A Digital Twin Architecture for Real-Time Supervision of an Olive Oil Production Mill

Hendrick Felipe Scheifer*, Luisa Jorge*, Flávia Pires* and Clara Vaz*
* Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC)
Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
Email: {hendrick, ljorge, clvaz, fpires}@ipb.pt

***Abstract*—The food processing industry is a highly complex sector that is increasingly embracing Industry 4.0 technologies, such as Cyber-Physical Systems, Artificial Intelligence, the Internet of Things, and Digital Twins (DTs), to enhance efficiency, quality, and sustainability. While DTs have demonstrated promising results in industries like dairy, beverages, and chocolate, their implementation in olive oil production remains limited. Key challenges include the need to create models adaptable to different device categories and the lack of standardized methods for capturing and transferring data between physical assets and their virtual counterparts.**

This work proposes a scalable architecture, based on ISO 23247, for developing DTs that support decision-making, 3D visualization, and simulation. The architecture was experimentally validated through a case study of an olive oil production mill where a DT was implemented. The resulting application integrated essential features such as data- and event-driven production monitoring for rapid anomaly detection, as well as digital model management to ensure the DT evolved in step with changes in the physical system's infrastructure.

Keywords: Digital Twin, Monitoring, Olive Oil Mill, Industrial IoT, Industry 4.0

I. Introduction

The food processing industry presents a high level of complexity due to physical and chemical factors resulting from processing conditions and the properties of raw materials [1]. Furthermore, the quality of the food produced can be influenced by input materials, especially during seasonal fluctuations in the supply of raw materials [2]. This complexity, combined with challenges such as variability in consumer demands, product diversification, difficulty in recruiting qualified personnel, and sustainability concerns, motivates the industry to seek approaches to improve production efficiency, product quality, user decision-making, and process sustainability [3].

Traditionally, food production has relied mainly on trial-and-error practices, conducted by experienced operators or specialists [1]. With the emergence of Industry 4.0, technologies such as Artificial Intelligence (AI) and the Internet of Things (IoT) have enabled beneficial advances for the food industry, such as process efficiency [3], [4], [5], product quality [2], [6], [5], better decision-making [7], [8], and sustainability [3], [5]. One technology that stands out in the context of Industry 4.0 is the Digital Twin (DT), which can be defined as *"an integrated data-driven virtual representation of real-world entities and processes, with synchronized interaction at a specified frequency and fidelity"* [9]. The DT enables the implementation of several functionalities, such as real-time monitoring, simulation, data analysis, control, and decision-support [10], and is often deployed within Industrial IoT (IIoT) environments.

The implementation of DT technology in the food industry comprises a diverse range of applications specific to the challenges of the sector. For instance, DTs have been utilized to model and optimize processes such as beverage [11] and milk [7] pasteurization, production of milk powder [8], and liquid foods [6]. In terms of material handling and transformation, applications include the design and optimization of filtration systems for flour [12], intelligent cutting strategies in fish processing [13], and simulations of oil milling operations [14]. DTs have also been applied to enhance product quality, streamline production lines, and reduce energy consumption in sectors such as chocolate and beverages [5]. These applications highlight the potential of DTs to improve operational efficiency, product quality, and sustainability across the food processing industry.

Despite the widespread application of DTs in the food industry, few studies have explored the use of this technology in the context of olive oil production. The work presented in [14] proposes a solution that employs a DT for the real-time monitoring of olive oil production, enabling process optimization and incorporating 3D visualization features to enhance process understanding. While this study represents significant progress, considerable potential remains to be unlocked through the implementation of additional functionalities, such as simulations, data analysis, and decision-support capabilities.

This work aims to develop a DT with a scalable architecture, allowing for the integration of 3D visualization, simulation, and decision-making features, thereby enhancing the efficiency and sustainability of the olive oil production process. Unlike previous DT approaches, which often focus on isolated stages of production or rely on static and highly customized models, the proposed architecture adopts a modular and interoperable design that supports continuous adaptation to process changes. This flexibility enables seamless integration of heterogeneous data sources and facilitates future extensions without reconfiguring the entire system. The architecture was implemented for monitoring the production of an olive oil mill, demonstrating great potential for efficient production monitoring and rapid anomaly detection. Moreover, its dynamic structure allows digital models to be easily adapted according to changes in the production infrastructure, ensuring that the real process remains synchronized with its digital counterpart.

The remainder of this paper is organized as follows. Section II presents a review of what has been implemented in related works, including the applications of DT on food processing, its benefits, technologies commonly used, and challenges faced. Section III describes the proposed DT architecture, highlighting its layers. Section IV presents the study case of the olive oil production line and details the implementation of the DT. Section V discusses the results obtained from the implementation and provides an analysis of the current capabilities and limitations. Finally, Section VI concludes the paper and outlines directions for future work, including the development of simulation models and decision-support features.

II. Context and Related Work

The applications of DTs in the context of food processing cover a wide range of scenarios and vary across different sectors of the food industry, involving mainly liquid or pasty food production.

DT can be implemented in very specific situations within particular sectors, such as predicting possible anomalies in plant operation in the beverage pasteurization industry [11], [8], optimizing production processes in the ketchup industry, or improving air flow and spray injection nozzle configurations for milk powder production [6]. Other examples include, the pasteurization of liquid foods, the management of bag filters in flour production [12], and intelligent cutting processes in fish processing [13]. DTs have also been used to simulate industrial oil mill processes [14], optimize milk pasteurization processes [7], ensure chocolate quality, and enhance production lines while reducing energy consumption in the food and beverage sector [5].

Beyond these sector-specific implementations, DTs are increasingly applied to more general situations within the food industry. For instance, they support the development of self-aware food processing systems capable of responding to fluctuations in raw material quality and securing food supply chains [15]. They are also used to capture and simulate the state of food products and processes during production [2], control thermal food processing operations [16], enable autonomous cooking processes, and optimize dynamic cross-flow filtration for solid–liquid separation of sensitive feed streams [1]. Additional applications include grain storage quality management and the optimization of storage practices [17], fault detection in food production systems [18], pre-warming plants for the cold sterilization of fluid foods [19], and predictive maintenance and process control in industrial pilot plants [20].

In the development of DTs for food processing, a wide range of enabling technologies has been extensively explored and applied to address the challenges of monitoring, controlling, and optimizing production processes. Cloud computing has been leveraged to provide scalable computational resources and storage, supporting the execution of DT implementations and enabling real-time data processing and analysis [11]. The IoT plays a central role in acquiring and transmitting data from the physical entity, connecting sensors, actuators, and microcontrollers to the digital environment and ensuring seamless communication across the production system [1], [5]. Big data analytics complements this infrastructure by enabling the extraction of valuable insights from large volumes of heterogeneous data, supporting optimized decision-making, such as determining the most efficient cutting strategies in fish processing [13].

Visualization technologies, including advanced 3D visualization tools [14], [19], enhance the interpretation of complex process dynamics, allowing operators to intuitively understand system states and interactions. Furthermore, AI methods have been extensively applied within DT frameworks to model, simulate, and control food processing operations. Techniques such as explainable AI [2], neural networks [16], and deep neural networks [1] have been used to simulate product and process states, model heating processes, provide real-time feedback and control, detect malfunctions even with limited data [7], identify anomalies, and enable automated decision-making through machine learning approaches [20]. These technologies also facilitate online monitoring, predictive maintenance, and dynamic process optimization [8].

Collectively, the integration of cloud computing, IoT, big data analytics, visualization tools, and AI provides a robust technological foundation for DT in the food processing industry. Their combined use contributes to significant improvements in operational efficiency [3], [4], [5], [7], product quality [2], [6], [3], [5], decision-making [7], [8], and sustainability [3], [5], demonstrating the transformative potential of DT technology in modern food production systems.

The implementation of Digital Twins in food processing faces several technical and organizational challenges. From a technical perspective, key issues include the seasonal variability of raw material quality and the complexity of modeling internal biological and chemical properties [15], there are also difficulties in accurately measuring the sensory properties of food [16], need for a large amount of accurate data for models training [17], [7], high computational requirements [17] and need for optimization of information exchange [14]. Among these challenges, two aspects stand out as particularly critical for the effective implementation of DTs in food processing. The first is the lack of standardized methods for capturing and transferring information between the physical and virtual layers [4], [2], which often leads to fragmented data flows, incompatibility between systems, and limited interoperability across platforms. The second concerns the difficulty of developing models that are generalizable across different product categories [2], [3], [18], resulting in solutions that are highly specific, difficult to scale, and costly to adapt to new processes or materials. Overcoming these limitations is fundamental to enable interoperability, scalability, and reliability in digital representations of food systems. In this study, we address

Feature	Martinez-Ruedas et al. [14]	This work
Architectural foundation	Custom architecture	ISO 23247 standard
Primary DT function	Real-time monitoring, 3D SCADA Integration	Dynamic model management, real-time data/event monitoring
Digital Model Management	Integrated with 3D SCADA and linked to OPC-UA nodes	Dynamic creation/ modification via GUI
Communication Protocols	OPC-UA, ModBUS TCP, S7	HTTP, MQTT
Primary Optimization	Reduction of communication latency	Model adaptability and system interoperability

Table I: Comparison between this work and [14].

these challenges by proposing an architecture that enhance standardization and model adaptability, contributing to the development of more robust, flexible, and interoperable DT frameworks for the food industry.

While the Digital Twin system presented by Martinez-Ruedas et al. [14] shares the goal of real-time monitoring for olive oil mills and represents a significant direct predecessor to this work, the proposed architecture offers distinct and fundamental advantages in scalability, model management, and interoperability.

The system in [14] achieves remarkably low communication latency by tightly integrating the DT and 3D Supervisory Control and Data Acquisition (SCADA) using proprietary control and data protocols within a CPS. This focus on a robust, low-latency connection for monitoring and simulation validation is highly effective.

In contrast, our approach prioritizes architectural flexibility, future extensibility, and standardized interoperability according to ISO 23247. By utilizing an open-source framework for digital twin management and lightweight messaging protocols, we effectively decouple the digital representation from traditional control systems and physical devices. This allows for superior dynamic Digital Model Management, including the addition, modification, and removal of equipment models via the User Entity, without requiring changes to the core system code.

To clearly highlight the novel contributions of our architecture over this direct predecessor, Table I provides a comparative analysis across key architectural and functional features.

III. Digital Twin Architecture

In this section, a DT architecture for the real-time monitoring of an olive oil production mill is presented.

As Figure 1 illustrates, this DT framework is composed of three distinct layers: Data Collection and Device Control Entity, Core Entity, and User Entity, which together enable the complete digital representation of the physical system. This architecture covers the entire data flow, from the acquisition of information from the physical environment to the visualization and interaction with its digital counterpart. The design is grounded on the principles proposed in ISO 23247 [21], which provides standardized guidelines for partitioning Digital Twin systems into functional layers, with the necessary adaptions for production instead of manufacturing.

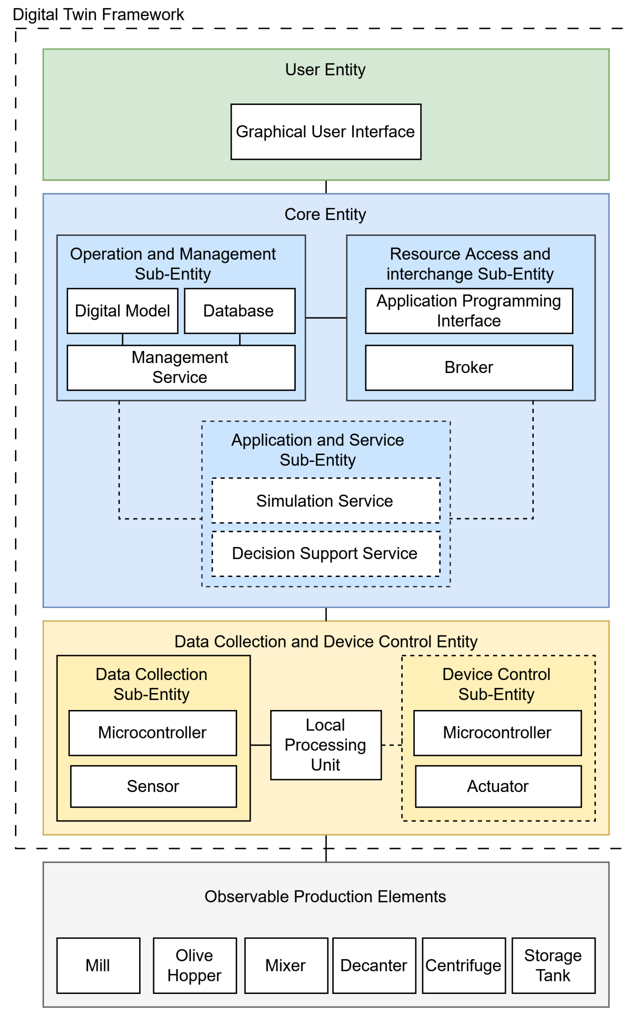

Figure 1: Digital Twin Architecture.

The **Data Collection and Device Control Entity** is responsible for the acquisition and pre-processing of data from the physical environment, and the device control. The Data Collection Sub-Entity employs a variety of sensors to monitor relevant characteristics of the olive oil production process, such as temperature or humidity. These sensors are connected to microcontrollers, which act as intermediaries between the raw sensor data and the local processing unit. Within the local processing unit, all collected data is centralized and transformed into structured payloads using widely adopted notations, such as JavaScript Object Notation (JSON) or Extensible Markup Language (XML). This pre-processing step ensures that the data is consistent, formatted correctly, and ready for further transmission. Finally, the processed data is sent to the internet or local networks using efficient communication protocols, such as Message Queuing Telemetry Transport (MQTT) or Secure Hypertext Transfer Protocol

(HTTPS), enabling real-time monitoring and minimizing latency in data delivery. Commands received from the DT are also handled by the local processing unit, which interprets and translates them into specific control signals. These commands are then transmitted to the Device Control Sub-Entity, where microcontrollers and actuators execute the required operations in the physical environment, ensuring bidirectional interaction between the digital and physical layers of the system.

For data reception, the **Core Entity** employs an MQTT broker, which facilitates lightweight and efficient communication between the local processing unit and the management service and also supports communication via an HTTP/HTTPS API. This design allows the system to handle high-frequency data streams without overloading the communication channels. Once received, the management service processes the incoming data, enabling the creation and management of the digital counterpart for each physical device through the use of Digital Models. These models capture the relevant features of the devices, allowing accurate digital representation. All processed data are stored in a database specifically designed to handle time-varying information, such as sensor readings and production metrics. Additionally, the Application and Service Sub-Entity leverages these digital representations to provide simulation and decision support functionalities, enabling process optimization, predictive insights, and enhanced operational efficiency.

The **User Entity** provides the graphical interface between the DT and the user, enabling the visualization of both the current state and historical states of the represented physical entities. These data are obtained from the Core Entity and organized in a visual and intuitive way for the system user. Within this layer, functionalities can be developed according to specific needs, ranging from the visualization of the current or historical states of physical entities to more complex and realistic representations through 3D visualization. Furthermore, it utilizes the data and insights provided by the Application and Service Sub-Entity to deliver enhanced visualizations and decision-support information to the user.

Finally, the **Observable Production Elements** represent the physical components of the production environment that are being monitored and will have a digital counterpart on the system. These include equipment such as the mill, olive hopper, mixer, decanter, centrifuge, and storage tank, all of which play specific roles in the production of olive oil. They serve as the data sources for the system, providing real-world operational information through sensors connected to the Data Collection Entity. This data enables the creation and updating of the Digital Model within the Core Entity. Each equipment will be described on the next section.

IV. Case Study and Implementation

This section presents details related to olive oil production mill case study and discusses key aspects of the implementation.

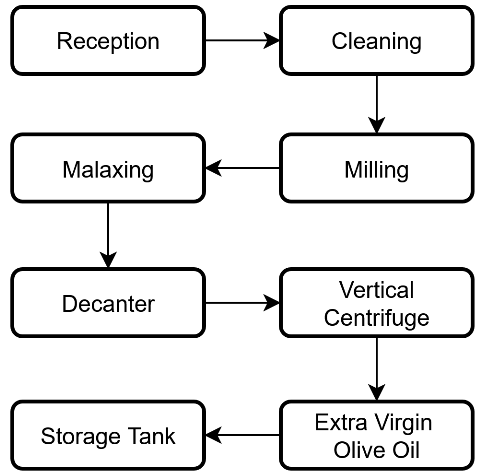

Figure 2: Olive oil production line.

A. Description of the Case Study

The olive oil production mill considered in this study consists of a production line integrating six types of equipment, each responsible for a specific stage in processing olives into extra virgin olive oil. Each stage plays a crucial role in ensuring the quality and purity of the final product, from fruit cleaning to oil extraction and storage. This production line is illustrated in Figure 2.

The olive oil production process begins with the reception of the olives, where they are unloaded and temporarily stored. The fruits then undergo cleaning and washing to remove leaves, stems, and other impurities, ensuring that only high-quality olives proceed to the production line. After cleaning, the olives are directed to the olive hopper, where received olives are temporarily stored before entering the production line. The olive hopper is monitored for temperature to control storage conditions. The olives are then directed to the mill, which crushes the fruits into a paste. The mill's operational parameters include temperature and water quantity, which influence the consistency of the paste, and consequently the quality of the final product.

This paste is subsequently transferred to the mixer, where controlled malaxing is performed to prepare the paste for the following stages. The mixer is monitored for temperature and humidity, providing information about the state of the paste. After mixing, the paste proceeds to the decanter, which separates the solid phase from the liquid one, containing both olive oil and vegetation water, which is the natural water content extracted from the olives during processing. The decanter is monitored for temperature and water quantity, allowing the observation of the conditions in which the phases' separation occurs.

The liquid is then processed by the centrifuge, which separates oil from water to obtain a higher-purity product. The centrifuge tracks temperature, water temperature, and water fat content, providing data on the quality of the extracted oil. Finally, the olive oil is transferred to the storage tank, where it is kept under specific conditions until bottling and commercialization. The storage tank is monitored for temperature, humidity, and gases (e.g., Methane, Propane, Hydrogen), which supports the management of storage conditions and product preservation.

B. Implementation

This subsection discusses the implementation of the proposed DT architecture for the previously presented case study. Several technologies were used in the implementation of the different elements that compose the architecture, relying mainly on Eclipse Ditto for the development of the management service, InfluxDB [22] for data persistence, and React [23] for building the graphical user interface.

The Data Collection and Device Control Entity includes the sensors and microcontrollers implemented in the olive oil production mill for the Data Collection Sub-Entity. These collect the data related to the observed features, sending them to the local processing unit for proper pre-processing, which relies on a device that can collect data inputs from several sensors, create a formatted payload, and send it to the Internet. The local processing unit was adapted to receive both Extensible Markup Language (XML) and JSON format messages, and also receive both communication protocols, HTTP/HTTPS and MQTT. These changes ensure greater flexibility and interoperability of the system, allowing seamless integration with different data sources.

On the Core Entity, for the development of the management service, which is the main component of the DT, Eclipse Ditto was deployed. This framework is based on the use of *Things* to represent each desired physical entity. *Things* are JSON objects that follow a structure defined in the framework's documentation [24]. These structures can be specified through models, which serve as templates for creating similar equipment (e.g., two similar mills follow the same model). Each *Thing* stores information such as an identifier, the access policy governing it, and sensor specific features. These correspond to the observed characteristics, such as temperature or humidity, that change over time. Each feature may contain attributes relevant to specific functionalities, such as the unit of measurement for data visualization and upper and lower value limits, used to determine the status of the feature, classified as critical if outside the configured value limits or non-critical if within range, enabling the implementation of alerts or automated control. Figure 3 shows an example of the JSON structure of a *Thing*, for this example, an olive hopper is represented with one observed feature, which is temperature, using a default start value of 20, a lower limit value of 16, and an upper limit value of 28.

To enable data transmission via MQTT on Eclipse Ditto, it is first necessary to use a broker to receive data published on specific topics, implemented using Mosquitto [25]. Additionally, specific connections must be created for each DT model. Connections are JSON objects that specify how data is received and also contain a function that correctly maps


```json
{
    "thingId": "olive.production:olive_hopper001",
    "policyId": "olive.default:policy",
    "features": {
        "temperature": {
            "properties": {
                "value": 20,
                "maxValue": 28,
                "minValue": 16,
                "unit": "\u00B0C",
                "timestamp": "2023-10-01T12:00:00.000Z"
            }
        }
    }
}
```


Figure 3: Thing JSON structure example.

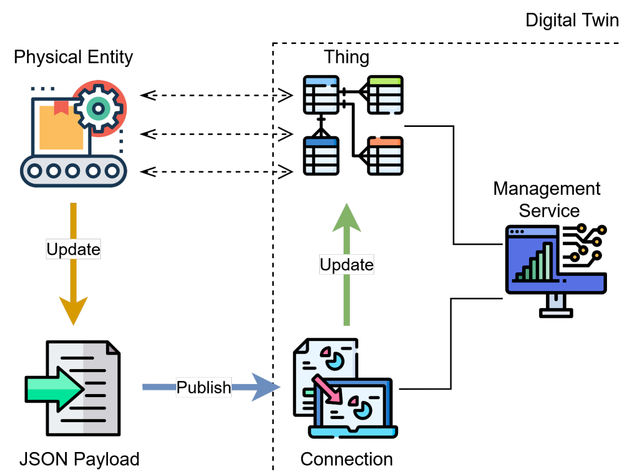

Figure 4: Flowchart of data updates. Icons from Flaticon [26].

the incoming JSON payload into messages using the Ditto Protocol (Eclipse Ditto's internal protocol). These connections are stored by Eclipse Ditto and are triggered whenever a message is published to one of the specified topics, indicating that there has been an update to the features of one of the *Things* associated with the model linked to that connection.

Figure 4 illustrates the data flow that occurs each time a reading is performed by the sensors.

Through the microcontrollers, the sensors monitor the features of the physical entity. The local processing unit generates a payload containing the identifier of the physical entity and the current values of the observed characteristics, publishing it to the specific topic of the *Thing* representing that entity. This payload is received by the broker and interpreted by the management service through the connection associated with that topic, which maps the received data and correctly updates the corresponding *Thing*, the digital representation of the physical entity. Both *Things* and connections are part of the DT, which uses them to manage the states of the physical entities. All data received by the management service is synchronized with the database through a connection between Eclipse Ditto and InfluxDB. This connection is not native to Eclipse Ditto, but it is available in the official Eclipse Ditto examples repository [27].

For the User Entity, a graphical interface in the form of a web application was implemented using the React framework [23] with the TypeScript programming language. The purpose of this interface is to provide easy visualization of the current state of the DT and to query historical state data, enabling more efficient monitoring of olive oil production. The data that can be observed in the application includes the values currently or previously collected by the sensors, the status of each observed feature, as well as state-change events, making it possible to identify when a feature has shifted to a critical or non-critical state.

In addition to the implemented monitoring features, the application includes a model management system, where it is possible to view which Things are currently present in the DT and their observed features. Within this system, users can modify the characteristics of existing *Things* or add new ones, either based on an existing model or a new one, for instance in the case of additional equipment being integrated into the production line. To facilitate the integration of new sensors for data collection, each *Thing* managed in this system includes a dedicated documentation screen, which provides details about its characteristics as well as the necessary configuration for data transmission to the DT, such as port, MQTT topic, and payload format. This system enables the expansion of the DT by integrating new equipment in the olive oil production line, with such expansion being easily carried out through the graphical interface without requiring a deep understanding of the entire DT structure, since all the information needed for data transmission is provided within the individual documentation of each model.

The Data Collection Sub-Entity has a lack in the sensing of the physical process, since not all of the production line is fully sensorized, which reduces the monitoring capacity and hinders the representation of its digital counterpart. In order to fully validate the potential of the DT architecture, simulated data will be used to fill in the gaps.

V. Experimental Validation and Results

A. Experimental Validation

The main functionalities of the developed DT for the digital representation of an olive oil mill lie in the visualization of the data collected for the observed features, the visualization of the status change events that occurred for each feature, and the management of the digital models. These functionalities are essential to ensure that the DT goes beyond a static digital representation, enabling the monitoring of the production process in real time and the integration of models that can be continuously updated and refined. To validate these characteristics, this section presents the implemented DT, focusing on those core functionalities: data and event visualization, and digital model management.

Figure 5: Specific equipment data visualization.

A fundamental functionality of the developed DT is the visualization of both the data collected from the production process and the events associated with status changes in the observed features. This functionality provides stakeholders with a clear and intuitive understanding of the operational state of the olive oil mill, making it possible to monitor critical variables in real time and to promptly identify deviations or anomalies.

The visualization is organized into different screens for data monitoring and event tracking. The data can be observed in a general overview, which presents the current state of each digitally represented equipment together with the last collected value for each observed feature. Alternatively, the data can be explored from the perspective of a specific equipment, allowing the analysis of all previously collected values for each observed feature over time. The general overview enables the rapid identification of features currently in a critical state, with the possibility of filtering the visualization to display only the devices that have features in such a state. On the other hand, the equipment-specific view provides graphical representations with threshold ranges that distinguish between critical and non-critical values for each observed feature, facilitating the identification of anomalies and abnormal behaviors over time. Figure 5 illustrates this functionality, showing the graphical visualization of the observed features for one equipment, where criticality thresholds are highlighted to support anomaly detection.

The events occurring in the system can be visualized in a manner similar to the data, through different screens that allow either a general overview of the last event recorded for each observed feature of every equipment, or a detailed view of all events associated with the features of a specific equipment over a defined period of time.

The general overview provides a quick insight into the last status change event for each observed feature, facilitating the identification of when the equipment entered its current critical state. In contrast, the equipment-specific event visualization enables the analysis of all status changes within a given time window, supporting the identification of recurring patterns in the occurrence of critical events.

In terms of management of the digital models, the developed DT enables a dynamic digital representation that can be

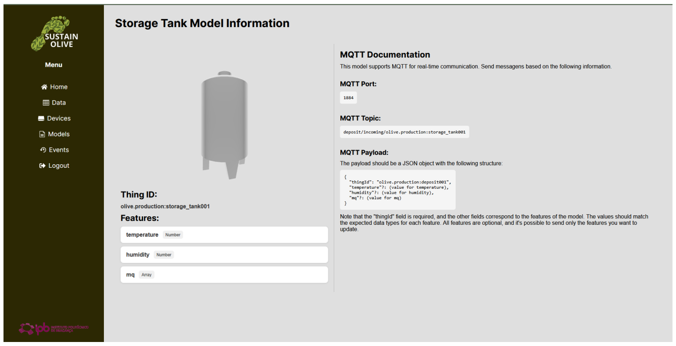

Figure 6: Model documentation screen.

adapted according to changes in the infrastructure of the olive oil mill. The system includes a model management functionality. This functionality allows the addition of new models for equipment incorporated into the production line, the modification of existing models to remove or add observed features depending on the available sensors, the removal of models for equipment that has been decommissioned, and the visualization of the documentation associated with an existing model. Which includes information about the model and its configuration for data transmission via MQTT, as shown on Figure 6. These operations make it possible to keep the DT continuously updated and aligned with the evolution of the production system, while also allowing the modification of models without requiring changes to the source code or prior knowledge of its implementation.

The implemented functionalities were validated by verifying their ability to support the monitoring and management needs of the olive oil mill. The visualization components proved effective in providing both a global overview and equipment-specific insights, enabling the identification of critical states and the analysis of historical behaviors. The event tracking ensured that stakeholders could follow the evolution of status changes and detect recurring patterns, supporting operational awareness and anomaly detection. Thus, the proposed DT should inform the decision makers about the need to control the critical parameters, such as the temperature and the water level. This procedure assures the quality of the olive oil during the process and its sustainability. In addition, the model management functionality confirmed the flexibility of the DT by allowing the addition, modification, and removal of equipment models without changes to the system code, ensuring that the digital representation remains synchronized with the physical infrastructure. Together, these validation steps demonstrate that the DT achieves its objectives of real-time monitoring, adaptability, and continuous alignment with the production system.

B. System Limitations and Discussion

The implementation revealed several technical constraints that must be addressed to ensure reliable system operation. The framework used, Eclipse Ditto, manages Things primarily through the HTTP protocol and does not provide native MQTT interfaces. As a result, connections between devices and their corresponding Things must be carefully managed to remain synchronized with the system's data structure. This requirement implies that any modification to the Thing models or their associated connections must be performed atomically, that is, whenever one is updated, the other must also be updated simultaneously. Failure to maintain this atomicity can lead to inconsistencies between the virtual representation and the physical device, affecting the reliability of the DT.

To ensure data security, a user management and authentication service should be implemented so that only authorized users can access the system's data. In addition, the components responsible for data reception, including the Local Processing Unit and the MQTT Broker, must be designed and deployed with high stability to prevent potential data loss or transmission failures.

VI. Conclusions and Future Work

The complexity presented in the food processing industry motivates the sector to investigate approaches to enhance production efficiency, product quality, user decision-making, and process sustainability through emerging technologies such as DT. This study presents a DT architecture for the real-time monitoring of an olive oil mill. The proposed architecture is based on ISO 23247 and separates the DT into different entities, allowing a clear distinction between the layers responsible for data collection, digital representation management and synchronization, and user interaction. The DT implemented in the context of olive oil production provides key functionalities, such as real-time and historical data and event monitoring, enabling the rapid detection of anomalies, and, most importantly, the management of digital models, which ensures the continuous synchronization of the DT with its physical counterpart.

As part of the planned system architecture, future developments will include the implementation of features such as 3D models for the visualization of each piece of equipment involved in olive oil production, providing a more accurate visual representation of the physical entity, as well as decision-making mechanisms using AI to optimize the olive oil production process and improve efficiency. Further enhancements are also planned in the field of IoT, such as the integration of additional sensors to monitor a wider range of characteristics of the physical entities, enabling a digital representation that is closer to reality. Moreover, the incorporation of actuators is foreseen to automate actions related to the control of the physical process, as defined on the Device Control Sub-Entity of the architecture.

Acknowledgement

This work was supported by Plano de Recuperação e Resiliência, within the project "SustainOlive", PRR-C05-i03-I-000187. The authors are also grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds: UID/05757 - Research Centre in

Digitalization and Intelligent Robotics (CeDRI); and SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020).

REFERENCES

[1] E. Purlis, "Digital twin methodology in food processing: Basic concepts and applications," *Current Nutrition Reports*, 12 2024.

[2] C. Krupitzer and T. Noack, "Digifoodtwin: Digital biophysical twins combined with machine learning for optimizing food processing †," *Engineering Proceedings*, vol. 19, 2022.

[3] E. E. M. Abdurrahman and G. Ferrari, "Digital twin applications in the food industry: a review," *Frontiers in Sustainable Food Systems*, vol. 9, 2025.

[4] E. Henrichs, T. Noack, A. M. P. Piedrahita, M. A. Salem, J. Stolz, and C. Krupitzer, "Can a byte improve our bite? an analysis of digital twins in the food industry," *Sensors*, vol. 22, 1 2022.

[5] K. A. Singh, F. Patra, T. Ghosh, N. K. Mahnot, H. Dutta, and R. K. Duary, "Advancing food systems with industry 5.0: A systematic review of smart technologies, sustainability, and resource optimization," *Sustainable Futures*, vol. 9, p. 100694, 6 2025.

[6] T. Eppinger, G. Longwell, P. Mas, K. Goodheart, U. Badiali, and R. Aglave, "Increase food production efficiency using the executable digital twin (xdt)," *Chemical Engineering Transactions*, vol. 87, pp. 37–42, 2021.

[7] O. Kadri and A. Abdelhadi, "Optimizing milk pasteurization diagnosis through deep q-networks and digital twin technology," *Int. J. Web Serv. Res.*, vol. 21, 1 2024.

[8] G. P. Tancredi, G. Vignali, and E. Bottani, "Integration of digital twin, machine-learning and industry 4.0 tools for anomaly detection: An application to a food plant," *Sensors*, vol. 22, p. 4143, 6 2022.

[9] Digital Twin Consortium, "The DTC Glossary," 2025. [Online]. Available: https://www.digitaltwinconsortium.org/glossary/glossary/digital-twin

[10] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitão, "Digital twin in industry 4.0: Technologies, applications and challenges," in *2019 IEEE 17th international conference on industrial informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 721–726.

[11] E. Bottani, G. Vignali, and G. P. C. Tancredi, "A digital twin model of a pasteurization system for food beverages: Tools and architecture," in *IEEE Int. Conf. Eng. Technol. Innov*, 6 2020.

[12] G. P. C. Tancredi, E. Bottani, and G. Vignali, "Digital twin-enabled process control in the food industry: proposal of a framework based on two case studies," *International Journal of Production Research*, vol. 62, pp. 4331–4348, 2024.

[13] J. Fu, Y. He, and F. Cheng, "Intelligent cutting in fish processing: Efficient, high-quality, and safe production of fish products," *Food and Bioprocess Technology*, vol. 17, pp. 828–849, 4 2024.

[14] C. Martinez-Ruedas, J. M. Flores-Arias, I. M. Moreno-Garcia, M. Linan-Reyes, and F. J. Bellido-Outeiriño, "A cyber–physical system based on digital twin and 3d scada for real-time monitoring of olive oil mills," *Technologies*, vol. 12, p. 60, 4 2024.

[15] C. Krupitzer, T. Noack, and C. Borsum, "Digital food twins combining data science and food science: System model, applications, and challenges †," *Processes*, vol. 10, 9 2022.

[16] M. Kannapinn, M. K. Pham, and M. Schäfer, "Physics-based digital twins for autonomous thermal food processing: Efficient, non-intrusive reduced-order modeling," *Innovative Food Science & Emerging Technologies*, vol. 81, p. 103143, 10 2022.

[17] T. Anukiruthika and D. S. Jayas, "Ai-driven grain storage solutions: Exploring current technologies, applications, and future trends," *J. Stored Prod. Res.*, vol. 111, p. 102588, 5 2025.

[18] S. Saraeian and B. Shirazi, "Digital twin-based fault tolerance approach for cyber–physical production system," *ISA Transactions*, vol. 130, pp. 35–50, 11 2022.

[19] G. P. C. Tancredi, G. Vignali, and E. Bottani, "Implementation of a remote control and system monitoring via a digital twin on an industrial pilot plant," *IFAC-PapersOnLine*, vol. 55, pp. 1128–1133, 1 2022.

[20] G. P. C. Tancredi, "Digital twin-based control system enhanced by machine learning in food processing," Ph.D. dissertation, UNIVERSITA' DEGLI STUDI DI PARMA, 2023.

[21] "Iso 23247 digital twin framework for manufacturing." [Online]. Available: https://ap238.org/iso23247/

[22] "Influxdb time series data platform | influxdata." [Online]. Available: https://www.influxdata.com/

[23] "React." [Online]. Available: https://react.dev/

[24] "Eclipse ditto™ • open source framework for digital twins in the iot." [Online]. Available: https://eclipse.dev/ditto/

[25] "Eclipse mosquitto." [Online]. Available: https://mosquitto.org/

[26] "Vector icons and stickers - png, svg, eps, psd and css." [Online]. Available: https://www.flaticon.com/

[27] "eclipse-ditto/ditto-examples: Eclipse ditto™: Digital twin framework - examples." [Online]. Available: https://github.com/eclipse-ditto/ditto-examples