Digital Twins for Industrial Edge 4.0: Concepts and Tools

Adika Bintang Sulaeman
adika.sulaeman@aalto.fi

Tutor: Prof. Hong-Linh Truong

Abstract

Industry 4.0 is expected to be the next big phase in industry. Digital Twins (DT), defined as digital representations of physical objects such as machines, have an important role in Industry 4.0. This seminar paper discusses an overview of DT technology, key technologies to implement DT, DT frameworks, DT tools and how they can be combined to build DT systems. From the literature study conducted, the existing works from previous research can be combined to build DT systems.

KEYWORDS: Digital Twins, Industry 4.0, Frameworks, Tools

1 Introduction

Industry 4.0 is the next big phase in industry. With Industry 4.0, it is possible to gather real-time data from the machines that run in industry and process the data into something meaningful and useful. Industry 4.0 mainly consists of three supporting technologies: IoT, Cyber-Physical Systems (CPS), and Smart Factories [5]. The combination of these technologies builds interconnected devices forming Digital Twins (DT).

A DT models a physical object by creating a digital representation using real-time data [9]. The data is gathered throughout its life-cycle and used as the source to monitor, learn from, and enhance decision making. A DT enables engineers to monitor and understand how the machines behave once it is released and run by users. Furthermore, engineers can analyze the data and predict the future performance of the machines.

There are some use cases of DT for Industry 4.0. Consider a Printed Circuit Board (PCB) printer for electronic manufacturers. The PCB printer must be very precise, because the smallest error by the laser cutter may lead to PCB flaws. A DT enables engineers and technicians to monitor and analyze the data to predict the time when the spare parts wear out. Another example would be monitoring the jet engine of airplanes. By analyzing data gathered in real-time, engineers and technicians may predict failures in jet systems, which will lead to the reduction of airplane incidents. Furthermore, DT may give feedback to the engineers who design the machine to help them realize an agile development system.

The main value that the DT delivers is an understanding of product performance [9]. By understanding performance, manufacturers may detect and understand faults better, create an effective maintenance schedule, troubleshoot machines remotely, and decide appropriate add-on services.

The DT has some challenges in its development and implementation. In [1], the author has stated a number of challenges such as data consistency between the real physical assets and the digital representation, as well as connectivity and security concerns of cloud computing for DT. Software architectural aspects such as internal structure, APIs, integration, and runtime environment are also critical challenges for DT [8].

1.1 Scope and Goals

This paper aims to review the concept of DT for manufacturing in Industry 4.0 as well as technologies to build the DT system. This seminar paper is intended as a review paper for DT developers to build DT systems.

1.2 Structure

The rest of this paper is organized as follows. Section 2 discusses the key technologies for DT. Section 3 discusses the existing DT framework approaches and tools to build the DT framework. Section 4 concludes this review paper.

2 Principles of Building Digital Twin Systems

2.1 Requirement Analysis of Digital Twin Systems

There are for requirements for building DT systems [13]. These requirements can be used to determine key technologies for building DT systems.

From the aspect of DT applications, the first requirement is DT systems must be general enough to support various applications such as manufacturing, construction, and health-care.

From the technologies aspect, since the main aim of the DT is to fuse the interaction in CPS, the DT must embrace the integration of new generation information technologies such as IoT, cloud computing, big data, and artificial intelligence (AI) to support its goal.

From the modeling object aspect, the DT must integrate and fuse operational data from physical space and simulated data from virtual space. The DT must also be encapsulated as services which have user-friendly interface and ease the user operations.

From the modeling method aspect, the DT requires high-fidelity virtual modeling. The virtual modeling includes geometrical, physical, behavior, and rules modeling.

2.2 Key Technologies of Digital Twins

The requirements analysis leads to the abstraction of the DT concept model. Fig. 1 shows the DT concept model which is divided into five dimensions, i.e., the Physical Entity (PE), Virtual Entity (VE), Services (Ss), DT Data (DD) and Connection (CN). Each of these components has its own key technologies to build the DT system as a whole as shown in Fig. 2 [13].

The PE is the real physical entity in the physical space. It contains the real machines with their sensors and actuators. The technologies in the PE includes embedded systems to embed computing capabilities to the entities, RFID to track and identify the entities, Wireless Sensor Network (WSN) to transmit sensors' data and distributed sensor layout optimization to reduce information redundancy.

The VE consists of a set of models representing the PE. Modeling techniques such as three-dimension solid modeling, physics modeling, behavior modeling, and rule modeling are needed to create useful modeling of

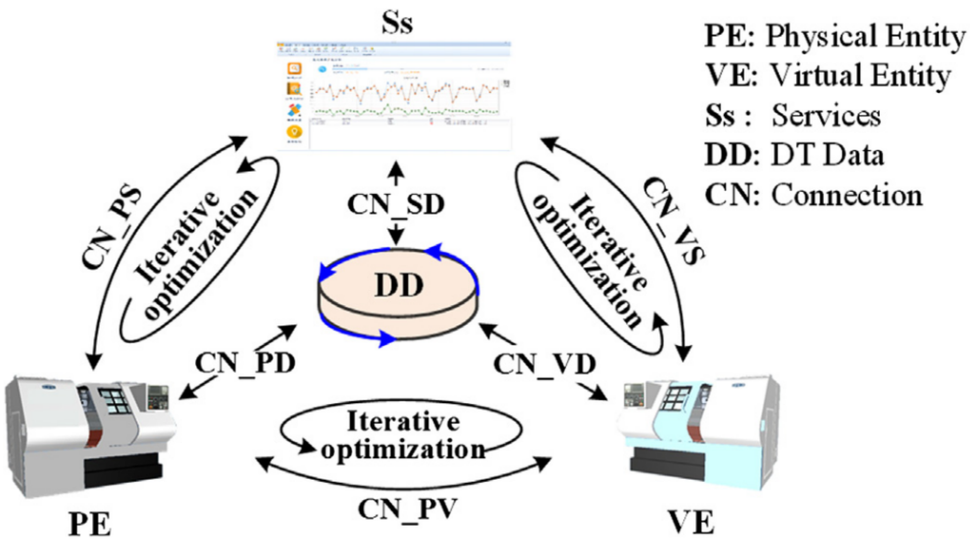

Figure 1. DT concept model [13]

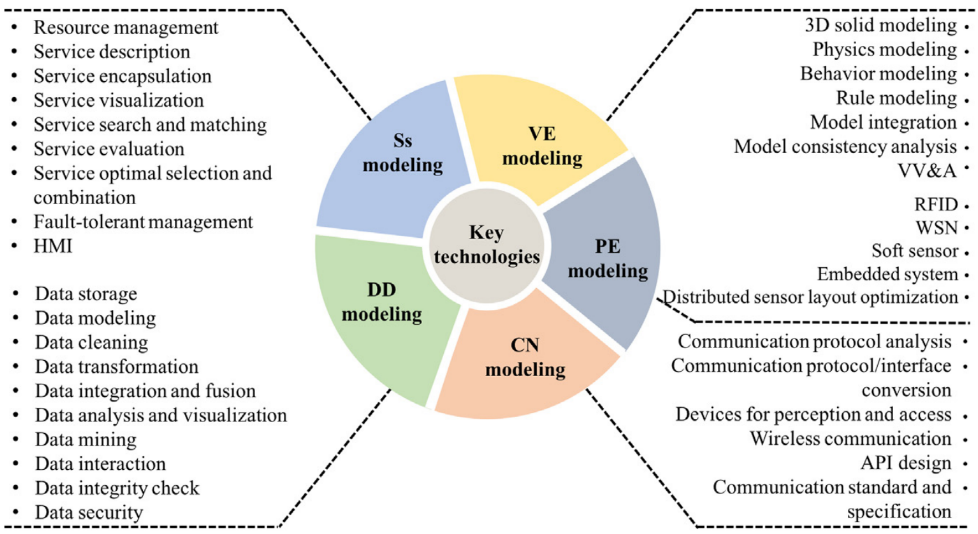

Figure 2. Key technologies for DT [13]

the PE. To keep PE and VE model consistent, model consistency analysis, verification, validation and accreditation must be implemented.

The Ss of the DT serves other DT dimensions' needs. The services used by PE include monitoring service, state prediction service, and energy consumption optimization service. VE uses the construction service, calibration service and test service for modeling. Some aspects are needed to create and maintain Ss, i.e., resource management, service description, service encapsulation, service visualization, service search and matching, service evaluation, service optimal selection and combination, fault-tolerant management and human-machine interface (HMI).

The DD contains data from physical and virtual space. Therefore, the number of heterogeneous data is high. Technologies for processing and securing data, such as data storage, data modeling, data transformation, data cleaning, data analysis, data mining, data integrity check and data security, are required in helping modeling DT data.

The CN connects different elements of DT. It enables PE, VE, Ss, and DD to communicate with each other. Key technologies of CN includes communication protocol analysis, communication protocol/interface conversion, wireless communication, Application Programming Interface (API) design, as well as communication standard and specification.

3 Implementing Digital Twin Systems

This section starts by discussing the existing DT frameworks, which give guideline or supporting structures for building DT systems. Then, it continues to discuss tools to realize DT framework. Finally, the discussion of the combination of DT frameworks and tools is presented.

3.1 Digital Twin Frameworks

Some researchers have proposed frameworks for building DT. While some of them propose holistic frameworks, there are also frameworks focusing on one particular part of DT. Table 2 shows a brief summary of the discussed frameworks.

Two examples of holistic frameworks for building DT were proposed by Yu Zheng et al. [15] and Zhuang et al. [16]. The framework proposed by Yu Zheng et al. aims to realize full-physical system mapping, life-cycle dynamic modeling, and the whole process of real-time optimization. The framework proposed by Zhuang et al. aims to build DT-based smart production management and control framework for product assembly shop-floor.

Although both frameworks were proposed to build DT for manufacturing, they have different approaches. The framework proposed by Yu Zheng et al. divided the system into three dimensions, i.e, physical space, virtual space, and information-processing layer. On the other hand, the framework proposed by Zhuang et al. divided the system into four dimensions, i.e., management of the physical space, construction of the virtual space of the shop-floor, DT and big-data driven prediction and production management and control service of the assembly shop-floor.

Framework	Focus	Future works (not implemented yet)
Zheng et al. [15]	Application framework for DT in manufacturing	Data synchronization; mapping methods between physical and digital space; application mode of DT
Zhuang et al. [16]	Framework of DT-based smart production management and control for assembly shop-floor	Construction and optimization of IoT networks in a physical assembly shop floor; construction, optimization, and running of an assembly shop-floor DT; construction of big data management platform; modeling and implementation of smart decision making algorithm
Zhang et al. [14]	DT modeling based on perception data modeled as ontology	Dynamic linkage between physical, digital, 3D; fully functional service system; AR for 3D models
Qi et al. [10]	Combining edge, fog, and cloud to build DT system	Increase the number of and the capbilities of edge nodes; architecture, platform, and standard of fog computing; data filtering to choose which data to process in edge, fog, and cloud; energy consumption; security
Lynn et al. [7]	modeling and controling physical space through DT	Automated plan generation; enhanced trajectory planning
Botkina et al. [2]	modeling and controling physical space through DT	Integrating non-tweeting machines to the system
Ciavotta et al. [3]	Software service for DT based on microservice	IoT protocols such as MQTT for CPS communication; AutomationML for data exchange

Table 1. Existing DT frameworks

Zhang et al. proposed another framework focusing on modeling DT workshop based on perception data [14]. The framework consists of three parts, i.e., physical model, ontology-based digital model, and the virtual model. The ontology-based data modeling is what makes this framework different from the others.

The aforementioned frameworks are the examples of the complete DT framework, starting from the physical to the virtual entity. However, they do not thoroughly describe the frameworks for each of the specific DT dimensions.

The service dimension of DT mentioned in section 2 is crucial for DT system because it connects one dimension to the others. There are many ways to build software services.

Ciavotta et al. suggested the use of microservice architecture for building middleware as a service for DT [3]. This middleware supports data collecting, authentication, and accessing DT resource and data. The benefits of using microservice architecture include agility, isolation, and resilience as well as elasticity. However, it also increases the system complexity at the same time.

The PE dimension of DT may come with different messaging formats and protocols. Software service middleware must support a variety of data formats coming from different dimensions of DT. For example, Schroeder et al. proposed a framework for building middleware to accept AutomationML data format and convert them into other data format via REST API [12].

Several frameworks for virtual entity have also been proposed. Schroeder et al. proposed the use of Augmented Reality (AR) to visualize DT [11], leveraging Vuforia Engine to build the AR system. For controlling physical devices through DT in VE, Lynn et al. [7] and Botkina et al. [2] have slightly different approaches. The framework proposed by Lynn et al. used raw TCP for controlling the device and raw UDP for getting the feedback data from the device, whereas the Botkina et al. approach used the standard Line Information System Architecture (LISA) and ISO 13399.

So far, all the frameworks discussed suggest the use of cloud computing for their processing and storage. However, some applications are latency sensitive. Bringing data to the cloud which can be physically far from the source of data may add significant latency overhead.

Qi Zhang et al. proposed a DT framework based on edge computing, fog computing and cloud computing to improve efficiency and reduce latency [10]. In many cases, the data processing and control of physical objects require very low latency. Therefore, edge computing architecture resides on the unit level to achieve that goal. The unit level entities are connected to the information management systems such as Enterprise Resource Planning (ERP) and Manufacturing Execution System (MES). Fog computing is suitable for connecting CPS to DT to improve efficiency. Finally, the data can also be stored and processed in the cloud at the SoS level for various purposes such as data analysis and long-term storage.

3.2 Tools for Digital Twins

In this subsection, tools or sets of software which equip the DT frameworks are discussed.

The PE needs tools to control, sample the operation and environment data, as well as send them to other dimensions of DT. MachineKit is an open-source platform for controlling machines, enabling users to control servo motors of Computer Numerical Control (CNC) machine based on the commands sent by VE [7].

The CN dimension, which connects PE to other dimensions of DT, represents the IoT in general. While there are many protocols and communication mechanisms for IoT, not all of them fits into industrial applications. OPC Unified Architecture (OPC UA) is a machine-to-machine communication protocol that is used by the framework proposed by Yu Zheng et al. [15]. MTConnect is another mechanism to retrieve data from machine tools. Coronado et al. used MTConnect to retrieve the data from the physical space to their DT system [4].

MES software is commonly used for both collecting data and monitor process and operation of the manufacturing process. MES is generally proprietary and expensive. As a result, small manufacturing enterprise may have difficulties in deploying MES in their system.

Coronado came up with a low-cost MES system based on Android [4]. This tool captures part and tooling information using an Android-based MES. The data from MES is combined with data from device sensors to represent parts, operators, capital equipment and consumable in shop-floor DT. This tool supports the Ss dimension of DT as an HMI and resource management tool.

There are various supporting tools for building the DT services, depending on the system architecture of the service. FIWARE is a middleware for exhanging data that can be used as a service for DT [12]. MAYA Support Infrastructure middleware used Netflix Eureka for service discovery and Elasticsearch-Logstash-Kibana (ELK) for log monitoring [3]. The recent trend in microservice technology is to scale the service horizontally; leveraging containers to deploy the services. For this case, containers orchestration tools such as Docker Swarm and Kubernetes may support the system. However, the containerization is not a silver bullet solution since it may impact the network performance in transfer rate and packet loss negatively [6].

The data from both physical space and virtual space must be stored in a database system. Depending on how the data is structured, several options for database platforms are available. For structured data, ODBC-compliant database such as MySQL can be used to store the data [14]. For unstructured data, NoSQL platform such as Apache Cassandra can be used [3].

There are some tools, open source or proprietary, available to support the DT framework in VE dimension. To control machine tools from the VE, Computer-Aided Manufacturing (CAM) software such as SculptPrint can be used [7]. Vuforia Engine can be used to leverage AR technology in building the VE [11]. To build 3D models of the physical objects, SolidWorks, Pro/E and CATIA can be used [16]. FlexSim can be used as a software simulation for manufacturing process [14]. Table 2 summarizes the tools for building DT systems.

3.3 Discussion

Several DT frameworks have been discussed to analyze the use cases and limitations. From the limitations and future works of those frameworks, a more complete framework can be built. In principle, the new framework can be formed by combining existing frameworks and adjusted to the application requirements.

For controlling machines in physical space, the framework proposed by Botkina et al. [2] must have tweeting machines in the shop-floor. While this tweeting machines development is not discussed, Lynn et al. [7] describe the mechanism of sending feedback data from actuators with raw UDP. The use of raw UDP by Lynn et al. might not be applicable if it does not comply with existing standards. Instead, OPC UA can be used as the protocol, as the framework developed by Yu Zheng et al. used [15].

The services in DT is described by the framework proposed by Ciavotta

Tools	Domain	Roles
MachineKit	PE	Platform for machine control applications
MTConnect	CN	Manufacturing standard to retrieve information from machines
OPC UA	CN	Machine to machine communication for industrial automation
FIWARE	Ss	Middleware for data exchange
Netflix Eureka	Ss	Service discovery in microservice architecture
Docker/containerd and Docker swarm/Kubernetes	Ss	Containerization and orchestration of deployed services
Relational data storage (MySQL, PostgreSQL)	DD	SQL database
NoSQL data storage (Apache Cassandra)	DD	NoSQL database
FlexSim	VE	Simulation software to model, simulate, predict, and visualize systems in manufacturing
SculptPrint	VE	Software for generating tool paths for 5-axis machine tools
CATIA	VE	3D physical modeling
SolidWorks	VE	
Pro/E	VE	
Android-based MES [4]	Ss and VE	Low cost and open source based Manufacturing Execution System (MES)

Table 2. Tools for building DT systems

et al. [3]. This service is supposed to be deployed in the cloud. However, it does not cover the latency sensitive applications. The combination of the framework proposed by Ciavotta et al. and Qi et al. [10] might be the solution for case sensitive applications since it offers the use of edge, fog, and cloud computing to deploy the services. Furthermore, the use of containers and orchestration in fog, edge, and cloud computing might be used to ease the deployment, maintenance and horizontal scaling of the services.

For modeling the physical space in the virtual space, framework proposed by Zhuang et al. [16] covers the data flow, Zhang et al. [14] proposed the use of ontology for modeling the data, and Schroeder et al. [11] proposed the use of AR to model the physical space in VE domains. The combinations of these three modeling frameworks might be useful to create a robust, complete, and useful mechanism of modeling DT systems.

In conclusion, the fusion of these frameworks can be combined according to the application requirements, cost and available resources.

4 Conclusion and Future Work

DT represents physical objects such as industrial machines as digital entities. DT can help monitoring machines and understanding product performance.

DT can conceptually be divided into five dimensions, i.e., the Physical Entity (PE), Virtual Entity (VE), Services (Ss), DT Data (DD) and Connection (CN). This seminar provides a review of existing frameworks and tools, as well as how they correlate to the aforementioned DT dimensions.

Seven frameworks have been analyzed and their chosen tools for DT are also mentioned. These frameworks and tools can be combined to form a more complete framework or to create a framework specific to some certain application requirements. The tools needed to build the DT system to support the frameworks are also discussed. The key technologies used by these frameworks and tools include IoT, embedded systems, software services and APIs, data storage and processing as well as modeling and controlling software.

The research directions or future work for this DT frameworks and tools include the standardization of connection between physical to digital space, adaptive service for modeling different data format, benchmarking and analyzing existing protocols and message formats as well as latency impact of DT to the control systems.

References

[1] Diethelm Bienhaus. Patterns for the industrial internet/industrie 4.0. In *Proceedings of the 22nd European Conference on Pattern Languages of Programs*, page 17. ACM, 2017.

[2] Darya Botkina, Mikael Hedlind, Bengt Olsson, Jannik Henser, and Thomas Lundholm. Digital twin of a cutting tool. *Procedia CIRP*, 72:215–218, 2018.

[3] Michele Ciavotta, Marino Alge, Silvia Menato, Diego Rovere, and Paolo Pedrazzoli. A microservice-based middleware for the digital factory. *Procedia Manufacturing*, 11:931–938, 2017.

[4] Pedro Daniel Urbina Coronado, Roby Lynn, Wafa Louhichi, Mahmoud Parto, Ethan Wescoat, and Thomas Kurfess. Part data integration in the shop floor digital twin: Mobile and cloud technologies to enable a manufacturing execution system. *Journal of Manufacturing Systems*, 48:25–33, jul 2018.

[5] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 3928–3937. IEEE, 2016.

[6] Nane Kratzke. About microservices, containers and their underestimated impact on network performance. *arXiv preprint arXiv:1710.04049*, 2017.

[7] Roby Lynn, Mukul Sati, Tommy Tucker, Jarek Rossignac, Christopher Saldana, and Thomas Kurfess. Realization of the 5-axis machine tool digital twin using direct servo control from cam. *National Institute of Standards and Technology (NIST) Model-Based Enterprise Summit*, 2018.

[8] Somayeh Malakuti and Sten Grüner. Architectural aspects of digital twins in iiot systems. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, page 12. ACM, 2018.

[9] Matthew Mikkel and Jen Clark. Cheat sheet: What is digital twin? internet of things blog. `https://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/`, January 2018. (Accessed on 01/30/2019).

[10] Qinglin Qi, Dongming Zhao, T Warren Liao, and Fei Tao. Modeling of cyber-physical systems and digital twin based on edge computing, fog computing and cloud computing towards smart manufacturing. In *ASME 2018 13th International Manufacturing Science and Engineering Conference*, pages V001T05A018–V001T05A018. American Society of Mechanical Engineers, 2018.

[11] Greyce Schroeder, Charles Steinmetz, Carlos Eduardo Pereira, Ivan Muller, Natanael Garcia, Danubia Espindola, and Ricardo Rodrigues. Visualising the digital twin using web services and augmented reality. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pages 522–527. IEEE, 2016.

[12] Greyce N. Schroeder, Charles Steinmetz, Carlos E. Pereira, and Danubia B. Espindola. Digital twin data modeling with AutomationML and a communication methodology for data exchange. *IFAC-PapersOnLine*, 49(30):12–17, 2016.

[13] Fei Tao, Meng Zhang, and A.Y.C. Nee. Five-dimension digital twin modeling and its key technologies. In *Digital Twin Driven Smart Manufacturing*, pages 63–81. Elsevier, 2019.

[14] Qi Zhang, Xiaomei Zhang, Wenjun Xu, Aiming Liu, Zude Zhou, and Duc Truong Pham. Modeling of digital twin workshop based on perception data. In *Intelligent Robotics and Applications*, pages 3–14. Springer International Publishing, 2017.

[15] Yu Zheng, Sen Yang, and Huanchong Cheng. An application framework of digital twin and its case study. *Journal of Ambient Intelligence and Humanized Computing*, 10(3):1141–1153, 2019.

[16] Cunbo Zhuang, Jianhua Liu, and Hui Xiong. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. *The International Journal of Advanced Manufacturing Technology*, 96(1-4):1149–1163, feb 2018.