in MBE, traceability refers to the ability to link models, intended as cornerstone engineering artifacts, created during the system's lifecycle. This ensures that every element of the model can be traced back to its source, making it easier to understand, verify, and maintain the system. In their systematic mapping study, Batot et al. [8] explored this concept, emphasising trace links as direct connections between system artifacts, including MBE ones.

This paper provides a comprehensive overview of the state-of-the-art in model-based DTE, with a particular focus on traceability and federation. It clarifies the definitions of these concepts, identifies related industrial needs and existing barriers, and reviews current modeling languages, tools, and techniques addressing these challenges. Additionally, it systematically evaluates existing solutions against industrial requirements, highlighting areas that are ready for adoption and those that require further research.

Integrating MBE with DTE creates a powerful synergy. MBE utilizes abstraction and automation to effectively address the inherent complexities of DTs, streamlining their creation, management, and evolution. This study is guided by two key Research Questions (RQs):

- *RQ1:* What are the key enablers and current barriers to support traceability in the model-based engineering of DTs?
- *RQ2:* What are the key enablers and current barriers to support federation in the model-based engineering of DTs?

To address these research questions, we adopted a mixed-methods research design consisting of a Multivocal Literature Review (MLR), composed of a Systematic Literature Review (SLR) and a Gray Literature Review (GLR), and a survey. The SLR analyzed peer-reviewed scientific literature to establish the current state of the art. Recognizing the value of practitioner-focused sources, we conducted a GLR, capturing insights from blogs, white papers, and industry reports to bridge the gap between academia and practice. Additionally, the survey, involving expert stakeholders from academia, industry, and government, provided pragmatic perspectives, enabling the formulation of evidence-informed recommendations and guidance for practitioners and researchers. This guidance highlighted where existing MBE solutions can be effectively applied, thereby acting as an enabler for DTE, and identified barriers requiring further investigation.

Our study reveals how traceability and federation are currently conceptualized, supported, and challenged within model-based DTE.

Traceability in model-based DTE is primarily addressed in early design phases but rarely maintained throughout the full lifecycle. Emerging approaches such as blockchain, runtime models, and PLM-driven digital threads help enhance trace link maintenance. However, challenges remain, including the limited use of MBE techniques for dynamic traceability, lack of standardized metamodels, and weak end-to-end integration. Adoption is strongest in manufacturing, aerospace, transportation, and healthcare, where DTs support predictive maintenance, compliance, and optimization.

Federation in model-based digital twin engineering is supported through hierarchical architectures, semantic data models, and real-time synchronization, yet remains largely conceptual, with limited executable coordination across distributed DTs. While tools facilitate integration and runtime coordination, challenges persist, particularly the absence of formal orchestration standards and shared semantics, which hinder scalability. Adoption is strongest in manufacturing, aerospace, and smart cities, where federated DTs support integrated system-level collaboration. However, poor traceability, together with lack of standardized metamodels and lifecycle integration weakens federation, while organizational and security constraints further restrict model sharing and cross-domain federation.

The paper is structured as follows. Section 2 outlines the foundational concepts and surveys prior work pertinent to the DTE. Section 3 details our research method. Results concerning enablers and barriers for traceability and federation are presented in Section 4 and Section 5, respectively, with a focus on model-based techniques for DTE, tools, and application domains. Section 6 presents the survey results. Section 7 discusses the results collected by the MLR. Section 8 addresses validity threats. Finally, Section 9 concludes the study.

2. Background and Related Work

This section provides the necessary conceptual background and summarizes relevant literature on DTE. Section 2.1 introduces key concepts, including DTs, MBE, traceability, and federation. Section 2.2 reviews prior studies related to these topics, highlighting how our work builds upon and differentiates from existing research.

2.1. Background

Digital Twins (DTs) have evolved from isolated virtual replicas into federated systems-of-systems, where multiple autonomous DTs are integrated within a shared virtual environment to represent complex assets or processes [9]. Federation enables not only data exchange but also the compositionality of models and knowledge across organizational and technical boundaries. Applications span manufacturing, aerospace, and healthcare, where DTs of subsystems (e.g., machines, avionics, or patients) interact to support system-level optimization and decision-making.

To support such federation, several frameworks and standards have emerged. The Digital Twin Consortium promotes model-based interoperability and federated data repositories, while technical standards like the IEEE High-Level Architecture (HLA) and protocols such as OPC UA facilitate the integration of heterogeneous twin models. Industrial platforms like Azure Digital Twins further enable federated deployments through standardized twin definitions. Nonetheless, challenges persist in achieving semantic interoperability [10, 11] across domains, as heterogeneous data schemas and ontologies can hinder seamless integration.

A core promise of DT technology is the establishment of a *digital thread*, i.e., a continuous traceability framework that links requirements, designs, simulations, and operational data throughout a system's lifecycle. Traceability shall ensure that every design decision is connected to its corresponding artifacts and real-world performance data, thereby creating a feedback loop for validation and refinement. Additionally, blockchain-backed DT architectures have been proposed to guarantee data authenticity and lifecycle integrity through immutable, auditable logs of model changes and runtime states [12].

In this context, MBE provides a foundational approach to both traceability and federation for DTs. System models created via standard, general-purpose, or domain-specific languages (e.g., SysML[1], Azure DTDL[2], or Modelica[3] to mention a few) serve as a single source of truth, allowing system requirements, design elements, and verification results to be systematically linked. When DTs are developed atop these models, they can continuously monitor real-world performance and trace anomalies back to design assumptions [13].

2.2. Related work

To the best of our knowledge, this is the first mixed-method study to systematically examine the current state of model-based DTE, with a specific focus on traceability and federation concerns. The study combines insights from a survey involving 51 respondents with a comprehensive multi-vocal literature review comprising 20 peer-reviewed publications and 56 gray literature sources.

We identified only one tertiary study partially related to our topic: the work by Souza et al [14], which presents a high-level review of the Industry 4.0 (I4.0) landscape based on 46 systematic reviews indexed in Scopus and Web of Science. Their content analysis categorized the literature into three main areas: conceptual foundations, enabling technologies, and operations management. While their study highlights the multidisciplinary nature of I4.0 and the relative lack of real-world industrial applications, it does not specifically address DTE, nor does it explore federation or traceability—critical concerns that our work investigates in depth.

Nevertheless, several secondary studies, surveys, and related works have addressed aspects of DTE. For instance, Barricelli et al. [15] conducted a survey to examine the definition, characteristics, applications, and design of DTs. One of their key findings emphasized the importance of

[1] OMG SysML: https://www.omgsysml.org/
[2] Digital Twins Definition Language (DTDL): https://azure.github.io/opendigitaltwins-dtdl/DTDL/v3/DTDL.v3.html
[3] Modelica Language: https://modelica.org/language/

DTs being "tied" and "linked." Our work complements and extends these insights by focusing specifically on federation and traceability within the context of interoperability in model-based DTE. Another study that addresses traceability in DTs is presented by Suhail et al. [16], who investigate blockchain-based DTs. One of the key benefits highlighted in their work is the enhanced data traceability enabled by blockchain technology, which allows deviations between DT entities to be identified efficiently through end-to-end access to the full transaction history. Our study complements their findings by incorporating federation aspects, thus providing a broader perspective on traceability in interconnected and interoperable DT environments.

Ferko et al. [17] conducted a literature study investigating the software architectures of DTs. Among their findings, compatibility emerged as one of the most frequently discussed quality attributes, an aspect closely tied to the realization of DT federation. Additionally, they observed that service-oriented architectural patterns are the most commonly adopted in DT systems, which inherently support interoperability and federation. Our work complements Ferko's by further exploring federation from a MBE perspective, with a specific focus on traceability and industrial needs.

Perno et al. [18] identified integration challenges as one of the key barriers to DT implementation in the process industry. However, their study is domain-specific and limited by the available literature in that sector, which constrains its ability to capture the broader spectrum of enablers and barriers across industries. In contrast, our work reviews a wider and more diverse body of literature, providing a more comprehensive perspective on DT implementation challenges, particularly in relation to federation and traceability.

Minerva et al. [19] identified composability as a key characteristic of DTs in the context of the Internet of Things (IoT). They argue that, with the support of existing middleware platforms, composability enables the formation of system-of-systems through large-scale DT integration. Our work extends and complements their findings by emphasizing how composability contributes to interoperability in model-based DTE, particularly through the lens of federation and traceability.

In addition to the above works, a growing body of research has explored the use of MBE in the development and management of DTs. Notable examples include the study from Michael et al. [20] and Lehner et al. [21]. The former presents a conceptual study on the role of Model-Driven Engineering (MDE) [6] in supporting the development and maintenance of DTs across their lifecycle. Their work emphasizes the importance of modeling both the physical system (via simulation, systems engineering, and data processing models) and the DT as a complex software system. They argue that MDE techniques, such as model reuse, runtime models, and code generation, can significantly enhance the efficiency, adaptability, and traceability of DT implementations. Their contribution includes a road map outlining future research directions for integrating MDE into DTE. While their work offers a high-level vision and conceptual road-map, it does not empirically assess industrial needs or provide a synthesis of current practices. In contrast, our study complements this vision by combining a multi-vocal literature review with an industry survey, providing grounded insights into how federation and traceability are currently addressed in practice.

The latter reported on a systematic mapping study to investigate the application of MDE automation techniques, such as model-to-model transformation, code generation, and model interpretation, to DT systems across domains. Analyzing 66 primary studies out of 189 initially retrieved, their work categorizes how these techniques are applied in various domains and DT types. The study found that while MDE is gaining traction in several sectors, applications are currently dominated by the manufacturing and transportation sectors. The authors highlight a lack of consolidated knowledge about the relationships between DT characteristics, domains, and the effectiveness of MDE techniques, and they propose a taxonomy and research road-map to guide future efforts. While their work offers a technical classification and valuable research road-map, it does not explicitly address traceability or federation, which are central to our investigation. Furthermore, their focus is limited to peer-reviewed sources. In contrast, our mixed-method approach incorporates academic and gray literature alongside empirical insights from industry, offering a broader and more applied perspective on model-based DTE.

3. Research Method

This section presents the methodology adopted in this study, which comprises an MLR, including 20 peer-reviewed publications and 56 gray literature sources, and a survey with 51 respondents. The overall research methodology is illustrated in Figure 1. It starts with defining the research goals (RGs) and corresponding research questions (RQs).

The MLR activities were organized into two parallel flows: one addressing peer-reviewed scientific literature and the other focusing on gray literature, i.e., materials produced outside traditional academic publishing channels.

The search activity involved an automated search across major research databases, including Scopus, ACM Digital Library, IEEE Xplore Digital Library, and Web of Science, to ensure comprehensive coverage of the scientific literature. In parallel, Google Search, the most widely used search engine globally, was employed to retrieve relevant gray literature sources.

Subsequently, the selection and filtering activities were conducted by applying the Inclusion and Exclusion Criteria (IC/EC) defined during the planning phase to both

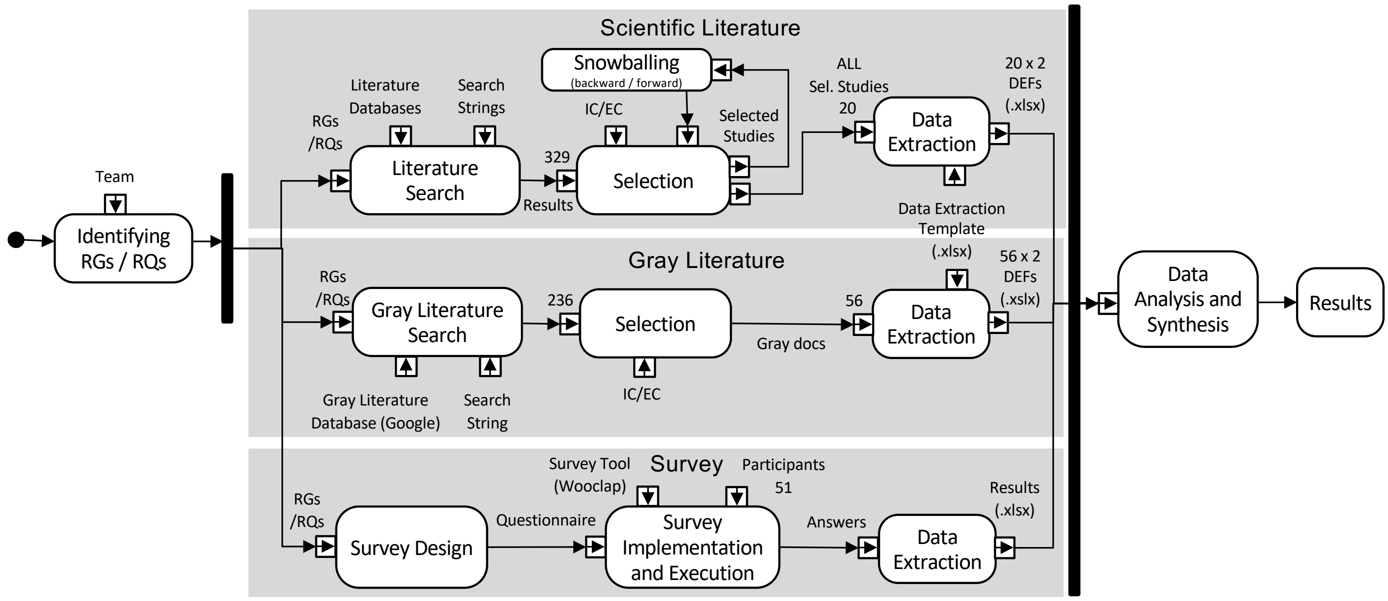

Figure 1: The research methodology and detailed activities. Legend: RG = Research Goal, RQ Research Questions, IC = Inclusion Criteria, EC = Exclusion Criteria, DEF = Data Extraction Form.

scientific and gray literature datasets. In addition, the filtered set of scientific studies underwent a closed-recursive, semi-automated forward and backward snowballing process using Google Scholar, following the guidelines proposed by Wohlin et al.[22]. During the data extraction and synthesis phase, a data extraction form (DEF) was developed to classify and compare the selected studies. Two reviewers independently completed DEF instances for each selected study, and the results were subsequently aggregated to support analysis.

As a third parallel activity, the survey encompassed the design, implementation, and execution of a questionnaire targeting expert stakeholder to complement the findings from the MLR. The questionnaire was developed, deployed, and administered to an audience of 51 participants using the Wooclap survey tool for response collection.

Finally, to systematically address the defined research questions, the data analysis phase entailed both quantitative and qualitative analyses of the aggregated data.

To enable the independent replication and verification of this study, we provide shared online material, including the data obtained from the search and selection procedure, along with the complete list of results, the data extraction forms, and the script used for data analysis and synthesis [23].

The following sections explore in detail the phases by discussing their steps.

3.1. Research Goal and Questions

We defined the Research Goal (RG) using the Goal-Question-Metric perspectives [24], as illustrated in Table 1. We decomposed the RG into the two RQs presented in the introduction, each addressing distinct aspects of the overall RG. By addressing RQ1, we identify and evaluate how

Table 1: Research goal

Purpose	Identify, classify, and evaluate
Issue	publication trends, founding concepts, benefits, supporting tools, and application domains
Object	of model-based digital twin engineering for socio-technical systems with a focus on traceability and federation
Viewpoint	from the point of view of researchers and practitioners.

traceability is considered in the creation and management of digital models, contributing to a deeper understanding of the benefits and supporting tools available for ensuring traceability in DT MBE. RQ2 focuses on identifying and evaluating the role of federation, supported by traceability among digital artifacts, as a key enabler in managing complex systems composed of multiple digital twins. By answering this question, we gain insights into the fundamental concepts and supporting tools employed for achieving federation in model-based DTE.

3.2. Search and Selection

For the peer-reviewed scientific literature search, the research databases were selected following the guidelines and recommendations provided by Kitchenham et al. [25] and Wohlin et al. [22]. Based on these guidelines, four primary databases were consulted to ensure comprehensive and high-quality coverage of the literature.

- **Scopus** (`http://www.scopus.com/`), a multidisciplinary database aggregating abstracts and citations from major publishers (ACM, IEEE, Springer, Elsevier).

- **ACM Digital Library** (http://dl.acm.org/), included to ensure complete coverage of Computer Science publications beyond Scopus indexing.
- **IEEE Xplore** (http://ieeexplore.ieee.org), consulted separately to verify search string effectiveness and enhance technical content coverage.
- **Web of Science** (https://www.webofscience.com/wos/) to further broaden the search scope, spanning science, social sciences, and humanities.

The search string was constructed based on the overarching research goal and associated research questions. It was designed to be inclusive yet straightforward, aiming to maximize the retrieval of relevant studies while preserving the specific focus of this review on Digital Twins and Model-Based Engineering. For this reason, we centered the query on the canonical DT and model-based terms rather than expanding it with a broader set of adjacent industrial, lifecycle, or simulation-related expressions. Such terms may indeed appear in relevant studies, but in isolation they can also refer to broader streams of work that fall outside the intended scope of model-based DTE. Potential threats to validity associated with this search strategy are discussed in the following sections. Given the extensive volume of primary studies on DT, we deliberately incorporated terms into the search string to restrict the peer-reviewed literature search to secondary studies, in line with the focus of this work. It should be noted that we did not include specific terms such as traceability and federation in the search string, as preliminary tests showed that more specific queries yielded very few or no relevant results. Therefore, we opted for a broader query to ensure sufficient coverage and later filtered results based on their relevance to these topics.

Table 2: Search strings

Keyword	Search String
Digital Twin	Digital Twin*, DT*
Modelling	model-based, model-driven
Secondary Studies	systematic literature review*, literature review*, systematic review*, systematic mapping stud*, mapping stud*, systematic review*, systematic stud*, SLR, SMS, survey*

Table 2 summarizes the sets of terms composing the search string, which was formulated as a conjunctive logical proposition of three groups of search terms, each internally combined through a disjunctive logical proposition. The final search string is displayed below.

("Digital Twin" OR "DT*") AND ("model-based*" OR "model-driven*") AND ("systematic literature review*" OR "literature review*" OR "systematic review*" OR "systematic mapping stud*" OR "mapping stud*" OR "systematic review*" OR "systematic stud*" OR SLR OR SMS OR survey*)*

For the gray literature search, we employed the Google search engine, which accounts for approximately 92.2% of global web searches, to ensure comprehensive coverage of practitioner-oriented sources. A dedicated search string was formulated for Google Search, as shown below. This search string was designed to retrieve results containing the exact phrase "Digital Twin" along with either "model-based" or "model-driven" and either "federation" or "traceability". The complete Google Search string available in the repository [23] was restricted to results published from 2018 onwards and excluded content from major academic publishers and scholarly databases, thereby avoiding duplicates and unintended inclusion of academic papers[4].

"Digital Twin" AND ("model-based" OR "model-driven") AND ("traceability" OR "federation") after:2018

Both the peer-reviewed and gray literature searches were conducted in October 2024.

By systematically following the steps outlined in Figure 2a, we identified a total of 20 relevant peer-reviewed studies for inclusion in our investigation. The initial list, comprising 373 papers, was reduced to 330 after the removal of duplicates. We then applied the selection process, based on predefined Inclusion and Exclusion Criteria (IC/EC), to these 330 papers, identifying 12 papers as potentially relevant to our study. Table 3a presents the IC and EC used for selecting peer-reviewed. The IC encompass studies related to model-based DTE, specifically focusing on the federation and traceability of DTs. Conversely, the EC eliminate non-peer-reviewed papers, papers shorter than four pages, non-English or inaccessible papers, primary studies, and studies unrelated to MBE.

Following a fully recursive forward and backward snowballing process, the list was expanded to 24 candidate papers. Finally, two reviewers independently conducted a full-text reading of all 24 papers, resulting in a final selection of 20 relevant peer-reviewed studies considered in our investigation.

Figure 2b illustrates the search and selection process for the gray literature. The initial list comprised 225 links. Based on predefined IC/EC, we then applied the selection process to these 225 links, removed three duplicates, and obtained 56 gray literature documents as potentially relevant to our study. Table 3b presents the IC/EC applied for selecting gray literature from Google Search. The IC encompass links addressing DTE, with a specific focus on the federation and traceability of DTs, as well as tools explicitly supporting the engineering of DTs. Conversely, the EC remove links to peer-reviewed papers or pre-prints

[4] By appending "-site:<sitename>" to the string.

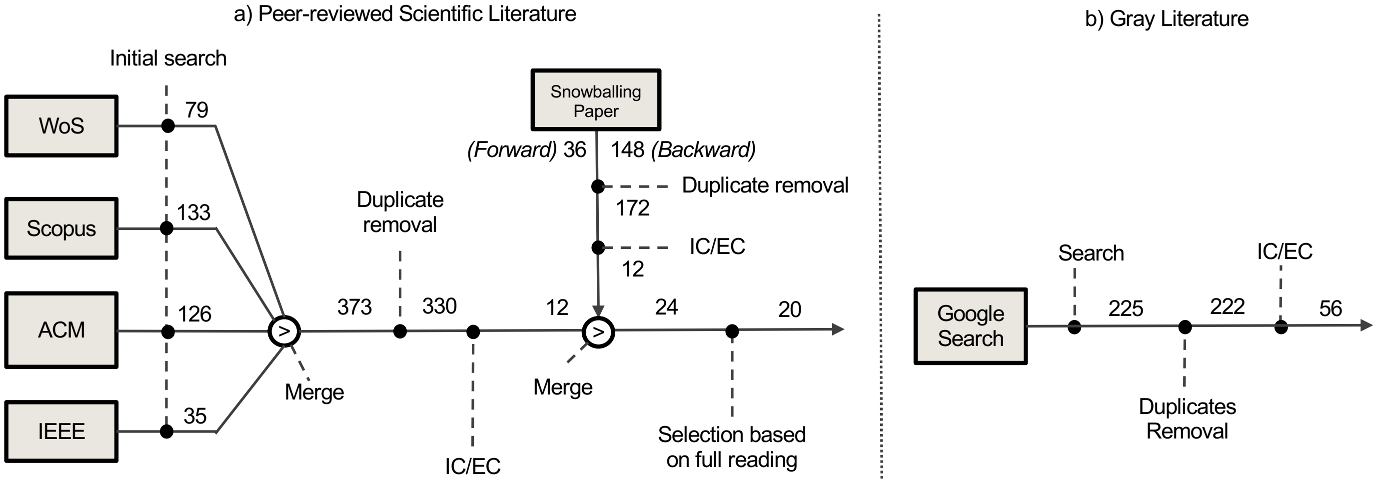

Figure 2: Overview of the search and selection process for the peer-reviewed (a) and gray literature (b).

Table 3: Inclusion and exclusion criteria for peer-reviewed literature

(a) Inclusion and exclusion criteria for peer-reviewed literature

Criteria	Assessment Criteria
Inclusion	We include studies addressing DTE, including any engineering activity.
	We include studies explicitly addressing the federation of DT.
	We include studies explicitly addressing the traceability of DT.
	We include studies published after 2019.
Exclusion	We exclude non-peer-reviewed papers.
	We exclude papers shorter than four pages as they provide insufficient information.
	We exclude papers that are not in English or accessible.
	We exclude any primary study.
	We exclude non-MBE studies.

(b) Inclusion and exclusion criteria for gray literature

Criteria	Assessment Criteria
Inclusion	We include links addressing DTE, including any engineering activity.
	We include links explicitly addressing the federation of DT.
	We include links explicitly addressing the traceability of DT.
	We include tools that explicitly address the engineering of DTs
Exclusion	We exclude links to peer-reviewed papers or preprints that point to literature/publications.
	We exclude links older than 5 years (2019)
	We exclude links not in English.
	We exclude non-MBE studies.

that point to academic publications, links older than five years (before 2019), non-English links, and studies unrelated to MBE. Compared to the criteria adopted for scientific literature, the IC for gray literature are broader, encompassing non-academic research, practical activities (including commercial initiatives), and tools directly related to DTE practices. In contrast, the EC are more stringent concerning publication age, ensuring that only current and accessible resources are considered. Furthermore, the explicit exclusion of peer-reviewed papers guarantees that the focus remains on integrating diverse practitioner-oriented perspectives from gray literature sources, while avoiding overlaps with the peer-reviewed corpus. Subsequently, the research team selected and fully inspected all 56 documents.

In cases of conflicting evaluations in both the peer-reviewed and gray literature, a third team member was consulted to resolve the disagreement and make a final decision.

The final list of selected peer-reviewed papers (SP) is given in the Selected Peer-reviewed Literature section. Similarly, the list of selected gray literature (GL) results is listed in the Selected Gray Literature section.

To further strengthen the treatment of gray literature, we explicitly appraised the selected sources at the source-type level in line with established MLR guidelines for software engineering. More specifically, we classified the included gray sources according to their outlet control and credibility [26]. The final gray corpus was treated as second-tier gray literature, that is, sources with moderate outlet control and moderate credibility, such as industrial reports, presentations, videos, practitioner-oriented web material, and similar non-peer-reviewed artifacts. Since the included gray sources fell within the same credibility tier, we reported this appraisal at the method level rather than discussing each item separately. We acknowledge, however, that even within the second tier, individual sources vary in their degree of credibility — for instance, a structured industry study (e.g., VDE) differs from a practitioner blog or a conference video presentation. This intra-tier variability was considered qualitatively during synthesis.

3.3. Data Extraction Form

To address the RQs, the team extracted various types of data. Each selected peer-reviewed and gray study underwent a comprehensive review process requiring a full reading. A DEF template was made available to all reviewers, accompanied by detailed guidelines that indicated whether and how a given study contributed to answering the stated RQs. To ensure the reliability of the extracted information, two different reviewers independently reviewed each document.

The DEF template, organized as a spreadsheet, is available in our online repository [23]. The design of the DEF template followed several key rationales. First, the form consists of closed-form questions (Yes/No), answered via check-boxes to facilitate the aggregation and analysis of the collected data, while also allowing limited optional textual comments to clarify specific judgments when necessary. Second, two principal facets were considered: DTE and MBE.

The facet on DTE captures information across key engineering phases, including DT modeling, simulation, data integration, real-time monitoring, predictive analytics, optimization, interoperability, scalability, security and privacy, lifecycle management, user interaction, federation, traceability, and any other aspects cited by the reviewed documents. If a primary study reported on DT modeling, reviewers were instructed to collect additional information regarding potential MBE techniques or practices mentioned. Additionally, these facets supported the collection of information regarding federation and traceability aspects, including potential interrelations among modeling, federation, and traceability activities.

The model-based facet was structured based on the MBE Body of Knowledge proposed by Burgueño et al. [7], which identifies and organizes key model-based topics according to their importance and coverage in existing curricula.

The DEF also captured information on the application domains of DT technologies, including agriculture, aerospace, automotive, construction, education, energy, food production, healthcare, logistics, marine engineering, manufacturing, retail, robotics, smart cities, telecommunications, and other sectors.

3.4. Survey

We conducted the survey following the process proposed by Kasunic [27] and Wohlin et al. [28]. Aligned with our RQs, the survey was designed to complement and contextualize the findings emerging from the MLR. More specifically, the questionnaire design was informed by the main themes, concepts, and gaps identified during the MLR, which helped us derive the survey dimensions and formulate questions on DTE practices, traceability, federation, application domains, standards and tools, AI usage, and MBE expertise. We first identified the target audience and analyzed its characteristics to ensure that the questionnaire used terminology and examples appropriate for both researchers and practitioners. The intended target group included: (i) academics with expertise in MBE for DTs and (ii) industry professionals from organizations that have adopted or are in the process of adopting DTs.

To reach this audience, we leveraged the plenary meeting of the MBE of Digital Twins for early verification and validation of Industrial Systems (MATISSE) project. MATISSE is a European project under the KDT Joint Undertaking, involving more than 30 organizations from seven countries. We distributed the survey during an open session of a project plenary meeting, which lasted two hours and was attended by project partners as well as external researchers and practitioners joining either in person or via Microsoft Teams. This setting gave us direct access to a relevant expert audience, but it also implies that the respondent pool primarily reflects researchers and early adopters already interested in DT and MBE topics.

Based on this target group and the MLR findings, we designed the questionnaire by defining: (i) the key questions to ask, (ii) the question types (e.g., closed-ended, Likert-scale, and open-ended), (iii) the sequence of questions, and (iv) the overall structure and layout of the instrument. Before distributing the questionnaire to the full audience, we administered it to a sample group of respondents to validate its clarity, structure, and relevance. The feedback collected during this preliminary step helped us refine the wording of selected questions and improve the overall consistency of the instrument. The questionnaire was then organized into blocks covering general information, DTE concepts and challenges, application domains, standards and tools, AI, research references, MBE expertise, and perceptions on DTE and MBE. The authors were present to provide context, clarification, and support during the session.

The session began with a brief presentation of the survey's objectives, followed by the collection of basic participant demographics, such as industry affiliation and years of experience. The survey consisted of a mix of closed-ended questions and open-text fields, with each question accompanied by a short explanation and practical examples to ensure clarity and shared understanding. When participants were asked to evaluate specific items, a 5-point Likert scale was used. Responses were collected using Wooclap [29]. Additionally, participants were encouraged to provide free-text comments for each item, allowing us to gather qualitative insights alongside the quantitative data. The session concluded with an open discussion, during which attendees reflected on potential improvements to the checklist and shared general feedback about the survey process. In total, we collected 51 unique responses.

The results and analysis of the survey are presented and discussed in Section 6. The complete questionnaire survey instrument is reported in Appendix A and is also released through the online replication package [23] as a stable supplementary artifact. This material includes the final question set, the anonymous response export, and the aggregated workshop results. The instrument followed a fixed section structure with only limited conditional elaboration prompts (e.g., "If yes, please specify"), rather than complex branching logic. To improve construct interpretability, Appendix A also includes a mapping between the main analytical constructs used in this study and the corresponding survey items and response formats in Table A.7.

3.5. Mixed-method integration strategy

Our study adopted a triangulation-oriented mixed-method integration strategy [30]. The MLR and the survey addressed the same research questions from complementary evidence sources. The survey instrument was connected to the MLR at design time, since its dimensions and wording were derived from the themes emerging from the literature review. The two strands were then analyzed separately and merged at the interpretation and reporting level through a joint evidence trace.

For each major enabler and barrier, we compared: (i) peer-reviewed evidence from the MLR, (ii) gray literature evidence, and (iii) survey evidence, combining Likert-scale responses and open-text comments. This integration was used to identify convergence, complementarity, and divergence across evidence sources, rather than to treat the survey as a purely confirmatory instrument.

3.6. Data Analysis and Synthesis

We extracted, analyzed, and synthesized the data following the guidelines proposed by Cruzes et al. [31]. Our study incorporated both quantitative and qualitative analyses, combining content analysis [32] and narrative synthesis [33]. This mixed-methods approach enabled the systematic categorization and coding of findings: content analysis provided a structured classification framework, while narrative synthesis offered detailed explanations and interpretations, thereby enriching the overall understanding of the results.

To identify trends and derive insights for each cluster in the DEF, we employed a line-of-argument synthesis process [28]. Initially, each primary study was analyzed individually to classify its main characteristics. Subsequently, the entire dataset was examined collectively to uncover broader patterns and interrelationships. The extracted data were then grouped, cross-tabulated, and systematically compared across the different facets captured in the DEF. To identify relationships between these facets, contingency tables were employed, facilitating the extraction and evaluation of pairwise relationships. This multifaceted approach ensured a comprehensive and rigorous synthesis and interpretation of our findings.

In the synthesis, we did not treat peer-reviewed and gray literature as equivalent forms of evidence. Consistent with the adopted MLR guidelines [26], we assigned greater analytical weight to the peer-reviewed literature and used the gray literature as complementary evidence to capture industrial practice and practitioner-oriented perspectives. More specifically, we used a weighting factor of 1 for peer-reviewed literature and 0.5 for the gray literature corpus [26], reflecting its classification as second-tier gray literature. Accordingly, stronger conclusions were primarily anchored in the peer-reviewed studies, while gray literature was used to complement, contextualize, and illustrate practice-oriented trends and concerns.

4. Enablers and Barriers for Traceability (RQ1)

This section presents the results derived from the comprehensive application of the research methodology concerning RQ1, which investigates how traceability is conceptualized and implemented in model-based DTE. We identify key enabling factors and supporting tools, as reported in Table 4, including techniques, technologies, and application domains. Additionally, we report on the main barriers and challenges that hinder effective traceability across the digital twin lifecycle.

4.1. Model-based Digital Twin Engineering Techniques

Jones et al. [SP01] emphasize the importance of design traceability in the early stages of product development, particularly for capturing design rationale and managing data effectively across various lifecycle phases, including disposal. Despite recognizing these critical aspects, the paper notes a gap in the existing literature regarding the comprehensive addressing of traceability across the entire product lifecycle. In this context, Jones et al. [SP01] references only one primary work by Heber et al. [34], which proposes using blockchain technology to support the traceability of MBE artifacts. This approach aims to enhance traceability, ensure data integrity, and promote seamless interoperability across all phases of the product lifecycle.

Rathore et al. [SP05] discuss traceability more directly, especially in the context of maintaining accurate digital representations of physical assets. They emphasize the significance of monitoring and updating DTs to reflect real-world changes, citing Zhang et al. [35], who assert that *"traceability frameworks enable the tracking and updating of DT data"*. This highlights the functional role of traceability in ensuring the reliability and consistency of DT systems over time.

Brauner et al. [SP08] refer to traceability within the framework of the Internet of Production (IoP), where the need for provenance information is paramount. The authors describe how provenance enables, for example, the identification of data sources used to train neural networks that underpin digital shadows, as well as the usage history of mathematical models employed in solving complex production tasks. As they note: *"With the latter, it is possible to get, e.g., the origin of data used for training an artificial neural network representing a particular Digital Shadow or the usage history of mathematical models comprised within the Digital Shadow to solve difficult production steps"*. This illustrates how traceability and provenance, when combined, are vital for transparency, accountability, and reproducibility in DT environments.

[SG07] describes traceability in terms of the ability to link simulations directly with requirement verification, reusable models, and automated design maps. This capability supports more efficient validation processes and reinforces the consistency of DTE practices.

The blog post in [SG03] explores the concept of the Digital Thread within the broader context of digital artifacts generated by a Model-Based Enterprise, which leverages digital models as the authoritative source of all product-related information throughout their lifecycle, allowing for end-to-end tracking of product states and revisions. Here, traceability is emphasized as a critical enabler, connecting models and data across the entire product lifecycle. The post clarifies that traceability is not limited to data linkage but also involves ensuring that these connections are context-sensitive, meaning they deliver the right information to the right users at the right time. The integration of DTs, IoT, and AI technologies is presented as a foundation for achieving this comprehensive traceability, which in turn enhances decision-making, cross-functional collaboration, and enterprise-wide optimization.

The presentation in [SG24] defines traceability as the capability to connect the various phases of Product Lifecycle Management (PLM), a major advantage attributed to the implementation of DTs. By linking stages such as design, testing, and production, traceability ensures continuity and coherence in the engineering process.

In [SG26], the digitalization of manufacturing processes is highlighted as a means to improve traceability, enabling organizations to track individual products and justify design or operational decisions. The source also emphasizes the importance of utilizing established standards to unlock the full potential of digital transformation initiatives.

According to [SG35], implementing automated traceability workflows significantly benefits change management, compliance validation, and regulatory reporting. These workflows help ensure that processes remain aligned with industry-specific standards and reduce the overhead associated with manual data tracking.

Finally, the role of automated traceability is also discussed in [SG37], where workflow updates are used to validate design modifications. This helps maintain consistency across models and ensures that engineering decisions align with overarching strategic objectives.

4.2. Tools

Rathore et al. [SP05] include Siemens Insights Hub (formerly MindSphere), which support PLM-like traceability across manufacturing processes. Tools such as ANSYS Twin Builder enable linkage between design models and simulations, which can be leveraged in traceability frameworks.

Wortmann et al. [SP06] highlight modeling languages like UML, SysML, and AutomationML. Although they don't explicitly reference tools, these languages support traceability through model transformations and metamodel conformance, a core practice in MBE [7].

Siriweera et al. [SP11] mention OPC-UA, a middleware standard for data integration, which is supported by Papyrus for Manufacturing, a tool with MBE capabilities that enables tracing system states and model consistency across lifecycles.

Merkle et al. [SP12] mention AutomationML, which in turn is based on the CAEX standard, and MTConnect for monitoring manufacturing systems. The AWS ecosystem (S3, DynamoDB, Lambda, Fargate) facilitates data provenance and traceability at runtime by storing and processing DT data streams.

Guinea et al. [SP14] introduce Azure DTs and their DT Definition Language (DTDL), both of which enable structured representations of twin states and transitions, as key enablers for automated traceability.

ARAS's PLM platform [SG03] fosters traceability by connecting DT models and lifecycle data, enabling decision-making and cross-functional collaboration. It is specifically tailored for Model-Based Enterprises.

Papyrus for Manufacturing [SG25], also mentioned by [SP11], contributes to traceability through model-based representations of assets using standard modeling languages like UML, SysML, and Modelica. The inclusion of components like the AAS Designer and OPC UA Designer enhances its capacity to structure and maintain traceable links across assets and lifecycle phases based on the AAS and OPC UA standards, respectively.

PROSTEP's OpenCLM solution [SG35] is explicitly focused on establishing traceable, intelligent links among distributed data sources. It integrates PLM, ERP, and MES systems, ensuring consistent access to authoritative product data while maintaining a traceable digital thread.

Northrop Grumman [SG37] employs ModelCenter to integrate SysML with analytical models (e.g., FEKO, MATLAB, HFSS, CST), offering virtual verification and automated traceability across the design and simulation stages.

Visure's ALM Platform [SG41] directly supports end-to-end traceability from requirements through to testing and deployment. It is especially relevant for regulated domains, as it integrates AI-assisted modeling and customizable traceability frameworks.

TECHNIA's ecosystem [SG14], based on Dassault Systèmes' 3DEXPERIENCE and tools like CATIA, Dymola, ENOVIA, and DELMIA InUse, maintains traceability across modeling, simulation, and operational monitoring layers, thereby linking static and dynamic aspects of product behavior. Therefore, the TECHNIA Magic System of Systems Architect tool is described as a key enabler for tracing requirements and dependencies throughout complex systems. This tool ensures data consistency and traceability across federated DT models, facilitating integrated development and lifecycle management.

Finally, Boeing's MBE approach [SG56] focuses on lifecycle traceability by integrating definitions and production planning via SysML and PLM solutions.

4.3. Domains

DT technologies are applied across a wide range of domains, with growing attention to traceability and ensuring consistent, lifecycle-driven system development.

The manufacturing domain emerges as a leading example, as highlighted by Wortmann et al. [SP06], where model-based languages such as SysML and AutomationML [36] are employed to establish clear traceability between requirements, functional models, and system execution. Numerous studies (e.g., Back et al. [37], Díaz-Madroñero et al. [38], Francalanza et al. [39]) demonstrate how AI-enhanced DTs support fault localization, predictive analytics, and automated V&V, ensuring that system behavior remains traceable to its design intentions. The use of real-time feedback loops, as shown by Chen et al. [40] and Lu [41], further reinforces runtime traceability between physical and digital entities. The manufacturing sector [SG35, SG14, SG47] also benefits from DT traceability to document verification processes and ensure regulatory compliance. PROSTEP's OpenCLM is used to record decision histories and manage data integrity across MES, ERP, and simulation environments.

In the transportation sector, Werbińska-Wojciechowska et al. [SP20] detail several application domains where traceability is critical. In air transportation, DTs enable traceable predictive maintenance and condition monitoring, ensuring that the operational status of aircraft and infrastructure is continuously documented. In railway transportation, traceability supports the lifecycle supervision of infrastructure and rolling stock, enabling predictive asset maintenance and historical performance tracking. In road transportation, vehicle diagnostics and traffic systems benefit from traceable operational data to improve safety and reliability.

Organizations such as Boeing and Northrop Grumman adopt PLM-driven digital threads to ensure traceability across design, simulation, and operational phases in the aerospace sector [SG35, SG37, SG56]. Boeing utilizes ISO 23247 to formalize standardized data exchanges across federated DT environments, enabling consistent tracking of changes and system behavior. Northrop Grumman utilizes workflows integrating SymML with analytical tools, such as MATLAB, to maintain traceable paths between system requirements and performance validations.

The defense domain [SG37, SG56] emphasizes rigorous traceability for automated design validation and simulation-to-operation consistency. Linking operational data with model artifacts ensures transparent and auditable decision chains, particularly for mission-critical systems operating across domains.

In the automotive industry [SG35, SG14, SG47], traceability is enabled through PLM platforms like ENOVIA and the 3DEXPERIENCE suite, which connect CAD, simulation, and IoT data streams. These links support design validation, safety analysis, and adaptive configuration management.

In energy and utilities [SG22, SG14], traceability supports the lifecycle of power systems, from planning to real-time operation, through a unified source of truth. Standards like RAMI 4.0 and SGAM provide structured reference models for managing traceable interactions between grid components and monitoring systems.

In the biomedical and life sciences domain [SG14], traceability is essential for compliance with safety and regulatory standards. DTs trace device design, simulation data, and test outcomes to ensure robust audit trails in engineering and clinical validation contexts.

Warehouse logistics and retail [SG47] use traceability to justify supply chain decisions, improve routing transparency, and support dynamic operational adjustments. It enables real-time tracking of goods and links decision logic to demand and customer service levels.

In healthcare [SG47], traceability ensures secure integration of patient records, treatment workflows, and DT behavior to support diagnostics and personalized care.

Lastly, in government and public administration [SG47], traceability underpins transparent policy enforcement, particularly in the management of utilities, transportation, and civic records. Digital threads help link decisions to data sources across institutional boundaries.

4.4. Barriers

Despite its acknowledged importance, traceability remains a consistently under-supported capability in model-based DTE.

One recurring theme is the fragmented treatment of traceability throughout the product lifecycle. As highlighted in Jones et al. [SP01], most traceability mechanisms are limited to the early design phases, with very few approaches ensuring consistent trace links during operation, maintenance, or end-of-life phases. Even when models are rigorously developed, the lack of tools and strategies to maintain semantic and temporal continuity between MBE artifacts and runtime system behaviors undermines the full realization of traceability.

This issue is compounded by the fact that traceability is often implicitly rather than explicitly engineered, as shown in Minerva et al. [SP03] and Lim et al. [SP04]. While lifecycle management and monitoring are acknowledged, traceability is often treated as a by-product rather than a core design concern. The lack of well-integrated frameworks that enable tracing between models, simulations, and live data flows leads to poor visibility into decision rationale and model evolution.

Even when standards such as ISO 10303 (STEP) or IEC 62264 are employed to promote traceability, as mentioned in Wortmann et al. [SP06], their implementation remains limited and often disconnected from tool-supported model transformations [42]. The manual effort required to maintain trace links across system updates introduces fragility, especially in highly dynamic environments.

In distributed and data-intensive contexts, such as those described in Brauner et al. [SP08] and Merkle [SP12], traceability is further challenged by the absence of formal data provenance mechanisms that span multiple modeling languages, data layers, and execution platforms. Although some studies (e.g., Rathore et al. [SP05] and Guinea and Holgado-Terriza [SP14]) explore traceability frameworks or security-enhanced tracking, these efforts are rarely integrated with the underlying MBE workflows, which limits their scalability and auditability.

Additionally, in environments that rely on AI, as emphasized in Arin et al. [SP15], the opacity of model behavior and learning pipelines makes it particularly difficult to establish traceable links between decisions and the original engineering models. Without rigorous traceability frameworks, it becomes difficult to validate AI-enhanced DT systems or ensure regulatory compliance.

Concerning the collected gray literature, a primary challenge lies in the lack of explicit linking mechanisms across hierarchical model layers, from requirements to design, simulation, and runtime data. When DTs span multiple domains or system layers, model-to-model traceability becomes difficult to enforce without robust tooling or formalized transformation pipelines ([SG07, SG14, SG24, SG26, SG35, SG37, SG41, SG47, SG56]).

Another significant barrier is the heterogeneity of data and modeling environments. Ensuring traceability across data-driven models, CAD assets, PLM systems, and embedded analytics requires shared semantics and standard metamodels, which are rarely available or consistently adopted [SG05, SG14, SG22, SG35, SG47]. Although platforms like PROSTEP and TECHNIA attempt to address this through lifecycle data threading, context-aware traceability (i.e., delivering the right information to the right user at the right time) remains a significant gap [SG14, SG24].

Even when traceability mechanisms are in place, governance and validation over time are often lacking. Models may become desynchronized due to evolving requirements, updates in simulations, or system state changes, with no automated way to preserve consistency or justify design rationale [SG37, SG41].

In short, traceability is often treated as a static artifact rather than a dynamic, living relationship across models, reducing its long-term reliability and usefulness.

Highlights – RQ1

- Traceability is primarily addressed at design time but rarely maintained across the full DT lifecycle; emerging approaches include blockchain, runtime models, and PLM-driven digital threads.
- MBE techniques such as model transformation and models@runtime remain underexplored for dynamic traceability support.
- Industrial tools (Siemens Insights Hub, PROSTEP OpenCLM, Papyrus) provide partial traceability support but lack semantic interoperability and full lifecycle integration.
- Traceability is often static and implicit; the absence of standardized metamodels and end-to-end trace link maintenance limits its effectiveness in dynamic and AI-driven DT environments.

5. Enablers and Barriers for Federation (RQ2)

This section presents the results derived from the comprehensive application of the research methodology with respect to RQ2, which explores federation in DT systems, with a focus on model-based strategies that facilitate coordination, integration, and interoperability across multiple DT instances. We identify key enabling factors and supporting tools, as reported in Table 4, including techniques, technologies, and application domains. Additionally, we report on the main barriers and challenges that hinder effective traceability across the DT lifecycle.

5.1. Model-based Digital Twin Engineering Techniques

Minerva et al. [SP03] do not explicitly define the concept of federation but rather introduce it as a capability associated with several challenges in the DT domain, particularly concerning interoperability. The paper cites foundational references on DT federation, including the works of Song et al. [43] and Vural et al. [44], which delve deeper into the federation paradigm.

Within Brauner et al. [SP08], federation is also embedded in the notion of the World Wide Lab (WWL), a conceptual framework intended to integrate data across multiple stakeholders. The authors clarify that the WWL does not adhere to a single architecture or stakeholder model, as such decisions are tailored to individual use cases.

Siriweera et al. [SP11] discuss federation in the context of decentralized cloud computing. The authors describe how such architectures are proposed as a solution to the limitations of centralized cloud systems. Federation here emerges through the interplay between central cloud resources and edge computing layers, forming a decentralized structure.

Arin et al. [SP15] examine the federation of DTs as a foundational principle to support interoperability, scalability, and efficient data exchange among distinct DT instances. While the term "federation" is not explicitly used in all instances, the paper articulates the importance of multi-system DT architectures, in which interconnected twins collaborate to form distributed, intelligent ecosystems. A central theme in the discussion is the hierarchical organization of DTs, where multi-system twinning is framed as a critical architectural layer. Arin et al. [SP15] also emphasize interoperability and data integration as key technical hurdles, advocating for standardized data models and communication protocols to facilitate federation. The authors argue that the absence of uniform standards presents a significant barrier to seamless interoperability among DTs. To address this, the adoption of common ontologies, data formats, and standardized frameworks is recommended. Furthermore, the study highlights the need for real-time data synchronization across distributed DTs to ensure their digital models remain aligned with physical asset changes. This real-time coordination is particularly crucial for achieving responsive and intelligent behavior within networks of federated DTs.

Papadonikolaki and Anumba [SP16] do not offer a formal definition of federation but indirectly reference the concept through Siemens' classification of DT types. According to the cited source, three types of DTs exist, i.e., product, production, and performance, and their integration can enable real-time, dynamic cyber-physical interaction throughout the entire product and production lifecycle. This implicit reference aligns with the idea of federation, where multiple DTs collaborate across different lifecycle stages to achieve comprehensive system-level awareness and control.

Concerning gray literature, the document in [SG22] outlines the importance of collaboration among DTs representing different devices and networks. It emphasizes that a meaningful lifecycle, one that delivers broad value, can only be realized when individual DTs can exchange and integrate information with one another, such as a grid twin drawing insights from a transformer twin. This interconnection is essential for maximizing utility and achieving holistic system behavior.

Although not explicitly framed as a discussion on federation, [SG26] implies the need for managing multiple models and datasets across the lifecycle. The integration of such models inherently requires mechanisms that support federation to maintain coherence, consistency, and traceability throughout a Model-Based Enterprise.

The report in [SG35] highlights how PROSTEP's OpenCLM solution facilitates federation at the model level by establishing intelligent links between distributed data sources. This federated approach empowers stakeholders to access real-time, reliable product information while upholding governance structures that protect sensitive engineering assets. The strategy enables seamless coordination between disparate teams and systems involved in complex product development.

In [SG37], federation is realized through MBE workflows that connect system requirements, simulation data, and physical-world telemetry. This connected framework enables Northrop Grumman to manage model fidelity and traceability across different project stages, ensuring that system models are synchronized with real-world performance and evolving specifications.

According to [SG14], TECHNIA supports model federation using Unified Architecture Frameworks (UAF), such as DoDAF, MODAF, and NAF. These architectural standards facilitate the integration of models from diverse domains, enabling the holistic representation of systems and supporting decision-making across multi-layered enterprises.

Finally, the blog post in [SG47] references the concept of federated DTs when describing scenarios that require multiple interconnected twins to address complementary phases of the lifecycle. It also discusses a federated network of DTs to support multidisciplinary decision-making. However, the post provides limited detail regarding the implementation of such federated infrastructures in practice.

5.2. Tools

Wortmann et al. [SP06] refer to languages like SysML and OWL as enablers of semantic interoperability, a key aspect of model federation. By grounding models in shared ontologies and metamodels, these languages support the federation of heterogeneous systems.

Siriweera et al. [SP11] introduce edge-cloud integration platforms such as Kubernetes, Docker Swarm, and Apache Mesos, which orchestrate DT components across geographically distributed nodes. OPC-UA remains central, providing a standardized interface for communication, particularly when combined with Papyrus for Manufacturing to bridge modeling and runtime federation.

Merkle et al. [SP12] mention Data Distribution Service (DDS) and the Eclipse AASX Package Explorer platform, both of which offer federation via real-time data synchronization in cyber-physical systems. These standards and tools support the dissemination of consistent state data across twin instances.

Guinea et al. [SP14] emphasize the federated nature of Azure IoT Hub and Eclipse Ditto, which provide distributed runtime environments where multiple twins can interact via common protocols and shared semantic models.

PROSTEP's OpenPDM and OpenCLM [SG35] enable model federation by linking distributed systems (e.g., PLM, ERP, MES), supporting real-time collaboration across enterprises while ensuring consistency and governance in complex development environments.

Papyrus for Manufacturing [SG25] supports federation through its integration with BaSyx, a middleware framework aligned with Industry 4.0. Its support for AAS and OPC UA modeling enables interoperability across systems and organizations, making it an enabler for federated DT environments leveraging MBE.

Northrop Grumman's MBE workflow [SG37] supports federation by linking SysML models with domain-specific simulation tools and real-time telemetry, thereby ensuring model coherence across domains.

TECHNIA's modeling stack [SG14], including 3DEXPERIENCE, CATIA, and ENOVIA, enables federation by connecting disparate system models into a unified framework. The architecture supports distributed simulations, integrated PLM, and IoT feedback, forming a basis for cross-domain model collaboration.

Boeing [SG56] also addresses federation by embedding standardized MBE practices into the DT lifecycle. The integration of simulation, manufacturing, and design data into a single federated environment facilitates synchronized collaboration across multiple engineering teams and platforms.

5.3. Domains

DT technologies are applied across a wide range of domains, with growing attention to federation to ensure interoperable systems.

Federation supports the integration of smart factory elements in manufacturing, as outlined by Wortmann et al. [SP06]. The use of domain-specific languages, SysML, and BPMN facilitates the interconnection of process models, machines, and automation layers, creating a cohesive representation of factory operations. Studies such as those by Hasan et al. [45] and Lu [41] emphasize how AI and real-time feedback loops contribute to adaptive and responsive manufacturing systems, where multiple DT instances are federated to function as a unified operational framework.

Werbińska-Wojciechowska et al. [SP20] also identify strong cases for federation in transportation and logistics. In air transportation, DTs are federated across aircraft, airport infrastructure, and air traffic management systems to optimize operations. In railway systems, the federation of infrastructure models, scheduling tools, and train telemetry supports coordinated, real-time optimization. Road systems similarly benefit from federated integration between vehicles, infrastructure, and control systems to manage traffic flow and ensure safety. Finally, in waterborne and intermodal logistics, DTs are federated across ports, vessels, and transportation hubs to enable synchronized management of multimodal transport systems.

Concerning gray literature, Boeing aligns federated DT platforms using ISO 23247 to standardize interoperability between DTs representing aircraft systems, simulations, and operational platforms in the aerospace domain [SG35, SG37, SG56]. Northrop Grumman achieves federation by linking SysML system models with analytical tools, enabling cross-domain model execution and coherence.

The defense sector [SG37, SG56] applies federation to integrate simulation tools and real-time operational data across air, sea, and land systems. This supports the synchronized operation of multi-domain assets in dynamic and mission-critical environments.

In the automotive sector [SG35, SG14, SG47], unified modeling platforms and architectural frameworks, such as UAF, are employed to integrate autonomous vehicle systems, production processes, and simulation environments into a federated DT ecosystem.

Manufacturing [SG35, SG14, SG47] relies on federation to connect simulation platforms, ERP, MES, and other enterprise systems. Platforms like PROSTEP's OpenCLM enable data federation across organizational layers, allowing for collaborative workflows and real-time decision support.

Smart city initiatives [SG47] necessitate the federation of DTs representing infrastructure, traffic, energy, and environmental systems. This federated coordination supports holistic urban planning and responsive governance by integrating data streams from both the public and private sectors.

In the energy and utilities sector [SG22, SG14], federation enables consistent operation across power generation, distribution, and control layers. RAMI 4.0 and SGAM ensure interoperable, layered architecture models that federate transformer twins, grid elements, and supervisory control systems.

The biomedical and life sciences sector [SG14] addresses federation by integrating engineering and clinical simulation platforms (e.g., CATIA, Dymola, DELMIA InUse) into unified DT environments. This supports lifecycle-wide data fusion for design validation, compliance monitoring, and performance analysis.

Logistics and retail [SG47] leverage federation to connect DTs of warehouses, transport fleets, and customer-facing systems. This allows real-time planning optimization and flexible response to changing supply and demand conditions.

In healthcare [SG47], federation integrates data from multiple monitoring systems, hospital infrastructure, and patient management tools into centralized DT frameworks. This supports smart analytics and coordinated clinical interventions.

Finally, public administration [SG47] benefits from federated DT infrastructures that connect departments like transportation, energy, and digital records. This federated coordination enables efficient service delivery and integrated public governance.

5.4. Barriers

A common barrier, identified early in Jones et al. [SP01] and further explored in Minerva et al. [SP03], is the lack of formalized architectures and interaction models that enable multiple virtual entities to be composed, synchronized, and managed. Although many DTs support modularity, this modularity often lacks behavioral coordination, which limits real-time cooperation among federated DTs.

Lim et al. [SP04] and Wortmann et al. [SP06] further point out that existing modeling languages and ontologies (e.g., OWL, AutomationML) are seldom equipped with the operational semantics required to support the orchestration of federated DTs. As a result, semantic integration does not automatically translate into executable coordination, which is essential for distributed decision-making and control.

Madubuike et al. [SP07] in the above-mentioned "World Wide Lab" promotes a global-scale DT federation. However, the study concedes that no standard architecture or stakeholder model exists to support such an initiative. Similarly, Siriweera and Naruse [SP11] and Merkle [SP12] emphasize the technical feasibility of multi-cloud and modular architectures but underscore the absence of shared control protocols and synchronization logic necessary for meaningful federation.

Studies like Arin et al. [SP15] and Papadonikolaki and Anumba [SP16] stress that while multi-system DTs are a growing necessity, especially in domains such as manufacturing, energy, and smart cities, the lack of interoperability standards, governance models, and synchronization frameworks severely constrains the deployment of federated DT ecosystems. Even when DTs operate in parallel, they rarely share behavior, data insights, or collaborative goals in a coordinated manner.

Regarding gray literature, federation is crucial for integrating complex, distributed, and multi-stakeholder DT environments. However, its practical realization in MBE-based DTE faces several systemic obstacles. The most frequently cited issue is the lack of shared data models, ontologies, and syntactic interfaces, which hinders seamless communication between DTs developed using different tools or platforms [SG22, SG26, SG35, SG37, SG47, SG56].

Even when common standards exist (e.g., RAMI 4.0, SGAM, ISO 23247), there is limited guidance on how to operationalize them for federated DT architectures. Current solutions typically rely on ad-hoc integrations, with no clear support for composability or behavioral coordination among federated models [SG22, SG24, SG39].

In many cases, federation is conceptually referenced but technically under-specified. The literature often refers to the benefit of linking multiple DTs (e.g., a grid twin with a transformer twin) but stops short of defining how this federation should be governed, updated, or validated over time [SG22, SG47]. As such, implementations struggle with issues such as data consistency, synchronization lag, and lack of control protocols. Moreover, in industrial contexts, organizational and security constraints further complicate the process of federation. Stakeholders may be unwilling to share sensitive model data without fine-grained access control or trust mechanisms [SG37, SG56]. The absence of dedicated federation middleware or orchestration frameworks reinforces this gap, limiting scalability and reusability across domains.

Ultimately, despite the potential of model-driven practices to promote consistency and modularity, MBE techniques are not yet fully leveraged to support the federation of DTs. This disconnect between modeling and execution layers limits the scalability and flexibility of DT networks, particularly in multi-stakeholder and cross-domain environments.

Highlights – RQ2

- Federation is enabled through hierarchical architectures, semantic data models, real-time synchronization, and shared interface definitions.
- Most current approaches remain conceptual; executable coordination across distributed DTs is still largely missing.
- Tools such as PROSTEP OpenCLM, Papyrus with BaSyx, and OPC UA provide partial integration and runtime coordination support.
- Key barriers include the absence of formal orchestration standards, lack of standardized metamodels, and organizational and security constraints that hinder cross-domain model and data sharing.

6. Survey Results

This section presents the survey results, which complement the MLR by capturing the expectations, challenges, and perceived gaps reported by expert stakeholders actively involved in DT adoption and engineering across diverse organizations.

Demographics and contextual information. The survey collected 51 responses from a diverse audience, representing a balanced mix of backgrounds, including 10 large companies ($\approx$ 20%), 22 academia and research organizations ($\approx$ 43%), and 12 SMEs ($\approx$ 23%), thus providing perspectives from both industry and research. Moreover, 7 participants ($\approx$ 14%) do not specify any affiliation.

Therefore, 20 participants were affiliated with large organizations of over 1,000 employees ($\approx$ 39%), including 13 from companies with 1,001–10,000 employees and 7 from those with more than 10,000. The remaining respondents represented mid-sized organizations, including 4 with 250–1,000 employees ($\approx$ 8%), 7 with 51–250 employees ($\approx$ 13%), and 10 with 11–50 employees ($\approx$ 28%), with 10 participants ($\approx$ 28%) not specifying any affiliation numbers.

In terms of roles, researchers constituted the largest group (26 out of 51, $\approx$ 51% out of the total), highlighting the academic orientation of the sample and the research community's strong interest in DT model-based engineering. Practitioners (e.g., engineers and consultants) were less represented (8 respondents, $\approx$ 16%), while managers accounted for 11 participants ($\approx$ 22%), indicating growing attention to the strategic and organizational dimensions of DT deployment.

Experience levels further reflect the field's maturity. The majority (38 respondents, $\approx$ 75%) reported 0–5 years of experience in DTE, with only 4 indicating 6–10 years ($\approx$ 8%), 1 with 11–15 years ($\approx$ 2%), and none beyond 15 years, pointing to DT's relative novelty. In contrast, experience with Model-Based Engineering (MBE) was broader: 30 respondents had 0–5 years ($\approx$ 59%), 6 had 6–10 years ($\approx$ 12%), 2 had 11–15 years ($\approx$ 4%), and 8 reported over 15 years ($\approx$ 16%), reflecting MBE's more established status.

The survey also captured participants' research and business domains (multiple selections allowed). The most cited areas were DTs and Model-Driven Engineering (both

Table 4: Contributions of MBE tools from the selected peer-reviewed papers (SP) and gray literature (GL). Mapping to DTE phases: (A) Modeling, (B) Simulation, (C) Data Integration, (D) Real-Time Monitoring, (E) Predictive Analytics, (F) Optimization, (G) Scalability, (H) Security and Privacy, (I) Lifecycle Management, (L) User Interaction, (Tra.) Traceability, (Fed) Federation. Application domains: (1) Aerospace, (2) Automotive, (3) Construction, (4) Education, (5) Energy, (6) Healthcare, (7) Manufacturing, (8) Robotics, (9) Smart Cities, (10) Telecommunications, (11) Transportation, (12) Defence, N.A. means Not Available.

Tools	**Source**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**L**	**Tra.**	**Fed.**	**Domain**
AAS Designer	[SG25]	✓	-	-	-	-	-	-	-	-	-	✓	-	2,6,8
AASX Package Explorer	[SP11, SP12]	✓	✓	✓	-	-	-	✓	✓	-	✓	✓	✓	2,6,7,8,9
Ansys Minerva	[SG20]	✓	✓	-	-	-	-	-	-	-	✓	✓	-	N.A.
Ansys ModelCenter	[SG20]	✓	✓	-	-	-	-	-	-	-	✓	✓	-	N.A.
AnsysTwin Builder	[SP05]	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	2,3,4,5,6,7,10,11
ARAS's PLM platform	[SG03, SG14]	✓	✓	✓	✓	✓	✓	-	-	✓	-	✓	-	2,5,6,7
AutomationML	[SP06, SP12]	✓	✓	✓	-	-	-	-	-	✓	✓	✓	-	2,6,7
AWS ecosystem	[SP12]	✓	-	-	-	-	-	-	-	-	-	✓	-	2,6,7
AWS Lambda	[SP05]	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	2,3,4,5,6,7,10,11
Azure DT	[SP14]	✓	-	-	-	-	-	-	-	✓	-	✓	✓	N.A.
Cameo Systems Modeler	[SG42]	✓	✓	✓	-	-	-	-	-	✓	-	✓	-	1
Eclipse Ditto	[SP14]	✓	-	-	-	-	-	-	-	✓	-	-	✓	N.A.
iMBSE	[SG33]	✓	✓	-	-	-	✓	-	-	✓	-	✓	-	N.A.
JuliaSim	[SG16]	✓	✓	✓	-	✓	✓	-	✓	✓	-	✓	-	2
ModelCenter	[SG37]	✓	✓	✓	-	✓	✓	-	-	-	-	✓	-	1,10,12
OPC UA Designer	[SG25]	✓	-	-	-	-	-	-	-	-	-	✓	✓	2,6,8
OpenModelica	[SG42]	✓	✓	✓	-	-	-	-	-	✓	-	✓	-	1
Papyrus for Manufacturing	[SP11, SG25]	✓	✓	✓	-	-	-	✓	✓	-	✓	✓	✓	2,6,8,9
PLM solutions	[SG56]	✓	✓	✓	-	-	-	-	-	-	-	✓	-	1,6
PROSTEP's OpenCLM	[SG35]	✓	✓	-	-	-	✓	-	-	✓	-	✓	✓	1,2,6,12
PROSTEP's OpenPDM	[SG35]	✓	✓	-	-	-	✓	-	-	✓	-	✓	✓	1,2,6,12
Siemens Insights Hub (formerly MindSphere)	[SP05]	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	2,3,4,5,6,7,10,11
Sparx EA	[SG42]	✓	✓	✓	-	-	-	-	-	✓	-	✓	-	1
SysML solutions	[SP06, SG37, SG56]	✓	✓	✓	-	✓	✓	-	-	✓	✓	✓	✓	1,6,10,12
TECHNIA	[SG14]	✓	✓	✓	✓	✓	✓	-	-	✓	-	✓	✓	2,5,6,7
Visure ALM Platform	[SG41]	✓	-	-	-	-	-	-	-	✓	-	✓	-	N.A.

with 19 votes, ≈ 37%), often mentioned together in relation to enhancing DT applications. Other prominent areas included Artificial Intelligence / Machine Learning (AI/ML) and Cyber-Physical Systems (CPS) (both with 17 votes, ≈ 33%), followed by Internet of Things (IoT) (13 votes, ≈ 25%) and Testing (8 votes, ≈ 16%). Less frequently mentioned domains included DevOps, Cloud Computing, Microservices, and Systems of Systems (SoS) (from 3 to 4 votes, ≈ 6–8%). Additionally, 7 votes indicated other domains (≈ 14%), such as Robotics and Energy Efficiency, highlighting the wide range of applications in the DTE landscape.

The survey also investigated participants' experience across various stages of the DT life-cycle, using a 5-point Likert scale where 1 indicates no involvement and 5 represents their main focus. The results indicate that modeling (≈ 67% with a level of 3 or more) and simulation (≈ 61% with a level of 3 or more) are the most established activities, with a notable share of respondents identifying them as primary areas of focus (28% and 8% scoring 5/5, respectively). Other activities, such as real-time monitoring (≈ 45% with a level of 3 or more) and predictive analytics (≈ 31% with a level of 3 or more), also show signs of growing adoption, albeit with lower intensity. In contrast, more advanced and systemic aspects of DT implementation remain underexplored. Topics such as Scalability and Federation (≈ 20% with a level of 3 or more), Security and Privacy, Lifecycle Management, and Traceability (≈ 17% with a level of 3 or more) received relatively low scores, indicating limited practical engagement. Notably, traceability and federation emerged as the least addressed areas, with 69% and 58% of respondents, respectively, selecting 1 (no use). These findings suggest that while foundational DT activities are increasingly adopted, critical challenges such as traceability and federation are still in the early stages of application and maturity in practice.

The survey also explored the application domains where participants have implemented DT in real-world case studies. Manufacturing emerged as the most represented domain (≈ 33%), followed closely by automotive (≈ 31%), underscoring the strong industrial orientation of DT deployments in production and mobility sectors.

Other notable areas included energy and robotics (both ≈ 23%), where reported use cases ranged from battery aggregation to robotic system optimization. Additional domains such as aerospace (≈ 18%) and healthcare (≈ 10%) were identified, typically associated with simulation-intensive and human-centered applications, respectively. A range of sectors, including smart cities, logistics, construction, and education (each ≈ 12–13%), further illustrate the breadth of DT adoption across both technical and socio-technical environments. Less common, but still notable applications were reported in agriculture, food production, marine engineering, retail, and telecommunications (≈ 3–13%), often reflecting emerging or specialized use cases. A small subset of participants (≈ 13%) indicated either domain-agnostic applications, emphasizing the cross-cutting nature of DT methodologies, or no direct application of DTs in specific domains to date.

Enabler and Barriers for Traceability. The open-ended and Likert-based survey responses revealed both enablers and barriers to achieving effective traceability in DT. Participants commonly described traceability as the systematic capability to link and track information across the entire DT life-cycle, connecting physical assets, digital models, data sources, and engineering artifacts. Traceability is widely recognized as a critical and achievable objective in DTE ($\approx 90\%$ with a level of 3 or more), though responses also reflect practical challenges, as $\approx 79\%$ of participants acknowledged that traceability is hindered by a lack of standardized methods and tools.

Key enablers from open-ended responses included the use of model-based systems engineering practices to manage relationships between requirements, models, and operational data ($\approx 20\%$), as well as mechanisms to establish traceable links among artifacts such as sensor data, AI outputs, and functional models ($\approx 36\%$). A few representative responses illustrate this point, such as one noting "*Tracing from Requirements to all other digital engineering artefacts*", highlighting how MBE enables consistent linkage across the DT lifecycle, and another stating "*Being able to locate how and where a given data item was produced*", emphasizing the need for mechanisms to trace the origin and flow of data within the system. Additional enablers included accountability structures, audit trails, and real-time monitoring of changes ($\approx 12\%$), all considered essential for debugging, validation, and certification. For example, "*every entity that builds the DT can be traced*" highlights responsibility tracking, while "*making the flow of data, logic, and decisions transparent and auditable*" reflects structured audit trails. "*Health monitoring of DT*" and "*follow and record everything regarding DT during the whole life cycle*" point to continuous, model-based change tracking. A particularly illustrative example describes the application of MBE in a real use case: "*We apply Model-Based Engineering principles by structuring system data and behavior into reusable models ⋯ These models form the foundation for simulation, validation, and integration with XR-based training environments*". This shows how MBE supports traceability not only across design and validation activities but also into operation and training, ensuring continuity and coherence across the DT lifecycle through integration of real-time sensor data and simulation.

Conversely, several barriers were identified. Participants highlighted the difficulty of tracing heterogeneous data and artifacts, particularly in multi-domain or large-scale systems. A lack of standardization and the need for domain-specific abstractions were cited as major obstacles to consistent implementation. Other concerns included the granularity and fidelity of traceability data, with challenges in balancing detail and manageability throughout the DT lifecycle. Tracing the origin and evolution of model components, especially in dynamic environments, was also seen as problematic. While traceability is widely recognized as critical, its practical implementation remains hindered by both technical and organizational challenges. Several responses illustrate these points. "*Keeping it updated with the real system configuration*" highlights the challenge of maintaining alignment between digital and physical systems. Others, such as "*Federation, Traceability, Unclear understanding ⋯ Too many tools to convert to DTs*" and "*DT integration and Traceability, real-time monitoring ⋯*" point to fragmentation, lack of standards, and broader integration issues. These examples confirm that despite its importance, traceability remains difficult to implement due to conceptual and technical complexity.

Enabler and Barriers for Federation. The responses concerning federation in DT converge on the idea of integrating and coordinating multiple autonomous DTs within a larger, collaborative ecosystem. Federation is broadly recognized as a key enabler for interoperability, data exchange, and system-of-systems (SoS) integration, allowing DTs developed by different stakeholders or vendors to work together while maintaining their independence. Moreover, federation appears as a significantly less mature area. A majority of respondents ($\approx 88\%$) disagreed or remained neutral (scores 1 to 3) with the statement that federation strategies are well-supported by existing tools. At the same time, federation is perceived as facing challenges similar to traceability, with $\approx 79\%$ of participants agreeing that its implementation is hindered by the absence of standard approaches (scores 3 to 5).

Participants identified several practical enablers, including the ability of DTs to communicate, share data, and interact ($\approx 48\%$), as well as the integration of independently developed twins into cooperative, interoperable groups ($\approx 55\%$). The reuse and composition of DT components and services were also cited as a facilitator of federation ($\approx 14\%$). A recurring theme was the need for effective interface management, which may involve the development of ontologies, APIs, or shared semantic models to align heterogeneous DTs and enable consistent information exchange ($\approx 10\%$). The notion of federated DTs collaborating toward a common objective, while remaining individually developed and governed, was highlighted as essential to realizing scalable, flexible DT ecosystems ($\approx 24\%$). Examples include (i) "*a DT can communicate (i.e. share data, call, control) with other DTs*"; (ii) "*Capability to reuse or combine several DT or DT components/services*"; (iii) "*Ability to 'connect' DTs, possibly developed by different vendors. This may require the development of ontologies, APIs, etc.*"; (iv) "*Federation enables the integration and interoperability of multiple DTs, ensuring data exchange and synchronization*". These insights highlight federation as a key principle for building modular and adaptable DT architectures. Therefore, several MBE enablers emerged from the responses. Techniques such as SysML modeling, simulation, and FMI-based co-simulation were cited as key to structuring modular, interoperable DTs. Requirement engineering and model validation support traceability and consistency, while composition and reuse of models enable scalable integration.

Conversely, respondents also underscored several critical barriers. Chief among these is the complexity of ensuring semantic interoperability across DTs built for different domains or by different vendors. The absence of unified standards and frameworks exacerbates integration challenges, particularly when incorporating legacy systems or reconciling divergent data models. Another major concern is the coordination overhead required to maintain a consistent operational view across federated DTs, which can hinder scalability and maintainability. Additionally, respondents noted the difficulty of orchestrating loosely coupled DTs without compromising their performance or model fidelity. In sum, while federation is viewed as a foundational strategy for scalable DT architectures, its practical implementation is constrained by significant technical and organizational challenges related to integration and interoperability. These challenges were frequently cited across responses. Interoperability and standardization appeared in ≈ 25% of the responses, including "*interoperability, composition/federation, feedback generation*", "*missing standards*", and "*standardisation that covers various fields*". Tool fragmentation and lack of conceptual clarity were highlighted in ≈ 15% of the responses, such as "*1000 different tools*", "*Lack of easy to use frameworks*", and "*Too many tools to convert to DTs*". The difficulty of maintaining synchronization and alignment with the real system was raised in ≈ 13% of the responses, including "*Update to real system evolution, synchronization with real system*" and "*Keeping it updated with the real system configuration*". Complexity and scalability concerns were evident in ≈ 15% of the responses, including "*multiple tools, complexity in integration, complexity in instantiation*" and "*Federation, Traceability, Unclear understanding of the concept itself*".

Model-based engineering, traceability and federation in DT. The survey results highlight the central role of MBE as a foundation for DTE, with a strong consensus among participants: 94% agreed or strongly agreed (scores 4 to 5) that MBE provides a robust framework for DT development (average score 4.0). Participants reported a moderate perception of tool support for MBE principles (≈ 34% agreed or strongly agreed with scores from 4 to 5). Responses were mixed, with ≈ 30% indicating neutrality and only ≈ 7% strongly agreeing that current tools provide effective MBE support. Perceptions of tool support for traceability and federation were notably lower, with ≈ 26% and ≈ 17% agreeing or strongly agreeing with scores from 4 to 5, respectively. A majority of respondents expressed disagreement or neutrality, i.e., 73% for traceability and 83% for federation, reflecting widespread dissatisfaction with current tool capabilities in these areas. In contrast, the strategic importance of collaboration between academia and industry in advancing MBE for DTE was strongly affirmed. This item received the highest score, with 80% of respondents agreeing or strongly agreeing (scores 4 and 5). However, a critical view of the current literature emerged with respect to traceability in MBE for DTs. Here again, ≈ 93% of participants expressed disagreement or neutrality regarding the adequacy of available guidance. These findings highlight a disconnect between the conceptual recognition of MBE's relevance and the practical limitations of current tools and literature in addressing advanced concerns such as traceability and federation. While MBE and cross-sector collaboration are widely seen as essential, the lack of mature, standardized methods and tooling remains a significant barrier to their effective implementation in practice.

7. Discussion

This section presents the main findings regarding the enablers and barriers to traceability and federation in model-based DTE. It examines how some of the identified barriers may be addressed through existing model-based techniques that are not yet widely applied in the DT domain. Finally, it outlines a forward-looking research agenda targeting those barriers for which no current solutions have been identified.

To facilitate a cross-RQ comparison, Table 5 synthesizes the main enablers and barriers identified in this study. For each key point, the table reports the dimensions in which the current work is concentrated, the main achievements obtained, and the corresponding advantages and limitations. This comparison complements the detailed narrative of Sections 4 and 5 by making the maturity, strengths, and remaining gaps of current model-based DTE practices more explicit.

To make the mixed-method integration explicit, Table 6 presents a compact joint evidence trace for the main enablers and barriers discussed in this section.

7.1. Summary and Thinking on RQ_1: Enablers and Barriers for Traceability

As detailed in the MLR results in Section 4 and triangulated with the survey evidence reported in Section 6 (see Table 6), traceability is consistently acknowledged as a fundamental capability in the MBE of DTs. However, despite this consensus, our study reveals that traceability remains unequally supported in the different phases of DTE.

Currently, traceability is mostly considered during the early design phases. Thus, traceability links are frequently built and used between requirement definitions, architectural models, and functional specifications. This is a positive foundation, but unfortunately, such links are less often propagated forward into downstream models for simulation, implementation, deployment, or monitoring. In dynamic DT environments (especially when involving AI components or runtime adaptation), this lack of continuity can limit the alignment of the DT with the system, as well as its explainability and confidence level. This global situation is particularly problematic in critical scenarios where

Table 5: Comparative synthesis of main enablers and barriers for RQ1 and RQ2

RQ	Type	Key point	Dimensions in which current work has been carried out	Achievements obtained	Advantages	Limitations / disadvantages
RQ1	Enabler	Design-time traceability through MBE artifacts	Requirements, architecture, functional models, simulation models, and early lifecycle phases	Explicit linking between requirements, models, and verification artifacts; better design rationale capture	Strong conceptual fit with MBE; supports consistency, auditability, and compliance	Rarely propagated to runtime, maintenance, or end-of-life phases
RQ1	Enabler	Lifecycle traceability via digital thread / PLM integration	PLM environments, CAD/CAE integration, industrial toolchains, and manufacturing/aerospace settings	Digital thread support across engineering repositories; improved cross-tool visibility	Better lifecycle continuity; supports validation, accountability, and change management	Often tool-specific, costly to customize, and weakly integrated with model transformations
RQ1	Barrier	Lack of end-to-end maintenance of trace links	Runtime evolution, monitoring, adaptive DTs, and AI-enabled DT contexts	Recognition of the need for dynamic traceability and synchronization	Motivates investment in models@runtime and automated trace link maintenance approaches	Trace links are often static, implicit, or manually maintained; poor support for automation
RQ1	Barrier	Heterogeneity and lack of shared metamodels / provenance mechanisms	Multi-model environments, cross-domain engineering, and data-intensive DTs	Partial use of standards and provenance-related solutions	Encourages interoperability-oriented thinking and formalization	Shared semantics and standard metamodels are still rare; scalability and auditability remain weak
RQ2	Enabler	Federation through shared semantic models and interface definitions	Hierarchical architectures, semantic data models, APIs, ontologies, and co-simulation	Integration of multiple autonomous DTs and exchange of data/services	Supports modularity, interoperability, and system-level collaboration	Semantic alignment does not automatically yield executable coordination
RQ2	Enabler	Partial runtime integration through tools and coordination platforms	Edge-cloud settings, middleware, and industrial integration stacks	Initial support for synchronization and coordinated execution across DTs	Enables reuse, modular composition, and distributed interaction	Implementations are still fragmented and often limited to specific platforms or domains
RQ2	Barrier	Lack of orchestration standards and executable federation support	Distributed DT ecosystems, multi-stakeholder deployments, and cross-platform settings	The problem is clearly recognized across literature and practice	Provides a shared vocabulary for orchestration and governance requirements across the community	Most current approaches remain conceptual; coordination logic is under-specified
RQ2	Barrier	Socio-technical and security constraints	Cross-organizational sharing, access control, governance, trust, and data ownership	Stronger recognition of non-technical conditions for federation	Broadens federation beyond pure technical interoperability	Limits data/model sharing and hinders real-world deployment, even when technical solutions exist

DTs are leveraged to support verification and validation activities. Without traceable pathways from system requirements to runtime behavior and vice versa, it becomes difficult to justify that the DT behaves in accordance with the real system. As a consequence, the credibility of the performed analysis, decision support, and certification activities can be weakened.

In addition, in our study, we observed that there is an issue with the way traceability is often treated in current practices. First, traceability information is quite often implicit and diluted in documentation artifacts (e.g., in text) rather than represented as explicit traceability links in/between models. Second, and directly related to the previous point, this traceability information is usually more static than dynamic and does not systematically evolve when the system or data does. This is incompatible with the demands of DTs operating in real-time, distributed, and learning-enabled system-based contexts. Importantly, the problem is not only technical, as we still lack a commonly recognized and standardized way of representing (i.e., modeling) the required traceability information. In fact, even if standards such as ISO 10303 (STEP) or ISO 23247 exist, they are rarely enforced or deployed across the DT and system boundaries.

Despite this global observation, several traceability-enabling technologies are already available on the technical side. We can notably mention blockchain for immutable tracking [46], PLM-driven digital threads for life cycle traceability [4], or runtime models for maintaining operational traceability [47]. Industrial tools such as Siemens Insights Hub (formerly MindSphere)[5], PROSTEP OpenCLM[6], CEA Papyrus[7], and Dassault Systemes 3DEXPERIENCE[8] are practical examples of such solutions. However, our study found that these are often adopted in isolation, need extensive customization, and lack semantic interoperability and formal integration into MBE workflows.

[5] PLM SW: https://plm.sw.siemens.com/en-US/insights-hub/
[6] PROSTEP: https://openclm.prostep.com/en/
[7] Papyrus: https://list.cea.fr/en/page/papyrus-outil-open-source-de-modelisation-graphique-uml/
[8] 3DEXPERIENCE: https://www.3ds.com/3dexperience/

Table 6: Joint evidence trace across peer-reviewed, gray, and survey evidence

Theme	Peer-reviewed	Gray	Survey	Integrated interpretation
Traceability is fundamental but unevenly supported across the lifecycle	[SP01], [SP05], [SP08]	[SG03], [SG07], [SG35], [SG37]	H.1.b; open-text definitions of traceability; high agreement that traceability is critical/achievable	Strong convergence: high relevance is acknowledged, but lifecycle continuity remains weak
Traceability lacks standardized methods, tool support, and guidance	[SP01], [SP03], [SP04], [SP06], [SP12], [SP15]	[SG05], [SG14], [SG22], [SG35], [SG47]	H.1.c; H.2.b; H.2.e; open-text comments on fragmentation and unclear support	Strong convergence between literature and survey on immaturity of support
Digital-thread / lifecycle platforms provide partial traceability enablers	[SP05], [SP14]	[SG03], [SG35], [SG37], [SG56]	H.2.b (low perceived tool support for traceability); Open-text comments on linking requirements, models, data, and audit trails	Complementarity with divergence: existing tools are documented across literature and gray sources, but H.2.b reveals low perceived adequacy, suggesting a gap between tool availability and practical utility
Federation depends on interoperability, shared semantics, and modular integration	[SP08], [SP11], [SP14], [SP15]	[SG14], [SG35], [SG37], [SG56]	Open-text comments on communication, integration, reuse, APIs, and ontologies	Convergence on federation as an architectural and semantic integration problem
Federation is still weakly supported by tools and lacks operational orchestration standards	[SP01], [SP03], [SP04], [SP06], [SP11], [SP12], [SP15], [SP16]	[SG22], [SG24], [SG39], [SG47], [SG56]	H.1.d; H.1.e; H.2.c (low perceived tool support for federation); open-text comments on missing standards, tool fragmentation, and coordination complexity	Strong convergence on the gap between conceptual federation and executable federation; H.2.c confirms divergence between documented tools and perceived support
Organizational, governance, and security constraints hinder federation in practice	[SP15], [SP16]	[SG37], [SG56]	Open-text comments on multi-stakeholder difficulty, fragmentation, and practical coordination issues	Complementarity: the survey sharpens the socio-technical dimension already visible in the literature

From an MBE standpoint, the potential of well-known techniques in the community, such as models@runtime or model transformation, remains underexplored. Some work already exists on the use of model transformation to support the generation of traceability links between different types of models [48]. These approaches could be further developed to enable the automated propagation and validation of traceability information, which is particularly relevant in our DTE context. In addition, models@runtime could be considered to support closed-loop synchronization between DT models and live systems [49]. However, all these approaches and the underlying techniques still require better tool support, more explicit methodological guidance, and more practical integration with runtime infrastructures.

To conclude, we can state that traceability in DTE is caught in a tension. Although it is widely valued and theoretically supported by MBE principles and techniques, it is still not sufficiently or efficiently covered in practice. It is a transversal concern that is affected by different aspects that could possibly disrupt the traceability effort: (i) fragmentation of the organizations and stakeholders, (ii) heterogeneity of the technical components (of the DTs and the systems), and (iii) multiplicity of the standards (e.g., for modeling). Therefore, unlocking its full potential requires treating traceability not only as a by-product but as a core strategic system-level capability. This should also be a fundamental capability of any DT and, as such, natively embedded in the supported tool chain across all the DTE phases.

7.2. *Summary and Thinking on RQ_2: Enablers and Barriers for Federation*

As also detailed in the MLR results in Section 5, and visible in the analysis of the associated survey in Section 6 (see Table 6), federation is increasingly recognized as a fundamental capability in the MBE of DTs. This is becoming more and more true in current industrial environments where large (systems of) systems and corresponding scalable DTs need to interact together instead of working in isolation. However, despite this growing theoretical interest, our study shows that federation remains widely underexplored in practice today.

In fact, we identified several different approaches and enabling technologies that possibly support DT federation or associated capabilities (e.g., integration, composition, synchronization). To the best of our knowledge, much of the existing work provides only conceptual visions and related models (e.g., for architecture or data). Thus, the current state-of-the-art still lacks executable implementations and practical validation in industrial use cases, considering the coordination between distributed DTs.

From a technical perspective, tools such as PROSTEP OpenCLM or CEA Papyrus (with BaSyx) already enable a certain level of integration and runtime coordination between federated DTs. Nevertheless, when it comes to the MBE of DTs, a clear disconnect remains between design-time modeling practices and runtime federation needs. Although models are already used to represent systems and their interactions, the modeled information is often not propagated to orchestrated execution environments. As a consequence, this strongly limits their practical value in the context of synchronized, federated DT-based operations. Addressing this gap requires not only more expressive modeling languages, but also corresponding tool chains to support continuity between different DTE activities (e.g, design, deployment, maintenance).

Moreover, traceability (as summarized in Section 7.1) has too often been treated as an isolated feature, while it is deeply entangled with federation in practice. Without traceability links between the different phases of DTE, any attempt to federate DTs by relying on their models, data, or behaviors can result in inconsistencies or fragmentation. In our study, we observed that there is still a lack of standardized metamodels and end-to-end maintenance support for traceability links in the context of distributed systems and corresponding DTs.

Another key aspect relates to the socio-technical dimension of federation. Even when technical traceability and federation are achievable, organizational and security constraints can limit their concrete application. For example, resistance to change, lack of trust between stakeholders, or restrictions imposed by internal processes often hinder model and data sharing (as a prerequisite for federation). The results of our study suggest that federation is as much about organizational alignment and policies as it is about technical concerns.

To conclude, we can state that federation is in a maturing state when it comes to the MBE of DTs. Although it is already recognized as fundamental, it is still more theoretically than practically considered. In fact, federation is a complex concern that implies embedding traceability as a first-class capability (cf. Section 7.1), bridging the existing execution gap in MBE workflows, while also considering organizational and security constraints, as well as associated limitations. Moving in this direction, the Manufacturing, Aerospace, and Smart Cities domains are already leading the way towards the adoption of federated DTs for integrated system-level collaboration. As a result, we believe that this will foster more and more operational validation of the proposed conceptual visions in the context of domain-specific industrial use cases.

7.3. Open Challenges and Research Roadmap

From a practical point of view, engineering teams cannot assume that MBE alone is sufficient to ensure continuity and coherence within and between systems and their corresponding DTs. The current limitations to properly integrating traceability into runtime contexts, or to realize in practice federation beyond high-level conceptual visions (i.e., architectures), restrict the ability of DTs to efficiently support trustworthy decision-making, system-level verification and validation, or system self-adaptation, among others.

In principle, MBE offers a promising foundation for providing the expected traceability and federation support. However, it remains largely underexploited to date. As a consequence, core MBE techniques such as model transformation [50], model views [51], or model execution [52], must still be refined, extended, and better integrated into dedicated tool chains. In this way, the community can enable more dynamic, context-aware, and cross-domain DT operations in the future.

Based on the results of our study, as summarized in Section 7.1 and Section 7.2, we notably identify the following main open challenges which are directly related to the MBE of DTs:

- **Provide life cycle-wide traceability**: We need to develop methods and tools dedicated to the creation, handling and maintenance of consistent, bidirectional trace links between all phases of the MBE of DTs. This implies being able to trace the involved models and data from the initial requirements and design phases to the deployment, implementation, and runtime phases (and vice versa). This also has to include support for evolving DTs when required by associated system evolutions.

- **Ensure semantic interoperability**: Using the traceability support provided, we have to propose approaches that allow different interconnected DTs to interact in a way that they can understand each other. To this end, there is a need for better aligning MBE standards, data formats, and ontologies, to support semantically traceable workflows across heterogeneous tools and domains.

- **Support executable federation**: Thanks to the work carried out on the previous items, we should be able to realize federation in practice. Thus, by combining traceability with semantic interoperability, we need to move from static and isolated DT architectures to orchestrated and real-time collaboration among distributed DTs. This should, in particular, include dedicated support for synchronization, coordination, and governance between the involved DTs.

- **Implement tool chain integration**: All these new or updated traceability and federation capabilities need to be implemented in a way that allows them to be easily combined. As a result, they can be used to compose different possible toolchains supporting various domain- and/or use-case-specific workflows. To do this, we need to bridge the current division that exists between many MBE tools and operational industrial platforms. In this way, we should be able to support feedback loops, co-evolution, and trace validation in the context of living systems and corresponding DTs.

- **Consider a socio-technical alignment**: Realizing that all the previous items will not be sufficient unless they also seriously consider different kinds of organizational and security aspects, among other non-technical ones. This means addressing resistance to change, data ownership and related access policies, trust in cross-stakeholder environments, etc., in the context of the MBE of DTs where traceability and federation must operate collaboratively.

Ultimately, advancing the MBE of DTs requires a paradigm shift. Instead of considering traceability and federation as secondary features, they must be regarded as first-class architectural principles and related capabilities. These are fundamental for dealing with the technical, organizational, and epistemic fabric of DTs and the corresponding large and complex systems.

8. Threats To Validity

This section discusses the threats to validity according to the classification by Cook and Campbell [53].

8.1. Internal Validity

Threats to internal validity are related to possible erroneous conclusions, i.e., how well the results represent those reported in the literature or how accurately its results reflect the studied group. Internal validity was safeguarded by rigorously following systematic guidelines across all research activities, including both the literature reviews and the expert survey. To ensure the accuracy of the extracted and synthesized data, we used descriptive statistics and conducted cross-analysis across multiple categories from the extraction form. Sanity checks were applied to detect and resolve inconsistencies, ensuring robust and reliable data handling throughout the study. We acknowledge that inter-rater reliability (IRR) was not formally quantified during the data extraction phase (e.g., via Cohen's kappa [25]). While two reviewers independently extracted data and disagreements were resolved through discussion with a third team member, the absence of a formal IRR metric represents a residual threat to internal validity. The expert survey was conducted during an Open-Door Event of the MATISSE project, involving both consortium representatives and external participants. However, some potential threats may have persisted, including limitations in the dissemination strategy, variations in participants' skill levels and domain knowledge, and challenges related to the real-time survey design, such as balancing the number and complexity of the questions.

8.2. External Validity

Threats to external validity concern the generalizability of the results. While the expert survey yielded timely and relevant insights, the participant pool, mainly researchers and early adopters, may limit broader applicability. This underscores the need for further engagement with practitioners and operational experts to better capture real-world DTE challenges. A core challenge in systematic reviews is ensuring the selected studies represent the broader state of the art and practice. To mitigate this, we conducted a comprehensive search across four major scientific databases and complemented it with a gray literature review to include insights from the field. Additionally, we applied recursive backward and forward snowballing to expand coverage. Language bias is a potential threat, as only English-language studies were considered. However, since English is the dominant language in computer science and software engineering, this threat is minimal. Another limitation is that the terminology surrounding DTs and model-based engineering is broad and evolving. By intentionally focusing the search on canonical DT and MBE terms, our review may have missed studies that discuss closely related concepts exclusively through alternative labels, such as lifecycle management, digital thread, or domain-specific simulation terminology, without explicitly framing them as DT or model-based engineering contributions. We mitigated this risk through gray literature analysis, snowballing, and expert input, but we acknowledge that some relevant studies may still have remained outside the final corpus.

8.3. Construct Validity

Construct validity may be influenced by the formulation of search strings for both peer-reviewed and gray literature searches. To address this, we deliberately used a simple yet inclusive search string aimed at maximizing coverage without compromising focus. All retrieved studies were rigorously screened using clearly defined inclusion and exclusion criteria to ensure the relevance and quality of the selected sources.

8.4. Conclusion Validity

To mitigate threats to conclusion validity, we followed a systematic and transparent approach across all stages of the study. All authors were actively involved in designing the extraction form, conducting data extraction, and performing the synthesis to ensure consistency and accuracy. Additionally, we provide a complete and publicly available replication package [23] to support independent verification and reproducibility. The iterative refinement of the extraction form, informed by primary studies, further contributed to a comprehensive and reliable analysis.

9. Conclusion

Digital twin engineering has emerged as a transformative approach in industrial and societal domains, with Model-Based Engineering playing a crucial role in structuring its methodologies. This study investigated the current state of model-based digital twin engineering, focusing on enablers and barriers to these pivotal concerns. Through a mixed-method approach combining a multivocal literature review and a survey of expert stakeholders, we identified the gaps and opportunities in existing methodologies

9.1. Concluding viewpoints

Based on the combined evidence from the MLR and the survey, we can now formulate a clearer concluding view of the barriers that currently hinder the development of DTs in model-based DTE. These barriers do not belong to a single category. Rather, they span at least three strongly connected dimensions: a technical dimension, an organizational and operational dimension, and a conceptual and methodological dimension.

The technical dimension concerns the engineering infrastructure required to make traceability and federation effective in practice. Across the reviewed literature, the most recurrent barriers are the lack of life cycle-wide trace link maintenance, the absence of standardized metamodels and shared semantics, the heterogeneity of tools, models, and data sources, and the weak continuity between design-time models and runtime environments. In the case of federation, additional technical barriers include the lack of executable orchestration, synchronization, and governance mechanisms across distributed DTs. The survey reinforces this interpretation. Participants largely recognized the value of MBE for DT development, yet they expressed much weaker confidence in the current support for traceability and federation, which confirms that the main challenge is no longer only to define these capabilities conceptually, but to operationalize them through mature methods and integrated tool chains.

The organizational and operational dimension concerns how DTs are engineered, governed, and used across stakeholders, teams, and enterprises. The MLR shows that even when technical solutions exist, federation and traceability remain constrained by data ownership, access policies, security requirements, fragmented responsibilities, and limited willingness to share models and data across organizational boundaries. Similar concerns emerged from the survey, especially around integration complexity, the coexistence of multiple tools, and difficulties in maintaining alignment with the real system over time. These observations indicate that the barriers are also rooted in enterprise structures and business practices. In other words, DT development is hindered not only by missing technologies, but also by the absence of governance arrangements that support collaboration, accountability, and trusted information exchange across the DT lifecycle.

The conceptual and methodological dimension concerns how the community frames, understands, and supports these concerns. Both the MLR and the survey indicate that traceability and federation are widely acknowledged as important, yet they are still often treated as secondary qualities, implicit by-products, or high-level aspirations rather than as explicit engineering capabilities. In the literature, federation remains largely conceptual, while traceability is often static, partial, or embedded in documentation rather than represented and maintained as first-class model relations. The survey complements this picture by showing dissatisfaction with the current guidance and by explicitly revealing that, for some respondents, the concepts themselves remain unclear. This suggests that the field still lacks sufficiently consolidated methodological guidance for how these capabilities should be modeled, implemented, validated, and evolved in practice.

Taken together, these findings lead to three main concluding viewpoints. First, the barriers hindering DT development are not purely technical; they are socio-technical and methodological, because they emerge from the interaction between engineering limitations, organizational constraints, and insufficiently mature conceptual guidance. Second, traceability and federation should be treated as first-class architectural capabilities in model-based DTE, since weaknesses in either one directly affect interoperability, validation, decision support, and the long-term credibility of DTs. Third, meaningful progress requires coordinated advances across all three dimensions: stronger MBE methods and tool integration, clearer governance and collaboration mechanisms across stakeholders, and more explicit methodological guidance to turn currently fragmented practices into repeatable and scalable engineering solutions.

9.2. Recommendations for Future Research and Key Breakthrough Areas

Based on the combined evidence from the MLR and the survey, future research should move beyond recognizing traceability and federation as desirable properties and focus on turning them into operational engineering capabilities. The main implication of our findings is that the next advances in model-based DTE will not come from isolated technical improvements alone, but from coordinated progress across technical, organizational, and methodological dimensions. In particular, the survey confirms that participants largely value MBE for DT development, while expressing much lower confidence in the current support for traceability and federation. This indicates that a central research priority is to close the gap between conceptual relevance and practical operability.

A first key breakthrough area concerns life cycle-wide traceability. Future studies should develop methods and tools that maintain explicit, bidirectional, and evolving trace links across requirements, design models, simulations, implementation artifacts, deployment configurations, and runtime data. This is especially important in dynamic DT settings, where systems evolve continuously and where static documentation is insufficient to preserve consistency and explainability over time. In this context, AI-driven mechanisms appear promising, not as a replacement for formal modeling, but as a complementary means to support trace link maintenance, anomaly explanation, and context-aware access to traceability information. Advancing this area would directly address one of the most persistent barriers identified in both the literature and the survey, namely the lack of end-to-end visibility across the DT lifecycle.

A second key breakthrough area concerns semantic interoperability and executable federation. Our results show that federation remains limited not only by the absence of shared semantics and standardized metamodels, but also by the lack of operational mechanisms for orchestration, synchronization, and governance across distributed DTs. Future research should therefore focus on protocols, modeling abstractions, and runtime mechanisms that allow multiple DTs to cooperate in a coordinated and verifiable manner. This requires moving from conceptual architectures toward executable federation support that can handle heterogeneous tools, data sources, and domain models while preserving consistency and trust. In this sense, cross-domain interoperability and system-of-systems perspectives should become central research directions rather than peripheral considerations.

A third breakthrough area concerns integrated tool chains and methodological consolidation. Current practices remain fragmented across modeling environments, simulation platforms, lifecycle tools, and operational infrastructures. Future work should investigate how core MBE techniques, such as model transformation, model views, and model execution, can be better integrated into coherent DTE tool chains that bridge design-time and runtime concerns. At the same time, the field needs clearer methodological guidance on how traceability and federation should be modeled, implemented, validated, and evolved in practice. This also calls for more empirically grounded studies in industrial settings, where proposed solutions can be assessed with respect to scalability, maintainability, governance constraints, and long-term adoption.

Finally, our findings suggest that future progress also depends on addressing organizational and operational barriers as first-class research concerns. Data ownership, access control, security, stakeholder fragmentation, and limited willingness to share models and data are not external constraints that can be postponed until after the technical solution is available. Rather, they shape whether technical solutions can be adopted at all. For this reason, future studies should investigate governance-aware approaches to DTE, in which technical interoperability, organizational accountability, and trusted collaboration are co-designed. Overall, we see the most promising future direction in combining standardized modeling approaches, AI-supported traceability, executable federation protocols, cross-domain interoperability, and system-of-systems engineering principles into repeatable and scalable DTE practices

Acknowledgments

This research work has been funded by the Chips Joint Undertaking (JU) through the project MATISSE, grant agreement No. 101140216.

Appendix A. Survey Questionnaire

This appendix reports the complete questionnaire adopted for the survey, including item wording, response formats, and conditional elaboration prompts. Table A.7 maps the main interpretive constructs used in Sections 6 and 7 to the corresponding questionnaire items.

A. General Information

A.1 Organization type. Response options: SME; Large company; Academic institution; Research organization.

A.2 Organization size. Response options: 1–10; 11–50; 51–250; 250–1,000; 1,001–10,000; more than 10,000.

A.3 Role in the organization. Response options: Researcher; Practitioner (e.g., engineer, technician); Manager; Other (please specify).

A.4 Years of experience in Digital Twin Engineering. Response options: 0–5; 6–10; 11–15; more than 15.

A.5 Years of experience in modeling and simulation. Response options: 0–5; 6–10; 11–15; more than 15.

A.6 Years of experience in Model-Based Engineering (MBE). Response options: 0–5; 6–10; 11–15; more than 15.

A.7 Years of experience in Artificial Intelligence. Response options: 0–5; 6–10; 11–15; more than 15.

A.8 Research/business area(s) (select all that apply). Response options: Digital Twins; Model-driven Engineering; DevOps; AI/ML; Cloud Computing; Internet of Things; Cyber-Physical Systems (CPS); Systems of Systems (SoS); Microservices; Testing; Other (please specify).

A.9 Direct involvement in the MATISSE project. Response options: Yes; No.

B. Digital Twin Engineering

B.1 What immediately comes to mind when you think of Digital Twin Engineering (DTE)?

B.2 Briefly describe how your work relates to DTE, MBE, and/or AI.

B.3 In which engineering activities have you applied or experienced the Digital Twin concept? (Select all that apply.) Response options: Modeling; Simulation; Data integration; Real-time monitoring; Predictive analytics; Optimization; DT interoperability; DT scalability; DT security and privacy; Lifecycle management; User interaction; Federation; Traceability; Other.

B.4 What are the main challenges in engineering digital twins?

Table A.7: Mapping from survey items to analytical constructs

Construct	Survey item(s)	Response format	Role in analysis
Conceptual relevance of MBE for DTE	H.1.a (MBE robustness)	5-point Likert + optional comment	General perception of MBE as enabler
Traceability relevance	H.1.b (traceability criticality/feasibility)	5-point Likert + optional comment	Perceived importance and feasibility of traceability
Traceability standardization gap	H.1.c (lack of standardized methods/tools)	5-point Likert + optional comment	Perceived barrier for traceability
Federation tool maturity	H.1.d (tool support for federation)	5-point Likert + optional comment	Perceived maturity of federation support
Federation standardization gap	H.1.e (lack of standardized methods/tools)	5-point Likert + optional comment	Perceived barrier for federation
Tool support for MBE principles	H.2.a (tool incorporation of MBE principles)	5-point Likert + optional comment	Practical MBE support
Tool support for traceability	H.2.b (tool support for traceability)	5-point Likert + optional comment	Practical traceability support
Tool support for federation	H.2.c (tool support for federation among DTs)	5-point Likert + optional comment	Practical federation support
Academia–industry collaboration	H.2.d (importance of academia–industry collaboration)	5-point Likert + optional comment	Contextual enabler
Adequacy of guidance	H.2.e (sufficiency of literature guidance)	5-point Likert + optional comment	Adequacy of available methodological guidance
Qualitative definition of traceability	B.5	Open text	Interpretive support for traceability meaning
Qualitative definition of federation	B.6	Open text	Interpretive support for federation meaning
Contextualization variables	A.*, C.*, D.*, E.*, F.*, G.*	Closed/open questions	Profiling, context, and analytical segmentation

B.5 How would you define **traceability** in DTE?

B.6 How would you define **federation** in DTE?

C. DTE and Application Domains

C.1 In which application domains have you already used Digital Twins? (Select all that apply.) Response options: Agriculture; Aerospace; Automotive; Construction; Domain-agnostic; Education; Energy; Food production; Healthcare; Logistics; Marine engineering; Manufacturing; Retail; Robotics; Smart cities; Telecommunications; Other.

C.2 In which application domains do you consider Digital Twins to be most relevant? Please rate each domain on a 1–5 scale. Rated domains: Agriculture; Aerospace; Automotive; Construction; Education; Energy; Food production; Healthcare; Logistics; Marine engineering; Manufacturing; Retail; Robotics; Smart cities; Telecommunications; Other; I do not know.

D. DTE, Standards, Best Practices, and Tools

D.1 Have you adopted or encountered standardized methodologies, standards, or best practices in DTE? If yes, please specify.

D.2 Have you adopted or encountered any solution, tool, or framework for DTE? If yes, please specify whether it is an internal or third-party solution, and whether it is domain-specific or broadly applicable.

E. DTE and Artificial Intelligence

E.1 In your DTE experience, have you integrated or encountered approaches based on Artificial Intelligence or Large Language Models (LLMs)? If yes, please describe your experience.

F. DTE and Research References

F.1 Please recommend any references (e.g., research papers, white papers, websites) that have significantly influenced your understanding or practice of DTE.

G. Model-Based Engineering Expertise

G.1 Please rate your level of expertise in the following MBE concepts as applied to DTE:

- Model
- Abstract syntax/metamodel
- Concrete syntax/notation
- Modeling language
- Model semantics
- Model simulation
- Executable model
- Model transformation
- Megamodel

Expertise was rated on the following five-point scale:

- **1 – Not familiar**: little or no theoretical knowledge; no practical experience.
- **2 – Beginner**: basic awareness; limited practical experience.
- **3 – Intermediate**: sound understanding of core principles; some practical experience.
- **4 – Advanced**: strong theoretical and practical knowledge across multiple projects.
- **5 – Expert**: extensive theoretical and practical mastery; able to address complex cases and mentor others.

H. Perceptions on DTE and MBE

H.1 Please indicate your level of agreement with the following statements using a five-point Likert scale (1 = strongly disagree, 5 = strongly agree):

(H.1.a) MBE provides a robust framework for DT engineering.

(H.1.b) Traceability is a critical and achievable aspect of DT engineering.

(H.1.c) Traceability in DT engineering is hindered by the lack of standardized methods and tools.

(H.1.d) Federation strategies are well supported by existing DT engineering tools.

(H.1.e) Federation in DT engineering is hindered by the lack of standardized methods and tools.

H.2 Using the same five-point Likert scale, please indicate your level of agreement with the following statements:

(H.2.a) The tools I use for DT engineering effectively incorporate MBE principles.

(H.2.b) The tools I use for DT engineering effectively support traceability.

(H.2.c) The tools I use for DT engineering effectively support federation among two or more digital twins.

(H.2.d) Collaboration between academia and industry is crucial for advancing MBE in DT engineering.

(H.2.e) The current literature provides sufficient guidance on integrating traceability into MBE for DT engineering.

References

[1] M. Grieves, J. Vickers, Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems, Springer International Publishing, Cham, 2017, pp. 85–113. doi:10.1007/978-3-319-38756-7_4.

[2] D. Groombridge, F. Karamouzis, Top strategic technology trends 2023, Gartner E-book: Available at: https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2023. Accessed on 11 (2022) 2022.

[3] T. Margaria, A. Schieweck, The digital thread in industry 4.0, in: Integrated Formal Methods: 15th International Conference, IFM 2019, Bergen, Norway, December 2–6, 2019, Proceedings 15, Springer International Publishing, 2019, pp. 3–24.

[4] C. Lehner, A. Padovano, C. Zehetner, G. Hackenberg, Digital twin and digital thread within the product lifecycle management, Procedia Computer Science 232 (2024) 2875–2886, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023). doi:https://doi.org/10.1016/j.procs.2024.02.104.

[5] A. Gartner, Gartner survey reveals digital twins are entering mainstream use, Gartner (2019).

[6] M. Brambilla, J. Cabot, M. Wimmer, Model-Driven Software Engineering in Practice, Second Edition, Morgan & Claypool Publishers, U.S.A., 2017.

[7] L. Burgueño, F. Ciccozzi, M. Famelis, G. Kappel, L. Lambers, S. Mosser, R. F. Paige, A. Pierantonio, A. Rensink, R. Salay, G. Taentzer, A. Vallecillo, M. Wimmer, Contents for a model-based software engineering body of knowledge, Software and Systems Modeling 18 (6) (2019) 3193–3205. doi:10.1007/s10270-019-00746-9.

[8] E. R. Batot, S. Gérard, J. Cabot, A Survey-driven Feature Model for Software Traceability Approaches, in: E. B. Johnsen, M. Wimmer (Eds.), Fundamental Approaches to Software Engineering, Springer International Publishing, Cham, 2022, pp. 23–48.

[9] C. Vergara, R. Bahsoon, G. Theodoropoulos, W. Yanez, N. Tziritas, Federated digital twin, in: 2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), IEEE, 2023, pp. 115–116.

[10] E. Ferko, A. Bucaioni, P. Pelliccione, M. Behnam, Analysing interoperability in digital twin software architectures for manufacturing, in: European conference on software architecture, Springer, 2023, pp. 170–188.

[11] E. Ferko, L. Berardinelli, A. Bucaioni, M. Behnam, M. Wimmer, Towards interoperable digital twins: Integrating sysml into aas with higher-order transformations, in: 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C), IEEE, 2024, pp. 342–349.

[12] L. Huang, L. Zhen, J. Wang, X. Zhang, Blockchain implementation for circular supply chain management: Evaluating critical success factors, Industrial Marketing Management 102 (2022) 451–464.

[13] A. M. Madni, S. Purohit, C. Madni, Exploiting digital twins in mbse to enhance system modeling and life cycle coverage, in: Handbook of Model-Based Systems Engineering, Springer, 2023, pp. 527–548.

[14] M. A. M. S. Lemstra, M. A. de Mesquita, Industry 4.0: a tertiary literature review, Technological Forecasting and Social Change 186 (2023) 122204.

[15] B. R. Barricelli, E. Casiraghi, D. Fogli, A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications, IEEE Access 7 (2019) 167653–167671. doi:10.1109/ACCESS.2019.2953499.

[16] S. Suhail, R. Hussain, R. Jurdak, A. Oracevic, K. Salah, C. S. Hong, R. Matulevičius, Blockchain-based digital twins: Research trends, issues, and future challenges, ACM Comput. Surv. 54 (11s) (Sep. 2022). doi:10.1145/3517189.

[17] E. Ferko, A. Bucaioni, M. Behnam, Architecting Digital Twins, IEEE Access 10 (2022) 50335–50350. doi:10.1109/ACCESS.2022.3172964.

[18] M. Perno, L. Hvam, A. Haug, Enablers and barriers to the implementation of digital twins in the process industry: A systematic literature review, in: 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2020, pp. 959–964. doi:10.1109/IEEM45057.2020.9309745.

[19] R. Minerva, G. M. Lee, N. Crespi, Digital twin in the iot context: A survey on technical features, scenarios, and architectural models, Proceedings of the IEEE 108 (10) (2020) 1785–1824. doi:10.1109/JPROC.2020.2998530.

[20] J. Michael, L. Cleophas, S. Zschaler, T. Clark, B. Combemale, T. Godfrey, D. E. Khelladi, V. Kulkarni, D. Lehner, B. Rumpe, et al., Model-driven engineering for digital twins: Opportunities and challenges, Systems Engineering (2025).

[21] D. Lehner, J. Zhang, J. Pfeiffer, S. Sint, A.-K. Splettstößer, M. Wimmer, A. Wortmann, Model-driven engineering for digital twins: a systematic mapping study, Software and Systems Modeling (2025) 1–39.

[22] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, Association for Computing Machinery, New York, NY, USA, 2014. doi:10.1145/2601248.2601268.

[23] L. Berardinelli, V. Muttillo, H. Bruneliere, R. Eramo, A. Bucaioni, B. Combemale, S. Ristov, P. Gritsch, D. Khelladi, Dao Khanh Duy, R. Hermawan, U. Yayan, A. Sadovykh, Replication package: Data, extraction forms, and analysis scripts for the MLR on traceability and federation in model-based DT engineering, https://github.com/jku-win-se/Barriers-and-Enablers-for-Traceability-and-Federation-in-Model-Based-Digital-Twin-Engineering, gitHub repository (2025).

[24] V. R. Basili, G. Caldiera, D. H. Rombach, The Goal Question Metric Approach, Vol. I, John Wiley & Sons, 1994.

[25] B. A. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report (07 2007).

[26] V. Garousi, M. Felderer, M. V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, Information and software technology 106 (2019) 101–121.

[27] M. Kasunic, Designing an effective survey, Tech. rep. (2005).

[28] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Computer Science, 2012.

[29] Wooclap, https://www.wooclap.com/ (2026).

[30] J. W. Creswell, V. L. Plano Clark, Designing and Conducting Mixed Methods Research, 3rd Edition, SAGE Publications, Thousand Oaks, CA, 2017.

[31] D. S. Cruzes, T. Dyba, Recommended steps for thematic synthesis in software engineering, in: Procs of ESEM, IEEE, 2011, pp. 275–284.

[32] R. Franzosi, Quantitative narrative analysis, 2010.

[33] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, J. Popay, Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: effectiveness of interventions to promote smoke alarm ownership and function, Evaluation (2009).

[34] D. T. Heber, M. W. Groll, How the blockchain fosters e/e traceability for mbse and plm in distributed engineering collaboration, in: 2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), 2018, pp. 125–130. doi:10.1109/ITMC.2018.8691219.

[35] Y. Zhang, S. Ma, H. Yang, J. Lv, Y. Liu, A big data driven analytical framework for energy-intensive manufacturing industries, Journal of Cleaner Production 197 (2018) 57–72. doi:https://doi.org/10.1016/j.jclepro.2018.06.170.

[36] R. Drath, A. Luder, J. Peschke, L. Hundt, Automationml - the glue for seamless automation engineering, in: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, 2008, pp. 616–623. doi:10.1109/ETFA.2008.4638461.

[37] M.-G. Back, D.-K. Lee, J.-G. Shin, J.-H. Woo, A study for production simulation model generation system based on data model at a shipyard, International Journal of Naval Architecture and Ocean Engineering 8 (5) (2016) 496–510. doi:https://doi.org/10.1016/j.ijnaoe.2016.05.005.

[38] M. Díaz-Madroñero, J. Mula, D. Peidro, A mathematical programming model for integrating production and procurement transport decisions, Applied Mathematical Modelling 52 (2017) 527–543. doi:https://doi.org/10.1016/j.apm.2017.08.009.

[39] E. Francalanza, J. Borg, C. Constantinescu, A knowledge-based tool for designing cyber physical production systems, Computers in Industry 84 (2017) 39–58. doi:https://doi.org/10.1016/j.compind.2016.08.001.

[40] D. Chen, D. V. Panfilenko, M. R. Khabbazi, D. Sonntag, A model-based approach to qualified process automation for anomaly detection and treatment, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016, pp. 1–8. doi:10.1109/ETFA.2016.7733731.

[41] Y. Lu, Industry 4.0: A survey on technologies, applications and open research issues, Journal of Industrial Information Integration 6 (2017) 1–10. doi:https://doi.org/10.1016/j.jii.2017.04.005.

[42] L. Lúcio, M. Amrani, J. Dingel, L. Lambers, R. Salay, G. M. K. Selim, E. Syriani, M. Wimmer, Model transformation intents and their properties, Software and Systems Modeling 15 (3) (2016) 647–684. doi:10.1007/s10270-014-0429-x.

[43] E. Y. Song, M. Burns, A. Pandey, T. Roth, Ieee 1451 smart sensor digital twin federation for iot/cps research, in: 2019 IEEE Sensors Applications Symposium (SAS), 2019, pp. 1–6. doi:10.1109/SAS.2019.8706111.

[44] S. Vural, R. Minerva, G. A. Carella, A. M. Medhat, L. Tomasini, S. Pizzimenti, B. Riemer, U. Stravato, Performance measurements of network service deployment on a federated and orchestrated virtualisation platform for 5g experimentation, in: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2018, pp. 1–6. doi:10.1109/NFV-SDN.2018.8725755.

[45] B. Hasan, J. Wikander, A review on utilizing ontological approaches in integrating assembly design and assembly process planning (app), International Journal of Mechanical Engineering 4 (11) (2017) 5–16. doi:10.14445/23488360/ijme-v4i11p102.

[46] M. S. Elapolu, R. Rai, D. J. Gorsich, D. Rizzo, S. Rapp, M. P. Castanier, Blockchain technology for requirement traceability in systems engineering, Information Systems 123 (2024) 102384. doi:https://doi.org/10.1016/j.is.2024.102384.

[47] R. Eramo, F. Marchand de Kerchove, M. Colange, M. Tucci, J. Ouy, H. Bruneliere, D. Di Ruscio, Model-driven design-runtime interaction in safety critical system development: an experience report, Journal of Object Technology 18 (2) (2019) 1:1–22. doi:10.5381/jot.2019.18.2.a1.

[48] D. Arcelli, V. Cortellessa, D. Di Pompeo, R. Eramo, M. Tucci, Exploiting architecture/runtime model-driven traceability for performance improvement, in: 2019 IEEE International Conference on Software Architecture (ICSA), 2019, pp. 81–90. doi:10.1109/ICSA.2019.00017.

[49] F. Bordeleau, B. Combemale, R. Eramo, M. van den Brand, M. Wimmer, Towards model-driven digital twin engineering: Current opportunities and future challenges, in: Systems Modelling and Management, Springer International Publishing, Cham, 2020, pp. 43–54.

[50] N. Kahani, M. Bagherzadeh, J. R. Cordy, J. Dingel, D. Varró, Survey and classification of model transformation tools, Software & Systems Modeling 18 (2019) 2361–2397.

[51] H. Bruneliere, E. Burger, J. Cabot, M. Wimmer, A feature-based survey of model view approaches, Software & Systems Modeling 18 (2019) 1931–1952.

[52] F. Ciccozzi, I. Malavolta, B. Selic, Execution of uml models: a systematic review of research and practice, Software & Systems Modeling 18 (2019) 2313–2360.

[53] T. Cook, D. Campbell, Quasi-Experimentation: Design and Analysis Issues for Field Settings, Houghton Mifflin, 1979.

Selected Peer-reviewed Literature

[SP01] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, “Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, may 2020, https://doi.org/10.1016/j.cirpj.2020.02.002.

[SP02] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019, https://doi.org/10.1109/ACCESS.2019.2953499.

[SP03] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 108, no. 10, pp. 1785–1824, oct 2020, https://doi.org/10.1109/JPROC.2020.2998530.

[SP04] K. Y. H. Lim, P. Zheng, and C.-H. Chen, "A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives," *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1313–1337, aug 2020, http://dx.doi.org/10.1007/s10845-019-01512-w.

[SP05] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat, and S. Bakiras, "The role of AI, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities," *IEEE Access*, vol. 9, pp. 32 030–32 052, 2021, https://doi.org/10.1109/ACCESS.2021.3060863.

[SP06] A. Wortmann, O. Barais, B. Combemale, and M. Wimmer, "Modeling languages in industry 4.0: an extended systematic mapping study," *Softw. Syst. Model.*, vol. 19, no. 1, pp. 67–94, jan 2020, https://doi.org/10.1007/s10270-019-00757-6.

[SP07] O. C. Madubuike, C. J. Anumba, and R. Khallaf, "A review of digital twin applications in construction," *J. Inf. Technol. Constr.*, vol. 27, pp. 145–172, feb 2022, https://doi.org/10.36680/j.itcon.2022.008.

[SP08] P. Brauner, M. Dalibor, M. Jarke, I. Kunze, I. Koren, G. Lakemeyer, M. Liebenberg, J. Michael, J. Pennekamp, C. Quix, B. Rumpe, W. van der Aalst, K. Wehrle, A. Wortmann, and M. Ziefle, "A computer science perspective on digital transformation in production," *ACM Trans. Internet Things*, vol. 3, no. 2, pp. 1–32, may 2022, https://doi.org/10.1145/3502265.

[SP09] E. Ferko, A. Bucaioni, and M. Behnam, "Architecting digital twins," *IEEE Access*, vol. 10, pp. 50 335–50 350, 2022, https://doi.org/10.1109/ACCESS.2022.3172964.

[SP10] E. Ferko, A. Bucaioni, P. Pelliccione, and M. Behnam, "Standardisation in digital twin architectures in manufacturing," in *2023 IEEE 20th International Conference on Software Architecture (ICSA)*. IEEE, mar 2023, https://doi.org/10.1109/ICSA56044.2023.00015.

[SP11] A. Siriweera and K. Naruse, "Survey on cloud robotics architecture and model-driven reference architecture for decentralized multicloud heterogeneous-robotics platform," *IEEE Access*, vol. 9, pp. 40 521–40 539, 2021, https://doi.org/10.1109/ACCESS.2021.3064192.

[SP12] L. Merkle, "Cloud-based battery digital twin middleware using model-based development," in *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*. New York, NY, USA: ACM, sep 2019, https://doi.org/10.1145/3386164.3387296.

[SP13] Z. Li, J. Lu, G. Wang, L. Feng, D. G. Broo, and D. Kiritsis, "A bibliometric analysis on model-based systems engineering," in *2021 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, sep 2021, https://doi.org/10.1109/ISSE51541.2021.9582526.

[SP14] M. A. Guinea-Cabrera and J. A. Holgado-Terriza, "Digital twins in software engineering—a systematic literature review and vision," *Appl. Sci. (Basel)*, vol. 14, no. 3, p. 977, jan 2024, https://doi.org/10.3390/app14030977.

[SP15] I. A. Arin, Meyliana, H. L. H. S. Warnars, and D. F. Murad, "A systematic literature review of recent trends and challenges in digital twin implementation," in *2023 10th International Conference on ICT for Smart Society (ICISS)*. IEEE, sep 2023, https://doi.org/10.1109/ICISS59129.2023.10291219.

[SP16] E. Papadonikolaki and C. Anumba, "How can digital twins support the net zero vision?" in *Lecture Notes in Civil Engineering*. Cham: Springer Nature Switzerland, 2023, pp. 81–97, https://doi.org/10.1007/978-3-031-32515-1_7.

[SP17] C.-R. Raitviir and I. Lill, "Conceptual framework of information flow synchronization throughout the building lifecycle," *Buildings*, vol. 14, no. 7, p. 2207, jul 2024, https://doi.org/10.3390/buildings14072207.

[SP18] W. Li, B. Li, H. Gao, C. Peng, W. Zhang, Y. Li, and X. Ji, "Survey and practice on architecture and deployment method of digital twin system for intelligent substation," in *2023 7th International Symposium on Computer Science and Intelligent Control (ISCSIC)*. IEEE, oct 2023, https://doi.org/10.1109/ISCSIC60498.2023.00046.

[SP19] M. Zahid, A. Bucaioni, and F. Flammini, "Model-based trustworthiness evaluation of autonomous Cyber-Physical production systems: A systematic mapping study," *ACM Comput. Surv.*, vol. 56, no. 6, pp. 1–28, jun 2024, https://doi.org/10.1145/3640314.

[SP20] S. Werbińska-Wojciechowska, R. Giel, and K. Winiarska, "Digital twin approach for operation and maintenance of transportation system-systematic review," *Sensors (Basel)*, vol. 24, no. 18, p. 6069, sep 2024, https://doi.org/10.3390/s24186069.

Selected Gray Literature

[SG01] Virtual Twin for Manufacturing, DELMIA, https://www.3ds.com/products/delmia/virtual-twin-manufacturing.

[SG02] Model-Based Systems Engineering to help navigate Digital Transformation, https://www.zuken.com/it/resource/model-based-systems-engineering-to-help-navigate-digital-transformation/.

[SG03] Let's Talk About the Model-Based Enterprise Digital Threads, https://aras.com/en/blog/let-s-talk-about-the-model-based-enterprise-digital-threads-and-relationships.

[SG04] Defence Digital Twin Implementation Road Map and Development Framework, https://www.teamdefence.info/wp-content/uploads/2022/03/20210121-Digital-Twin-Implementation-Road-Map-and-Development-Framework-White-Paper-V1.pdf.

[SG05] Digital Twin Core Conceptual Models and Services, https://www.iiconsortium.org/wp-content/uploads/sites/2/2023/10/Digital-Twin-Core-Conceptual-Models-and-Services_20231102.pdf.

[SG06] Adoption View, https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Adoption%20View%20Digital%20Twin%20Kit/.

[SG07] The TRICK to Digital Twin Technology Adoption, https://altair.com/blog/executive-insights/the-trick-to-digital-twin-technology-adoption.

[SG08] Digital Twin System Interoperability Framework, https://www.digitaltwinconsortium.org/pdf/Digital-Twin-System-Interoperability-Framework-12072021.pdf.

[SG09] How Can Digital Twins Be Used in Systems Engineering?, https://blog.fenstermaker.com/how-can-digital-twins-be-used-in-systems-engineering/.

[SG10] Creating a new "Digital Twin Model", https://documentation.mindsphere.io/MindSphere/apps/factory-twin/creating-new-digital-twin-model.html.

[SG11] How do digital threads & digital twins fit in MBSE?, https://www.analogictips.com/how-do-digital-threads-digital-twins-fit-in-mbse-faq/.

[SG12] Digital Risk Twin, https://www.phmtechnology.com/digital-risk-twin.html.

[SG13] The Digital Thread - Learn More Today, https://www.zuken.com/us/product/digital-engineering/digital-thread/.

[SG14] Digital Twin, https://www.technia.com/digital-twin/.

[SG15] DT FAQ: Digital Twin & Digital Engineering FAQ, https://digitaltwinworks.com/faq/.

[SG16] JuliaSim: Next-Gen Modeling, https://juliahub.com/products/juliasim.

[SG17] Transcality – Digital twin for efficient mobility, https://transcality.com/.

[SG18] Using model-based systems engineering to develop a case study: Mobiledrive, https://resources.sw.siemens.com/en-US/case-study-mobiledrive/.

[SG19] **BROKEN:** Network Digital Twin Use Case, https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-NDT_Use_Case-2.0.pdf.

[SG20] Model-Based Embedded Software Design & Development, https://cadfem.ai/engineering-services/embedded-software/.

[SG21] TRICK: An Easy Acronym to Conceptualize Digital Twin, https://www.automation.com/en-us/articles/february-2024/trick-easy-acronym-conceptualize-digital-twin.

[SG22] The Digital Twin in the Network and Electricity Industry, https://www.vde.com/resource/blob/2293228/3e50b3417c8777c85e9e33cb3e1e9ebc/vde-study-the-digital-twin-in-the-network-and-electricity-industry-data.pdf.

[SG23] Operational Digital Twin for Semiconductor Manufacturing, https://srcmapt.org/wp-content/uploads/2024/07/MAPT-Deep-Dive-on-Digital-Twins-PDF-Solutions-May-31-2024-V2-Final.pdf.

[SG24] SystemC and Digital Twin: Good match or Not?, https://workspace.accellera.org/document/dl/10951.

[SG25] OPC UA Designer, https://eclipse.dev/papyrus/components/manufacturing/resources/Papyrus4Manufacturing-OPCUA-Designer.pdf.

[SG26] Navigating the transition to a model-based enterprise, https://www2.deloitte.com/content/dam/Deloitte/us/Documents/process-and-operations/mbe-navigating-the-transition.pdf.

[SG27] THE ROLE OF HIGH-FIDELITY DIGITAL TWINS, https://www.oecd-nea.org/upload/docs/application/pdf/2021-08/dt_workshop_-_presentations_panel_3.pdf.

[SG28] 2x faster vehicle software development proven by digital twin, https://soda.auto/news/soda-and-vsoptima-digital-twin-wp.pdf.

[SG29] Driving Productivity and Quality Efficiencies with Model-based Digitalization, https://www.cyient.com/blog/driving-productivity-and-quality-efficiencies-with-model-based-digitalization.

[SG30] Model-Based Systems Engineering, https://eda.sw.siemens.com/en-US/pcb/model-based-systems-engineering/.

[SG31] Digital Twin Lessons for Engineers from the PLM Road Map, https://www.engineering.com/digital-twin-lessons-for-engineers-from-the-plm-road-map-pdt/.

[SG32] Using VeSyMA in MBSE, https://www.claytex.com/tech-blog/using-vesyma-in-mbse/.

[SG33] Integrated Model-based Systems Engineering (iMBSE) in Engineering Education, https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2021:iw2021_integrated_mbse_in_engineering_education.pdf.

[SG34] Digital Twin Solutions, https://www.boozallen.com/expertise/digital-twin-solutions.html.

[SG35] Digital Thread & Digital Twin Integration Solutions, https://prostep.us/cpmn/apidt/digital-thread-and-digital-twin-solutions/.

[SG36] Digital Twin Enabling Technology Catalogue, https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ddabb8bf&appId=PPGMS.

[SG37] A Model-Based Systems Engineering (MBSE) Approach to the Design Optimization of Phased Array Antenna Systems, https://www.phoenix-int.com/wp-content/uploads/2021/03/MBSE-Webinar-_-A-MBSE-Approach-to-the-Design-Optimization-of-Phased-Array-Antenna-Systems-_-NORTHROP-GRUMMAN.pdf.

[SG38] A Guide to Digital Twin Development, https://www.visartech.com/blog/digital-twin-solution-development-guide/.

[SG39] Digital Twin - DAIR - Acquisition Research Program, https://dair.nps.edu/bitstream/123456789/4777/3/NPS-SE-22-263.pdf.

[SG40] Overview - IMOCO4.E, https://www.imoco4e.eu/overview/about-imoco4-e/.

[SG41] What is Model-Based Systems Engineering (MBSE)?, https://visuresolutions.com/mbse-guide/what-is-mbse/.

[SG42] DT FAQ: What are the goals of Digital Twins?, https://digitaltwinworks.com/faq/what-are-goals-digital-engineering.html.

[SG43] Digital Threads: The Future of Compliance, https://www.leancompliance.ca/post/digital-threads-the-future-of-compliance.

[SG44] Digital Engineering - National Reactor Innovation Center, https://nric.inl.gov/wp-content/uploads/2021/03/NRIC-TT-Digital-Engineering-web.pdf.

[SG45] What Is Model-Based Systems Engineering (MBSE)?, https://aras.com/en/glossary/mbse.

[SG46] Model-Based Practices – Systems Engineering Vision 2035, https://violin-strawberry-9kms.squarespace.com/model-based-practices.

[SG47] Digital Twin - Dr Rajiv Desai, https://drrajivdesaimd.com/2023/03/12/digital-twin/.

[SG48] Virtual Stress-Testing a Digital Twin of a Motor Control Platform using Dynamic Fault Injection, https://valu3s.eu/virtual-stress-testing-a-digital-twin-of-a-motor-control-platform-using-dynamic-fault-injection/.

[SG49] Future Model-Based Systems Engineering Vision and ..., https://ntrs.nasa.gov/api/citations/20210014025/downloads/TM-20210014025.pdf.

[SG50] Learn How to Use the Bill of Characteristics BoC as Your Guide, https://www.youtube.com/watch?v=PnLPqjNBqLE.

[SG51] The Perfect Pair: Digital Twins and Predictive Maintenance, https://www.infoq.com/articles/digital-twin-predictive-maintenance/.

[SG52] Connecting Tracing and Managing the Lifecycle of Models, https://pdteurope.com/wp-content/uploads/2020/11/Ferrari_Final_Distribution_2020.pdf.

[SG53] Digital twins from design to handover of constructed assets, https://www.rics.org/content/dam/ricsglobal/documents/research/digital-twins-from-design-to-handover-of-constructed-assets.pdf.

[SG54] Digital Product Traceability in Automotive, https://www.ptc.com/en/resources/liveworx-on-demand/digital-product-traceability-in-automotive.

[SG55] How to create traceability system in manufacturing? Reduce ..., https://www.youtube.com/watch?v=WFdEC6fb5JA.

[SG56] Model Based Engineering @ Boeing, https://pdteurope.com/wp-content/uploads/2020/11/Plant_Final_Distribution_2020.pdf.