An Evaluation Method for Digital Twin Development Platforms

José Monteiro[1, 2][0000-0002-9000-9660] and João Barata[1][0000-0002-7456-594X]

[1] University of Coimbra, CISUC, DEI, Coimbra, Portugal
[2] Polytechnic of Coimbra, ESTGOH, 3400-124 Oliveira do Hospital, Portugal
{jafmonteiro, barata}@dei.uc.pt

Abstract. The Digital Twin (DT) offers an integrated solution for replicating physical (human and non-human) systems with monitoring capabilities and intelligent support for decision-making. Their popularity in academia and industry is growing, and different commercial and open-source development platforms are now available. However, there is a lack of detailed platform benchmarking studies and selection guidelines. This paper (1) identifies a portfolio of DT development platforms (DTDP) and (2) suggests a systematic method to evaluate them. Preliminary results of the method adoption are presented for a use case of a dry port DT deployment. This research will assist companies with their DTDP investments, presenting an assessment example for more complex DT deployment settings.

Keywords: Digital Twin, Digital Twin Development Platform, DT Platforms Benchmarking.

1 Introduction

Digital twins (DTs) are replicas of physical objects or systems, enabling synchronized bidirectional flows of information [2, 3]. Their origins were aimed at representing specific objects like single machines, but the context is drastically changing. Concepts like the organizational DT representing human practices are now a priority [1], and their application in industry and supply chains is rapidly expanding [2, 3].

DT design is a complex endeavor requiring multiple technologies. Therefore, multiple frameworks or development platforms have been built, both commercial and free/libre and open source (FLOSS). Interestingly, our literature review only found one study comparing these platforms with a prototype, restricted to the FLOSS segment [4]. Moreover, selecting the most suitable DT development platform by companies is challenging, and structured methods to evaluate them are missing.

Citation: Monteiro, J., Barata, J. (2024). An Evaluation Method for Digital Twin Development Platforms. In: De Paolis, L.T., Arpaia, P., Sacco, M. (eds) Extended Reality. XR Salento 2024. Lecture Notes in Computer Science, vol 15027. Springer, Cham.

The motivation for conducting this study started in a research center with an increasing number of DT research projects. The lack of guidance in platform selection is ineffective because a significant amount of time in DT design and development projects is spent testing platforms, or, in contrast, favoring the researchers/practitioners' experience with IT platforms and languages, lacking a clear justification for the choice. A roadmap for DT development platform selection is the main research aim, with two complementary research objectives (RO):

- *RO1: Identify open-source and commercial DT development platforms;*
- *RO2: Propose an evaluation method for DT development platforms.*

The remainder of the article is structured as follows. Section 2 presents key concepts on DT development platforms and evaluation criteria. Subsequently, the method for identifying and evaluating DT development platforms is proposed. Next, Section 4 describes preliminary results, including a use case instantiation and evaluation. The paper closes in Section 5, summarizing the conclusions, limitations, and next steps.

2 Background

2.1 Platforms for Digital Twin Development

DT designers have multiple options available, commercial and open source. Open-source solutions are more popular in the literature and have already been compared in a recent study [4]. The authors identified Eclipse Ditto, Equinox, AASX Package Explorer, PyI40AAS, SAP I4.0 AAS, Eclipse BaSyx, NOVA AAS, CPS-Twinning, Twined, Azure Digital Twins Definition Language, iTwin.js, Digital Twin Cities Centre Platform (DTCC), TerriaJS (NSW Digital Twin implementation), and INTO-CPS Co-simulation Framework (Github links for each one available in [4]).

Solutions like iTwin.js, for example, are specifically developed for 3D objects, but requiring more time and code to implement the solution (contrasting with Eclipse BaSyx, Eclipse Ditto, or SAP I4.0 AAS that do not provide real-time visual presentations). Nevertheless, other solutions are restricted to the communication layer (e.g., sensors). Despite not including other prominent solutions like Unity Digital Twins or commercial applications like Twinzo or Azure Digital Twins, the work presented by [12] is inspirational, and the limitation presented by these authors (not including commercial DT development platforms) is a starting point for additional research.

Our literature review found other studies that identified DT development platforms. For example, in construction, several BIM-related implementations use 3D Fem, Three.js, Draco 3D model, Unity 3D, Solidworks, 3D Max, Rhinoceros, or Autodesk Revit for digital modeling layers [5]. Yet, these solutions alone are insufficient to create a DT, requiring other technologies for the data acquisition and transmission. Another recent example identified both commercial and open source solutions, adding to our list Eclipse Vorto, Eclipse BaSyx, ScaleOut, Davra Platform, Fiware, Open Digital Twin, or commercial solutions like Oracle DT, IBM DT, AWS IoT TwinMaker, Bosch IoT Suite, XMPro, ScaleOut or Nvidia Omniverse [6]. Other works relevant to identifying DT development platforms include [7] and [8] comparing Microsoft Azure, Amazon Web Services, and Eclipse ecosystem. However, these examples did not use a specific use case for evaluation, focusing on assessing more ample criteria of each platform (e.g., automation protocols, interoperability, type – commercial or open source, or mobile support). This section summarizes the results for RO1.

2.2 Digital Twin Development Platform Evaluation

Some criteria recently used for evaluating DT development platforms include (P) platform-related and (U) use case-related criteria. Table 1 summarizes the most relevant [4–6, 8].

Table 1. Criteria used for DT development platforms evaluation.

Type	Criteria	Explanation
P	Communication/synchronization	Possibility to implement a physical model (only digital replica) or more advanced digital shadow (communication from the physical to the digital layer, the reverse, or full DT implementation (bi-directional) [4]
	Data storage	Tools available to store data in the platform in third-party databases [4]
	Development pipelines	Possibility to reuse models/code, support for continuous integration/continuous deployment to support changes [8]
	Connection and data security	Capacity to support physical interventions and fault tolerance [4]
	Business model	Commercial or open source [6]
	Scalability	Support for multiple connections or DT instances [4, 6]
	Standardization	The standards adopted by the platform for the different layers [4]
	User support	Guidance provided for developers, and documentation available [4]
	Latency	Optimization of data processing and communication [6]
	Mobility	Support for mobile applications [6]
	Robustness	Reliability measures [6]
	Interoperability	Support for syntactic and semantic exchange, cloud-edge interaction, automation protocols, system and platform interoperability [4, 6, 8]
	Context-aware	Capacity to interpret the data collection context [6]
	Geo-distribution	Allows distributed DTs and orchestration capabilities [6]
	Data processing and analytics	Possibility to explore data, via API, or integrated features like machine learning [4, 5]
	Market image	The companies using the platform, visibility, confidence in long-term support [4]

Type	Criteria	Explanation
	Hosting model	Cloud or on-premise; installation requirements [8]
	Development skills	Some more specific platforms and languages require strong programming knowledge, while others have support for specialists in the application context [8]
U	Virtual modeling	Structure and visualization dimensions allowed us to represent the DT and the existence of a modeling layer [5]
	User services	This criterion can be divided into five main capabilities, namely, near real-time monitoring and alerts, anomaly detection, prediction capabilities, simulation capabilities, and physical controller – capacity for physical intervention [4, 5]
	Fleet	Possibility of DT aggregation; single DT development or multiple (e.g., hierarchical link) [4, 8]

Legend: (P) Platform-related; (U) Use case-related

Table 1 presents a comprehensive selection of criteria for evaluating DT development platforms. They are aligned with papers suggesting traditional features to compare DT instances [9]. The majority are related to the platform, while others reach a higher level of detail, assessing the DTs produced by those platforms (products) in terms of (1) visual presentation, (2) services, and (3) aggregation complexity. Some criteria are common to different studies, but others only appear in specific analyses. For example, [17] compared the metamodel of AWS IoT TwinMaker, Azure DT, and Eclipse (the most popular), but no use case was implemented.

The research team also looked for studies comparing other types of development platforms. For example, in the field service area, [10] evaluates a mix of platform-related (e.g., processing time, loading time, analytics capabilities, data records, API interoperability) and domain-specific criteria (e.g., GPS integration). Another excellent example of low-code platform comparison adopted several criteria identified in Table 1 (e.g., scalability, security, interoperability) and two others that also seem interesting to the scope of DT: collaborative work support for developers and graphical tools for programming (e.g., workflows). An early study in mobile apps shares many of the most recent criteria, adding time to market, potential users, and costs [11]. Interestingly, some usability aspects, like effort measurement for routine tasks [12], were not found in related works.

The most usual criteria assessment is in a binary form (yes/no) or a three-value scale with 0-inexistent, 1-partial, and 2-full. It is found in DT studies [6, 8] and other contexts [13, 14]. However, a more detailed scale (e.g., with levels) is also possible, where each option represents a specific degree of accomplishing each criterion [4]. Although not all studies comparing platforms explain their methods explicitly, we consider this an essential element, as explained in the next section.

3 Method

The sequence of steps proposed in [13] starts with use case selection, which is an interesting option for designing more complex DTs. The second step is data collection to support the use case, followed by coding and debugging, comparison using default parameters in each platform, model validation, and output comparison. A different starting point was selected by [4], with the identification of candidate DT frameworks. However, these authors clearly explain which ones were discarded (e.g., lack of documentation) and their need for a taxonomy. The case study was selected afterward, opting for a simple DT object. The implementation of each platform and evaluation of case study requirements completes their approach. Contrasting to merely evaluating platform-related criteria, a use case allows a deeper understanding of the implementation [14]. For example, explaining how it was implemented for each criterion and a richer presentation of details beyond the mere yes/no assessment.

Fig. 1 depicts the proposed method for DT development platform evaluation.

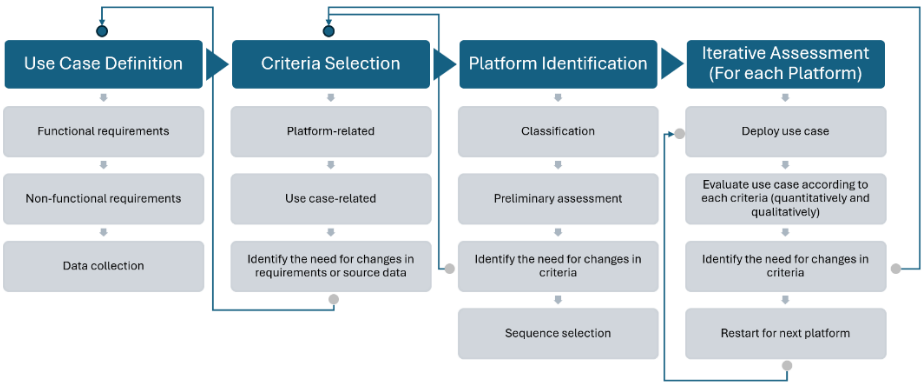

Fig. 1. A method for DT development platforms evaluation.

The first step of the proposed method is the use case definition, which requires capturing the social and fleet elements. This stage also includes establishing what the DT should do (functional), quality attributes, and data sources (real or synthetic data) required for development.

Based on a literature review, the criteria selection followed (presented in Section 2.3). We found advantages in this sequence to ensure unbiased inclusion of all relevant criteria, not looking at the selected platforms. Some criteria may require an additional specification of the use case. Therefore, our method is iterative and allows returning to previous steps. Based on a comprehensive list of candidate platforms, selecting which will pass to the deployment phase is necessary. In our study, we decided to start with commercial solutions, extending the work of [4]. Due to its popularity in the literature, Azure Digital Twins was the first. The subsequent stage of iterative deployment and assessment aims to continuously refine the assessment process (criteria may be added at this stage if the specific deployment justifies the choice), finalizing the process with a cross-platform evaluation and the proposal of a selection guide.

4 Preliminary Results

4.1 Use Case Requirements and Data Collection

Our research team adopted an agile DT development approach, starting with the identification of user stories (user – requirement – purpose structure). The DT context used for the selection is a dry port DT implementation included in a co-funded project for green and digital transformation in logistics. Examples for each type (platform-related or use case-related) are included in Table 2.

Table 2. Use stories for the DT prototype (extract).

User story	Related criteria
As a *team manager* I want to *obtain advice on how to allocate the workers to the most suitable tasks* so that *they feel more satisfied in the workplace and with more productivity*	User services (human-centric)
As a *developer* I want to *include digital twins in a wide area of operation (e.g., region, supply chain)* so that *I can monitor different stages of the port activities*	Geo-distribution
As a *developer* I want to *integrate machine learning models* so that *I can predict events and simulate alternatives*	Data processing and analytics

The first user story example in Table 2 is use-case-related (the users of the resulting DT), and the other two are platform-related (through the developers' lenses). The team iteratively created a complete list of user stories addressing the needs of the selected use case, following the framework presented in Fig. 1 for each criterion presented in Table 1. Our priority sequence followed the commercial offer, starting with Azure Digital Twins because of its popularity. Future work will include other DT development platforms identified in the literature review.

4.2 Assessment

The assessment template developed for the final stage and the results for each criterion are presented in Table 3, for the platform Azure Digital Twins.

Table 3. Assessment artifact for digital twin development platforms (extract).

Criteria	**E**	**How it was assessed**	**Evidence**
Communication/ synchronization	F	Analyzed based on bi-directional communication capability between physical and digital environments	Support for DT implementations with real-time synchronization

Criteria	E	How it was assessed	Evidence
Data storage	P	Third-party database integration explored	Native storage capabilities built in, but less advanced support for external databases
Development pipelines	F	Support for CI/CD practices verified	Full support for model and code reuse and CI/CD
Connection and data security	F	Security assessment and fault tolerance	Support for physical intrusion and data security
Business model	F	Based on the commercial availability of the platform	Commercial model, not open source
Scalability	F	Ability to support multiple instances and connections	Documented support for broad scalability
Standardization	P	Review of standards adopted by the platform	Adopts some industry standards but lacks universality across tiers
User support	F	Quality of support and documentation	Excellent documentation and developer support
Latency	F	Optimization in data processing and communications	Well-documented low latency capabilities
Mobility	P	Support for mobile applications	Some support for mobility, but not the primary focus
Robustness	F	Reliability measures implemented	Relevant resources and reliability measures
Interoperability	F	Syntactic and semantic interaction support	Broad interoperability with multiple platforms and systems
Context-aware	F	Ability to interpret the context of data collection	Capabilities to understand and act in the context of data
Geo-distribution	P	Ability to distribute DTs geographically	Supports some distribution, but with orchestration limitations
Data processing and analytics	F	Data processing and analysis capabilities	Support for data analysis and integration with machine learning
Market image	F	Reputation in the market and use by well-known companies	Recognized platform with market acceptance
Hosting model	F	Cloud and on-premises hosting options	Flexibility with cloud and on-premises hosting options
Development skills	P	Specific programming skills	Requires development skills but provides contextual support
Virtual modeling	F	DT modeling and visualization support	Modeling and visualization capabilities
User services	F	Provides monitoring, anomaly detection, and physical intervention capabilities	Support for user services with various integrated capabilities
Fleet	P	Support for the development and aggregation of DTs	Some ability to develop and manage multiple DTs, but limitations in hierarchical relationships

Criteria	E	How it was assessed	Evidence
Meta model	F	Review of the platform's ability to define and use meta models	The platform allows the definition of abstract models that specify the structure and relationships of DTs, facilitating the creation and management of complex twins
Support for developer collaboration	P	Analysis of the collaboration tools available to developers	Azure DT supports integration with other Azure collaboration tools but may not provide specific collaboration capabilities within the DT development platform itself

Legend: E (Evaluation): F-Full; P-Partial; N/A-No Support

Table 3 includes the assessment (second column) for each criterion (first column) previously identified in our method, for the use case deployed in Azure Digital Twins (and a template for subsequent evaluations). Additionally, we add an explanation of how it was done and evidence to support our decision based on the deployed use case.

Overall, the Azure Digital Twins platform fully covers 69.57% of the evaluation criteria we defined, while the remaining 30.43% are partially covered. The selected platform has good documentation and tutorials, which are essential during the instantiation phase, but two drawbacks were identified. First is the user interaction; while mechanisms allow the user to change the values of a given attribute, this action is done as if it were "emulating" the reading of a sensor. The second is related to integrating the 3D scenes into an external web application. These examples reinforce the importance of use-case evaluation strategies for DT development platforms.

5 Conclusion and Outlook

This study presented a systematic method to evaluate DT development platforms, integrating platforms, use cases, and contextual criteria. The detailed, use-case-based method provides a model for future studies and promotes a deeper understanding of the capabilities and limitations of available platforms. For practitioners, the results of this work are valuable for professionals seeking to develop systems that utilize DT technology, providing clear and insightful guidelines for selecting platforms.

Several limitations must be stated. This is a work in progress, and only preliminary results have been reported for the first commercial platform, which may not represent all available options. This evaluation will be extended to a wider range of platforms, and the method will be validated in real DT project implementations. Nevertheless, our work presents a starting point for evaluating DT development platforms, which will be a top priority in the digital transformation agenda.

Several avenues for future research are suggested. First, we will continue the iterative assessment with more platforms according to the defined priorities. We also aim to create additional assessment criteria. Although we found this use case suitable for the initial stage of our work, it does not yet capture human practices (e.g., using computer vision or wearable technology) in detail. We also plan to develop more detailed comparative studies using a variety of complex use cases (e.g., smart terminals, smart factories, smart cities) to validate and refine our evaluation criteria. In addition, we will explore the integration of our method with software validation techniques, such as automated testing and artificial intelligence frameworks, to improve the accuracy and scalability of the evaluations. It will also be important to create a selection guide for DT deployment scenarios, assisting developers in selecting the best platform. Finally, our work explores commercial platforms, complementing the work of [4] for open source. Nevertheless, it will also be important to evaluate both simultaneously, contributing to FLOSS market development and eventually extending commercial platforms with FLOSS integration that can solve some of their deficiencies, particularly in settings of human-centric DTs, not restricted to a single physical object.

Acknowledgements

This project was funded by Project "Agenda Mobilizadora Sines Nexus". ref. No. 7113, supported by the Recovery and Resilience Plan (PRR) and by the European Funds Next Generation EU, following Notice No. 02/C05-i01/2022, Component 5 - Capitalization and Business Innovation - Mobilizing Agendas for Business Innovation. This work is also funded by the FCT - Foundation for Science and Technology, I.P./MCTES through national funds (PIDDAC), within the scope of CISUC R&D Unit - UIDB/00326/2020 or project code UIDP/00326/2020. The first author is funded by FCT under the Ph.D. grant 2022.09950.BD.

References

1. Rantala, T., Ukko, J., Nasiri, M., Saunila, M.: Shifting focus of value creation through industrial digital twins—From internal application to ecosystem-level utilization. Technovation. 125, 102795 (2023).
2. Barata, J., Kayser, I.: How will the digital twin shape the future of industry 5.0? Technovation. 134, (2024).
3. Tu, X., Autiosalo, J., Ala-Laurinaho, R., Yang, C., Salminen, P., Tammi, K.: TwinXR: Method for using digital twin descriptions in industrial eXtended reality applications. Front. Virtual Real. 4, 1–14 (2023).
4. Gil, S., Mikkelsen, P.H., Gomes, C., Larsen, P.G.: Survey on open-source digital twin frameworks–A case study approach. Softw. - Pract. Exp. 929–960 (2024).
5. Tuhaise, V.V., Tah, J.H.M., Abanda, F.H.: Technologies for digital twin applications in construction. Autom. Constr. 152, 104931 (2023).
6. Hakiri, A., Gokhale, A., Yahia, S. Ben, Mellouli, N.: A comprehensive survey on digital twin for future networks and emerging Internet of Things industry. Comput. Networks. 244, 110350 (2024).
7. Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., Nee, A.Y.C.: Enabling technologies and tools for digital twin. J. Manuf. Syst. 58, 3–21 (2021).
8. Lehner, D., Pfeiffer, J., Tinsel, E.F., Strljic, M.M., Sint, S., Vierhauser, M., Wortmann, A., Wimmer, M.: Digital Twin Platforms: Requirements, Capabilities, and Future

Prospects. IEEE Softw. 39, 53–61 (2022).
9. Autiosalo, J., Vepsalainen, J., Viitala, R., Tammi, K.: A Feature-Based Framework for Structuring Industrial Digital Twins. IEEE Access. 8, 1193–1208 (2020).
10. Pretterhofer, M., Mezhuyev, V.: Evaluation of software tools in the domain of field service management. ACM Int. Conf. Proceeding Ser. 119–123 (2021).
11. Dalmasso, I., Datta, S.K., Bonnet, C., Nikaein, N.: Survey, comparison and evaluation of cross platform mobile application development tools. 2013 9th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2013. 323–328 (2013).
12. Pinho, D., Aguiar, A., Amaral, V.: What about the usability in low-code platforms? A systematic literature review. J. Comput. Lang. 74, 101185 (2023).
13. Sullivan, A.J., Malave, D., Cheekoti, N.: Traffic simulation software comparison study. Security. 2217, (2004).
14. Sahay, A., Indamutsa, A., Di Ruscio, D., Pierantonio, A.: Supporting the understanding and comparison of low-code development platforms. Proc. - 46th Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2020. 171–178 (2020).