Project Acronym: STAR
Grant Agreement number: 956573 (H2020-ICT-2020-1 – Research and Innovation Action)
Project Full Title: Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines
Project Coordinator: INTRASOFT International

Funded by the Horizon 2020 Framework Programme of the European Union

DELIVERABLE

D5.2 – Digital Models for Human Centric AI-based Production Processes – Final version

Dissemination level	PU -Public
Type of Document	Other
Contractual date of delivery	31/12/2022
Deliverable Leader	SUPSI
Status - version, date	Final – v1.0, 05/01/2023
WP / Task responsible	WP5
Keywords:	Human digital models, human digital twin, Digital twin

Executive Summary

This document is an accompanying of the software solutions developed within T5.1, providing an overview of the activities and the results achieved within M12-M24. The goal of T5.1 is to develop a reference model that enables the storage, exchange, and use of technical information that involves humans in AI-based production systems.

The main activities carried out during T5.1 between M3 and M12 were: 1) the definition of the reference model; 2) the development of the Human Digital Twin (HDT) Core Infrastructure; 3) the development of a functional module to recognize the worker intention (Worker's Intention Recognition Module).

From M12 to M24, the task focuses on extensive tests and bug fixes to increase the HDT Core Infrastructure robustness. SUPSI has revised some of its components to offer new functionalities to the partners of STAR and to improve the reliability and efficiency of the communication between the middleware and the Historical Data Manager.

A second prototype of the HDT core infrastructure has been provided and is currently available on GitHub only for STAR's partners actually involved in the pilots. This implementation is based on a modular architecture specifically designed to use the HDT in different applications, relying on and integrating different types of modules. This allowed making available three different HDT instances to support the DFKI pilot, the Philips pilot and the integration with the Worker Training Platform.

Finally, this document reports on the activities and results related to the Workers' Intention Recognition Model developed by DFKI.

Deliverable Leaders:	SUPSI
Contributors:	DFKI
Reviewers:	JSI, UPRC
Approved by:	Charalampos Ipektsidis, John Soldatos (INTRA)

Document History			
Version	**Date**	**Contributor(s)**	**Description**
0.1	01/10/2022	SUPSI	Table of contents
0.2	01/11/2022	SUPSI	First draft
0.3	09/12/2022	SUPSI	Second draft (section 1-5)
0.4	15/12/2022	DFKI	Third draft (section X)
0.5	18/12/2022	SUPSI	Consolidated draft
0.6	20/12/2022	JSI, UPRC	Reviewed draft
0.7	27/12/2022	SUPSI	Final version
1.0	05/01/2023	INTRA	QA and creation of the submitted version

Table of Contents

Table of Figures

List of Tables

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AL	Active Learning
AMR	Autonomous Mobile Robot
HDT	Human Digital Twin
WP	Work Package

1 Introduction

This document accompanies the Human Digital Twin (HDT) Core Infrastructure and the Worker's Intention Recognition Module, which are the main results included in the D5.2 - Digital Models for Human Centric AI-based Production Processes - Final version (Other).

Section 2 sum-ups the architecture and the reference model behind the HDT Core Infrastructure, detailed in D5.1. Moreover, it defines the value proposition of this solution underlining the main advantages that it brings.

Section 3 provides an overview of the development activities performed between M13 and M24, including the developed agents and gateways and the instances prepared for testing and validation in pilots.

Section 4 describes the work carried out to develop the Worker's Intention Recognition Module, which is one of WP5's modules dedicated to predicting workers behaviours in the production system.

2 The Human Digital Twin Core Infrastructure

The arising of the Industry 5.0 paradigm and the established key role of workers in manufacturing require new Digital Twins to represent also humans. In fact, as cognitive automation becomes more and more pervasive and its behaviour unintelligible to humans, it becomes essential for improving performance and well-being, at the same time, to model humans as data-driven agents and to represent their interaction with the factory systems.

Currently, a standardised solution for creating Digital Twins is missing, forcing industrial solution architects to resort to ad-hoc implementations and models. These solutions lack re-usability, scalability and extensibility, preventing the introduction of a human digital representation in existent twins, so hindering the complete shift to the new Industry 5.0 paradigm [REF-01].

Such limitations are addressed by the Human Digital Twin Core Infrastructure (Clawdite Platform), an extensible and flexible IIoT - industrial internet of things - based platform with a twofold benefit: on the one hand, to support the creation of customised data representations of production systems and their entities including humans; on the other hand, to provide a modular infrastructure, along with its interchangeable components, for easy digital twin instantiation and ramp-up.

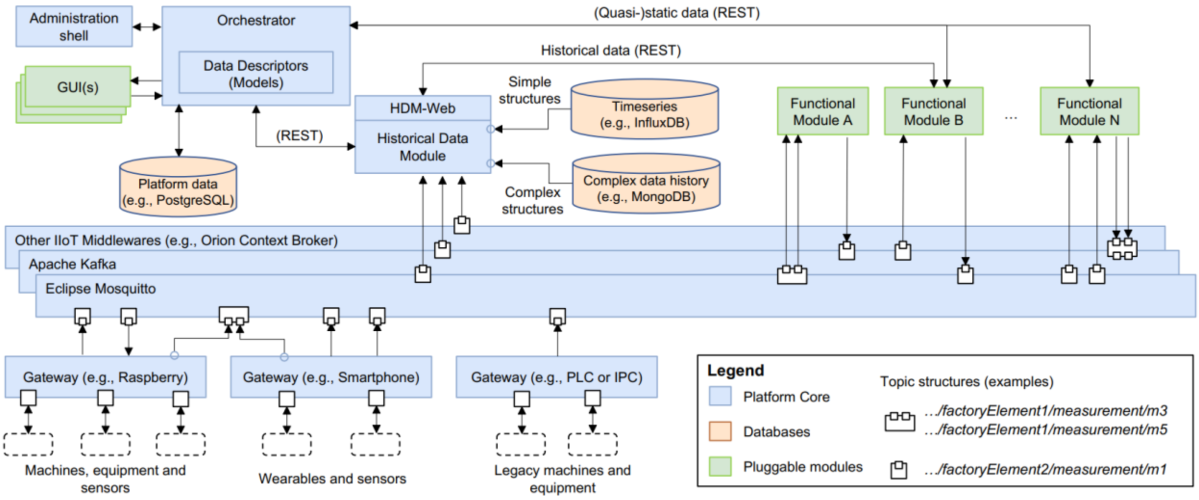

Figure 1 Human Digital Twin Core Infrastructure: architecture

The platform has been designed to satisfy the recurrent need for a solution capable of supporting the creation of HDT in different manufacturing applications. Instantiating the digital representation of factory entities (e.g., workers, cobots, workstations) requires a relevant effort in terms of modelling, infrastructure development, configuration and deployment [REF-02]. Moreover, the re-usability of the developed solutions and is limited.

The HDT Core Infrastructure has been designed to be a single source of production systems data, including worker-related ones. It offers a centralized access point that can be exploited by STAR's AI modules to compute complex features, feeding and enriching the HDT, or to make better decisions.

Starting from the ideas and the conceptualization provided by the human digital twin meta-model [REF-03], the reference model of the platform has been defined as in Figure 2 (details are available in D5.1) This aims to provide a common model that allows instantiating a DT, including humans and contextual elements. Section 3 provides the details on how the reference model has been exploited to instantiate the specific DTs for fulfilling the requirements of pilots.

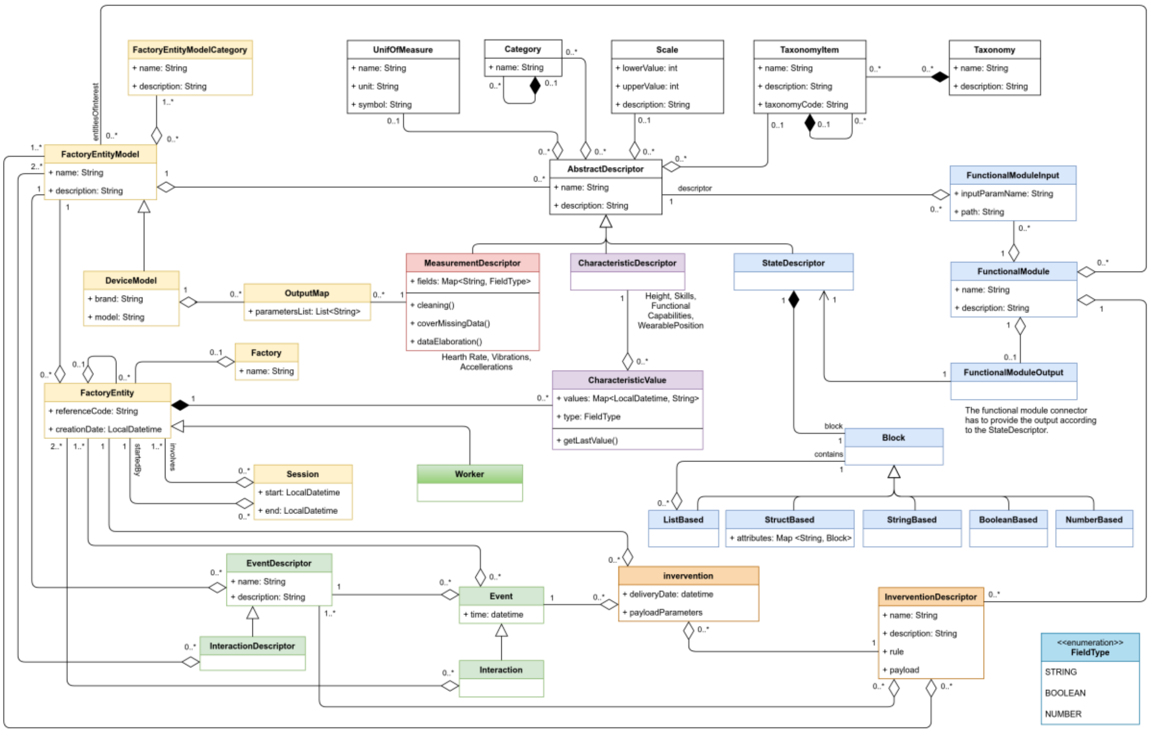

Figure 2 Human Digital Twin reference model

2.1 Benefits and Value proposition

The value proposition of the HDT Core Infrastructure is:

"Easy to customise and deploy solution for integrators and industrial solution architects to digitalise production systems, including human workers."

The main advantages brought by the adoption of this solution are the following:

- **Interoperability by design**: the Orchestrator relies on its underlying reference model to instruct satellite components (e.g., AI modules) on how, where, and when to publish/read data. The HDT Core Infrastructure allows users to define a common model to represent descriptors (characteristics, states, and measurements) within the DT, while the different data formats used in third-party functional modules are mapped to such descriptors by means of specific connectors. Since the mappings are stored within the DT, any applications at any time can understand how data are ingested into the DT by other components, making the applications interoperable by design. Existing platforms to realise DT like Ditto[1] and FIWARE[2] overlook the interoperability aspect, which is delegated to the final developer (e.g., by developing a specific ad-hoc connector to ingest data into the DT using a predefined data format).
- **Factory entities by composition**: Existing DT platforms tend not to provide users with a data model (e.g., FIWARE), or provide a very simple one (e.g., Ditto). In the first case, the duty of the data model design is in charge of the developer, while in the

[1] https://www.eclipse.org/ditto/
[2] https://www.fiware.org/marketing-material/fiware-for-digital-twins/

second case, the data model does not cover useful scenarios, e.g., the composition of different basic entities to represent complex entities. The HDT Core Infrastructure relies on the reference model that exploits both *composition* and *inheritance* concepts, allowing users to create *(i)* hierarchies of entities that inherit properties and functions from their parents, and *(ii)* complex entities built by different entities. The reference model simplifies and speeds up the definition of DT, while remaining flexible enough to meet specific domain requirements.

- **Easy and fast ramp-up**: the HDT Core Infrastructure integrates all the fundamental components to instantiate a DT, providing users with tools to manage data, orchestrate the DT, expose its entities and associated functions. To start using the DT, the user has only to perform a few configuration steps, including the definition of factory entities, the development of connectors to send data to the DT (if needed), the installation of source devices and functional modules, by specifying their mappings to DT's descriptors.
- **Extensibility**: the HDT Core Infrastructure exposes a REST API to ease the plug in of third-party functional modules. Functional module developers are aided by language-specific clients wrapping the API requests, speeding up the implementation process.
- **Humans as first-class citizens**: the HDT Core Infrastructure recognises workers' primary role in the factory and represents them with a dedicated model. This choice allows functional modules developers to apply ad-hoc policies to human entities (e.g., predicting modules may employ more sophisticated AI models when dealing with workers). Moreover, specific security policies may be applied when streaming workers' data to the DT by means of ad-hoc data connectors.

3 Advancements from M12 to M24

In M12 SUPSI delivered an already consolidated HDT Core Infrastructure for supporting the next phases of the project and the activities required for the downstream period. This means that the HDT Core Infrastructure was advanced enough to allow partners to showcase its application during the M18 review, where the HDT Core Infrastructure supported the implementation of a simplified scenario resembling the Philips pilot (i.e., a data gateway simulator sending physiological data to the HDT, feeding a functional module continuously predicting the perceived fatigue exertion of the monitored operator). In the period M13-M24, most of the work on the HDT Core Infrastructure covered bug fixing and testing to make the infrastructure more robust and ready to accommodate different pilots. Some improvements have been implemented as detailed in the following sections.

3.1 HDT Core Infrastructure

Aside from extensive tests and bug fixing to increase the HDT Core Infrastructure robustness, SUPSI worked on some of its components to provide STAR's partners with new functionalities, as well as improving the reliability and efficiency of the communication among the middleware and the Historical Data Manager, as following:

- **Orchestrator**: The Orchestrator of the HDT Core Infrastructure is implemented as a Spring Boot Web Service[3]. To ease the interaction of third-party components (e.g., gateways and functional modules), the activity focused on the automatic generation of clients by means of the OpenAPI generator[4], a library that speeds up the implantation of modules interacting with the HDT Core Infrastructure. The work required to adapt the API to the latest OpenAPI standards (v3.0.1) to make available the widest set of functionality and programming languages available[5]. Also, to automate the initialization of HDTs and make the pilot reproducible, SUPSI worked on population scripts that can be used for initializing from scratch and in a repeatable way the HDTs needed within the pilots (e.g., by creating the factory entities and functional modules relevant to the pilot). This activity will speed up the testing phases of the pilots. The improved version of the Orchestrator is available as a Docker image (tagged with version *0.2.0-star*).
- **Historical Data Manager**: The Historical Data Manager (HDM) oversees persisting data passing through the data middleware. This functionality is implemented by relying on the Eclipse Paho MQTT connector, which however comes with a bug ([#576]) that prevents the application from receiving messages from subscribed topic after a certain amount of time, leading to a data loss. A temporary workaround has been implemented in the HDM to allow the application to run until an official fix is released. The improved version of the HDM is available as a Docker image (tagged with version *0.2.0-star*).
- **Historical Data Manager Web**: The Historical Data Manager Web (HDM-Web) module exposes a REST API to make historical data accessible as timeseries. No major updates have been implemented in this module, other than updating the Java version (older versions caused an issue when executing the Docker container with some specific container management tools, i.e., Portainer).

[3] https://spring.io/projects/spring-boot
[4] https://openapi-generator.tech/
[5] The full list of available clients is accessible at https://openapi-generator.tech/docs/generators/#client-generators

3.2 Gateways and agents

At M12, a gateway has been provided as an Android 11 application for smartphones (Android Gateway), which integrates different agents for wearable devices, namely **Polar Agent, Empatica Agent,** and a **Garmin Agent.** In the context of STAR, we refer to "agents" as components supporting the connection to a specific family of devices (usually determined by the device manufacturer). At M12, the provided agents support the following devices:

- Polar Agent: Polar H10
- Empatica Agent: Empatica E4
- Garmin Agent: Garmin Venu 2

In the period M13-M24, the Android Gateway has been improved in different ways:

- The UI has been updated to improve the user experience
- Background services has been implemented to allow users using the application in background (i.e., when not displayed in foreground, or when the device is locked)
- The connection to multiple devices is supported by asking the user to select the device ID
- The Gateway now handles network issues (e.g., in contexts where the Internet connection is unstable). For short network unavailability (i.e., less than 10 minutes), the Gateway stores data in a buffer that is flushed to the HDT as soon as the network is back. For periods longer than 10 minutes, the Gateway persists a copy of collected metrics and sessions locally, on the internal memory of the device. When the network becomes available, the user has to manually flush the local data to the HDT by means of a dedicated feature.
- A new application section is available to support the labelling of workers' perceived fatigue. This feature will support the data collection for training the FaMS, a module required by the Philips pilot.

Concerning agents shipped with the Android Gateway, the Polar agent has been improved to extend the set of measurable parameters, i.e., photoplethysmogram (PPG), peak to peak interval (PPI), gyroscope (GYR), magnetometer (MAG), by including a connector for the Polar Verity Sense device. Given the limitations of the Garmin Connect IQ SDK (i.e., dependency on *Garmin Connect* application; dependency on an edge application to develop and install on the wearable device; limit of one connection at a time to a single device), the Garmin Agent has been discontinued

At M24, new gateways and agents have been released to support the execution of pilots, as it follows:

- **Raspberry Pi3 Gateway**: this gateway has been developed as a Java application, which already comes with a fully integration with the HDT Core Infrastructure middleware (i.e., Mosquitto MQTT) and Orchestrator. Currently, the Raspberry Pi3 Gateway supports the collection of environmental metrics data (e.g., luminosity, humidity, temperature, and cell-worker distance for worker presence).
- **Universal Robot Agent:** this agent is a Python application providing a connection to Universal Robot cobots (e.g., UR5) by means of the Real-Time Data Exchange (RTDE) interface. This agent enables a 2-way communication between the cobot and the HDT Core Infrastructure: the cobot continuously sends its execution parameters

to the agent, which forwards them to the HDT as States; conversely, the agent continuously reads new cobot configurations from the HDT middleware, and forwards them to the cobot via RTDE.

3.3 Instances available for validation

The HDT Core Infrastructure developed in STAR enables the easy and fast deployment of HDTs. In the next sections, the different instances of HDT created for the different pilots are described. In all the cases, the HDT Core Infrastructure is deployed as a set of docker containers on a server. The repository contains a Docker Compose file to speed up the deployment of the required services.

3.3.1 DFKI pilot

In the DFKI pilot, the HDT Core Infrastructure supports the integration of the Safety Zone Detection System (Functional Module), the AMR Fleet Optimizer (Functional Module) and a Robotino (Gateway).

The **Safety Zones Detection System** relies on its own data and video/image stream architecture. **Safety Zones Detection System** publishes computation results (mainly objects and workers positions) on the IIoT Middleware to update the HDT.

The **AMR Fleet Optimizer** accesses these data via HDT, together with the status of the AGV. Thanks to this information, it is able to define the best paths for the AGVs. These are published on the HDT. From the HDT Middleware, the Robotino accesses to this data, adapting its behaviour.

The Human Digital Twin Core Infrastructure main functions in this pilot are:

- Support the communication between the Robotino, Safety Zones Detection System, AMR Fleet Optimizer;
- Store and manage data from Robotino, Safety Zones Detection System, AMR Fleet Optimizer;
- Enable the integration of other third-party modules.

Figure 3 provides an overview of the integration from the point of view of the architecture.

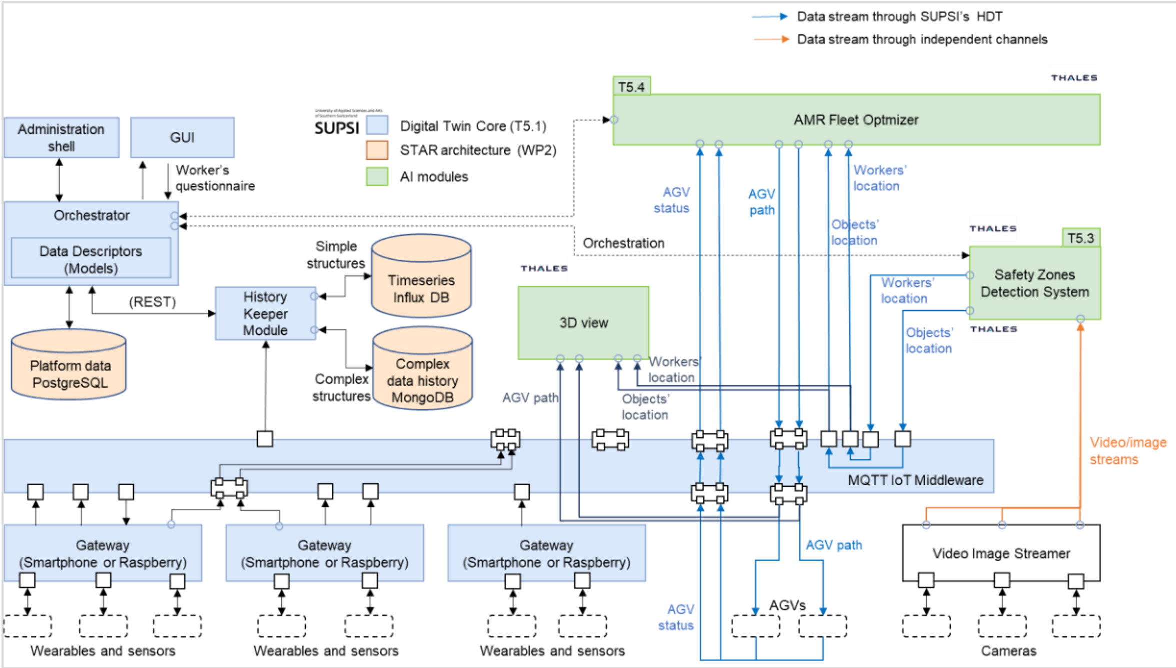

Figure 3 Integration overview: HDT, Safety Zones Detection System and AGV Fleet Optimizer

Table 1 describes the main entities model has been exploited to instantiate the specific DTs for fulfilling the requirements of pilots.

Table 1 Reference model instance and main entities: DKFI pilot

Name	Type	Description
RobotinoModel	DeviceModel	It describes any Robotino in the DFKI's laboratory. For each device, an instance of Type FactoryEntity is created.
Robotino_DFKI_1	FactoryEntity	It describes the Robotino moving in DFKI's laboratory.
WorkerModel	FactoryEntityModel	It describes any worker in the DFKI's laboratory.
Worker_DFKI_1	FactoryEntity	It describes the worker moving in DFKI's laboratory. A unique entity is used to represent any worker entering in the area (the first demo expects only one worker in the scene. Who is the worker is not relevant).
Obstacle	FactoryEntityModel	It describes any physical obstacle or barrier in the scene.
Table	FactoryEntity	It describes the table available in the laboratory, which represent the main obstacle.
AMRFleetOptimizer	FunctionalModule	It describes the module optimising the Robotino path. Thanks to its FunctionalModuleOutput, it feeds the RobotinoNextPosition. Its FunctionalModuleInput allows reading all the required inputs from the HDT.

SafetyZoneDetectionSystem	FunctionalModule	It describes the module identifying objects and workers positions Thanks to its FunctionalModuleOutput, it feeds the WorkerPosition and the ObjectPosition.
WorkerPosition	StateDescriptor	It describes the worker's position with coordinates X-Y. It refers to the Worker' FactoryEntity and it is feeded by the SafetyZoneDetectionSystem.
RobotinoPosition	MeasurementDescriptor	They describe the Robotino's features used by the AMRFleetOptimizer to define the best path. They refer to the Robotino's FactoryEntity and it is feeded by the Robotino Agent.
RobotinoBatteryStatus	MeasurementDescriptor	
RobotinoCurrentSpeed	MeasurementDescriptor	
RobotinoCurrentOrientation	MeasurementDescriptor	
RobotinoNextPosition	StateDescriptor	It describes the Robotino's instructions with coordinates X-Y. It refers to the Robotino's FactoryEntity and it is feeded by the AMRFleetOptimizer. The Robotino reads this State to define its motion path.

3.3.2 PHILIPS pilot

The goal of this pilot is to implement a system that makes automated quality inspections easier & faster by applying active learning (AL). The wanted result contains a system that can be implemented in the factory to be used to set up an automated quality control for a new product easily (meaning with little to no data) where the production personnel are able to check the products (in quality vs out-of-quality) and transfer that knowledge to the quality inspection algorithm by means of AL.

- The HDT Core Infrastructure supports the mental stress and attention estimation during the labelling activity in order to
- Understand whether the labelled data can be trusted or should be reviewed by multiple workers, to ensure the accuracy of the final label provided;

The Human Digital Twin Core Infrastructure main functions in this pilot are:

- Collect physiological data from users performing a labelling task;
- Make physiological data available to AI modules for models training and operations.

Figure 4 provides an overview of the integration from the point of view of the architecture.

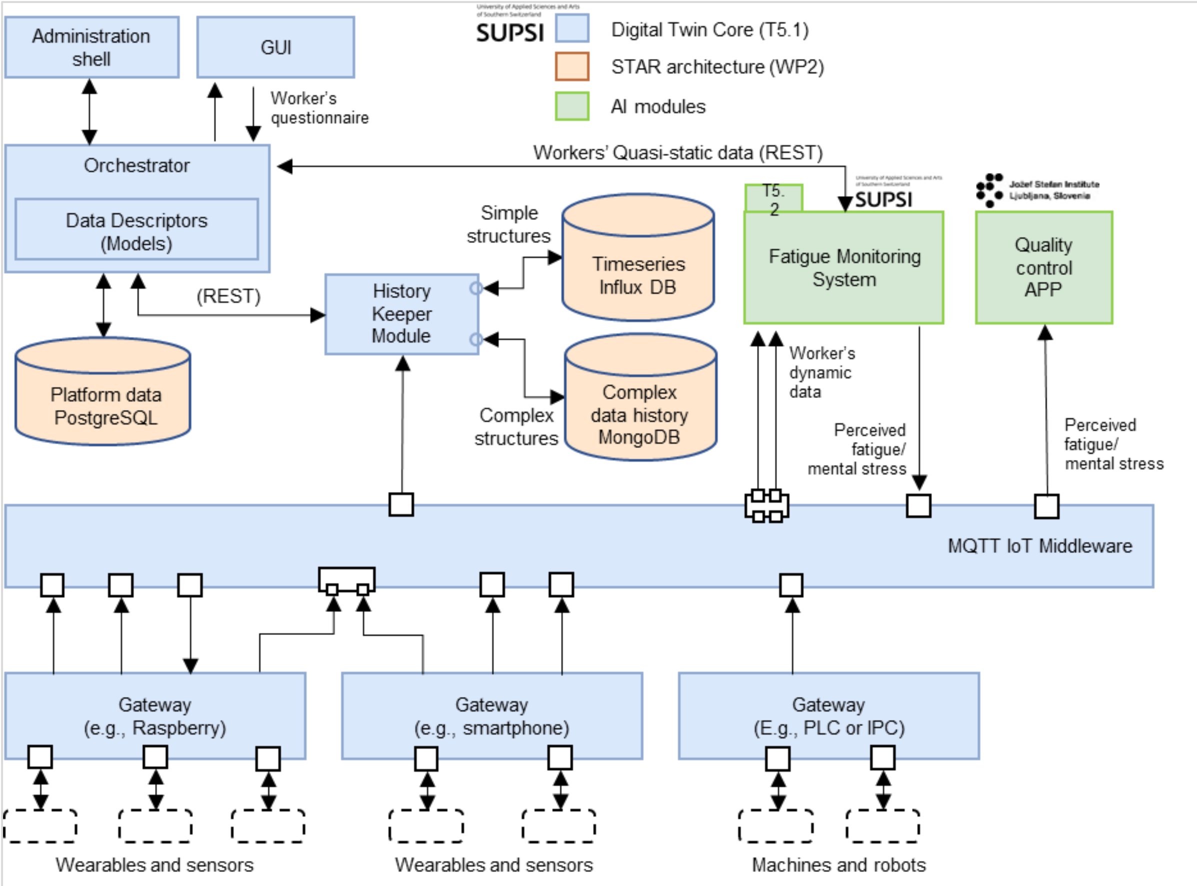

Figure 4 Integration overview: HDT, Fatigue Monitoring System and Quality control APP

Table 2 describes the main entities model has been exploited to instantiate the specific DTs for fulfilling the requirements of pilots.

Table 2 Reference model instance and main entities: Philips pilot

Name	Type	Description
UserModel	FactoryEntityModel	It describes any user performing the labelling task.
User_1; User_2;…; User_N	FactoryEntity	It describes the user performing the labelling task. Different entities are created to have specific Characteristics and Measurements to compare and to estimate the mental stress[6] of the different users based on their personal features.
Empatica_E4	DeviceModel	It describes any Empatica E4.
Polar_H10	DeviceModel	It describes any Polar H10.
Empatica_SUPSI_1	FactoryEntity	It describes the Empatica E4 used for the experiment and feeding the User's Measurements.

[6] Before archiving, data will be made anonymous (any link that will eventually be useful to identify a person will be removed from the data set, information will not be stored in the workplace or other personal information). The anonymized datasets will be maintained and will be available to other researchers to test and verify the results obtained by SUPSI.

Polar_SUPSI_1	FactoryEntity	It describes the Polar H10 used for the experiment and feeding the User's Measurements.
FatigueMonitoringSystem	FunctionalModule	It describes the module estimating the mental stress. Thanks to its FunctionalModuleOutput, it feeds the MentalStressLv. Its FunctionalModuleInput allows to read all the required inputs from the HDT.
MentalStessLV	StateDescriptor	It describes the estimated mental stress of the user.
Sex	CharacteristicDescriptor	They describe the characteristics of the user performing the labelling task collected via GUI.
birthday	CharacteristicDescriptor	
hiringDate	CharacteristicDescriptor	
jobExperience	CharacteristicDescriptor	
position in the organization	CharacteristicDescriptor	
HeartRate	MeasurementDescriptor	They describe the user's physiological parameters used by the FatigueMonitoringSystem to estimate the mental stress. They refer to the User's FactoryEntity and it are feeded by the Polar H10 and Empatica E4 devices.
RRInterval	MeasurementDescriptor	
Electrocardiogram	MeasurementDescriptor	
Electroencephalogram	MeasurementDescriptor	
GalvanicSkinResponse	MeasurementDescriptor	
SkinTemperature	MeasurementDescriptor	
OxygenSaturation	MeasurementDescriptor	
MainHandAccelerationX	MeasurementDescriptor	
MainHandAccelerationY	MeasurementDescriptor	
MainHandAccelerationZ	MeasurementDescriptor	

3.3.3 Worker Training Platform integration

Plant and machine operators are increasingly subject to external clocks in their work, brought about by intelligent automation systems, which, instead of opening up opportunities for better work experiences, pose a serious threat to worker autonomy, creativity, and satisfaction. There is an urgent need for more human-centred production systems that put the worker at the centre of data-driven, personalised training pathways to match the supply and demand for skills while promoting career development and job satisfaction. Formalising workers' skills, experience, and work preferences, as well as their physical, intellectual, and sensory abilities, is the first step in making clear the potential inherent in the workforce. To this end, skills gap analysis is fundamental to developing the workforce to meet both business needs and worker expectations and preferences.

The HDT Core Infrastructure supports data ingestion from an adaptive GUI, which provides users with an overview of their own skills gathered from the **Workers Training Platform**, in charge of recommending activities and tasks to bridge skill gaps. Batch results are sent to a dashboard representing the workers, their skills, training paths and occupations.

The **Workers Training Platform** relies on workers' quasi-static data (characteristics) stored in the HDT Core Infrastructure and collected through its dedicated GUI.

When the user, accessing to the HDT's GUI and completing the questionnaire, the answers are collected and stored in the HDT. Specific procedures for anonymization are applied.

The Human Digital Twin Core Infrastructure main functions in this validation scenario are:

- Collect quasi-static data from users performing skill assessment and make this data available to the Workers Training Platform;
- Make results available to the user through the GUI.

Figure 5 provides an overview of the integration from the point of view of the architecture.

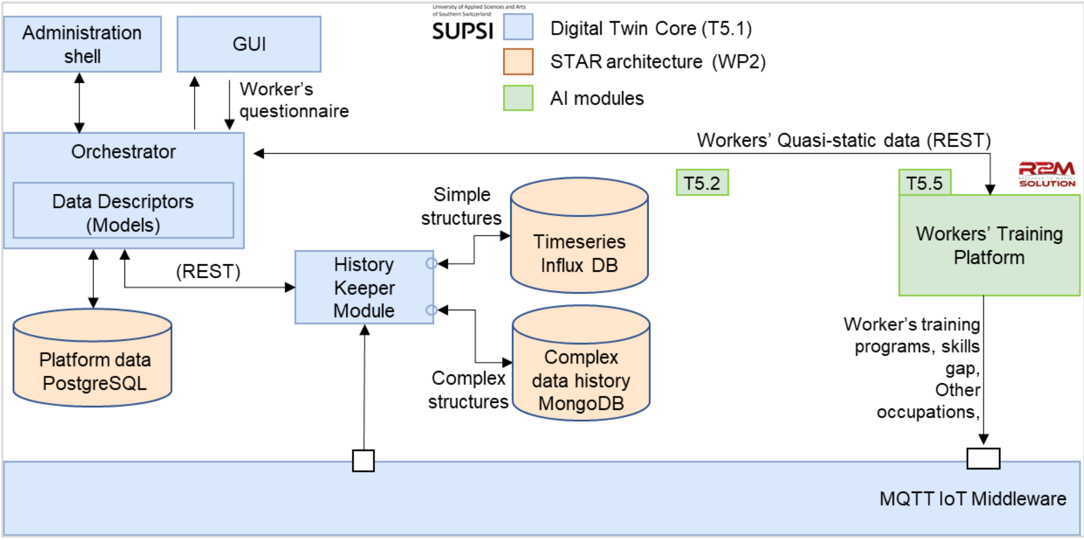

Figure 5 Integration overview: HDT and Workers' Training Platform

Table 3 describes the main entities model has been exploited to instantiate the specific DTs for fulfilling the requirements of pilots.

Table 3 Reference model instance and main entities: Worker Training Platform Integration

Name	Type	Description
WorkerModel	FactoryEntityModel	It describes any worker performing the skill assessment and training path definition.
Worker_1; Worker_2;…; Worker_N	FactoryEntity	It describes the worker performing the skill assessment and training path definition[7].
WorkerTrainingPlatform	FunctionalModule	It describes the module defining the skills required for a certain position in the organisation and the suggested training paths. Thanks to its FunctionalModuleOutput, it feeds the Skills and the Training Paths State. Its FunctionalModuleInput allows to read all the required inputs from the HDT.
Skills	StateDescriptor	It describes the required skills including their existing and desired levels.

[7] Before archiving, data will be made anonymous (any link that will eventually be useful to identify a person will be removed from the data set, information will not be stored in the workplace or other personal information). The anonymized datasets will be maintained and will be available to other researchers to test and verify the results obtained by SUPSI.

Sex	CharacteristicDescriptor	They describe the characteristics of the user performing the labelling task collected via GUI.
Sex	CharacteristicDescriptor	They describe the characteristics of the user performing the labelling task collected via GUI.
birthday	CharacteristicDescriptor	
hiringDate	CharacteristicDescriptor	
jobExperience	CharacteristicDescriptor	
position in the organization	CharacteristicDescriptor	

4 Worker's Intention Recognition Module

From M3 to M12, DFKI focused on sensor and experimental design, and data collection. From M13 to M24, DFKI collected and annotated Apple Watch data and the designed sensor data, and evaluated the annotated data with two different neural network models.

The goal of the Worker's Intention Recognition Module is to prevent collision between a mobile robot at the DFKI Smart Factory testbed and workers by recognizing worker's activities by using time-series sensor data from wearable sensors including accelerometer, gyroscope, magnetometer, and capacitive sensors. To recognize the activities, we first define the worker's activities at the testbed. The defined activities are shown in Table 4.

Table 4 Definition of worker's activities at DFKI Smart Factory testbed

Activity	Description
Press Button	A worker presses various types of physical buttons on the machine.
Slide Doorlock	To open the door of the machine on the front side, some doors have a sliding door latch. This activity is sliding the door lock to open the door of the machine.
Open Door	To check electric devices or parts in the machine and fix the problems in the machine, the worker should open the upper door or lower door of the machine.
Close Door	After checking inside the machine, the worker closes the door.
Check machine	The worker operates the task of checking and maintaining the inside of the equipment using his/her hands.
Walk	The worker moves to the next module after checking and maintaining the module of the machine.
Take Key	Several doors are locked and require the key to be opened. And, the key is plugged in another hole. Thus, the worker needs to take the key from the hole.
Rotate Key	After taking the key, the worker rotates the key to open the door. And, the worker rotates the key after closing the door.
Place Key Back	After locking the door, the worker places the key back to the original location/hole.
Check Doorlock	Before opening the door, the worker checks whether the door is locked or unlocked.
Touch Screen	A worker touches the screen on the machine.

To recognize worker's activities, we utilize two different types of wearable sensors. The first one is Apple Watch and other one is the sensor designed in the first period of T5.1 and presented in D5.1.

The Apple Watch provides only IMU sensor data with 100 Hz, while our designed sensor provides IMU sensor data and body capacitance data with 25 Hz. We put the Apple Watches on both wrists and iPhone Mini 13 on left arm. The designed sensors are attached to worker's both wrists as well. The collected data from Apple Watch and iPhone have a total of 27 channels:

- Three devices with three channels of acceleration
- Three channels of gyroscope
- Three channels of a magnet.

The collected data from the designed sensors worn on both wrists have a total of 20 channels:

- Three channels of acceleration
- Three channels of gyroscope
- Three channels of a magnet
- One channel of body capacitance data.

Based on the synchronized video data, we have annotated the user's activities on the collected sensor data as the defined activities manually. To evaluate the effectiveness of Apple Watch and the designed sensor, we designed two different neural networks named multi-channel time-series convolutional neural networks (MC-CNN) and Deep Convolutional LSTM (DeepConvLSTM). To train and validate the neural network models, we followed a leave-one-session-out scheme. In each fold, one session data from each dataset was used as the test data, another session data from each dataset was used as the validation data, and the remaining sessions (3 sessions) were used as the training data. Also, we augmented the Apple Watch data with a sliding window length of 100 (1 sec) and a step size of 4 (0.04 sec), same as the designed sensor data with a sliding window length of 25 (1 sec) and a step size of 1 (0.04 sec). To compare and evaluate the performance of the neural network models and the hardware, we adopted two evaluation metrics, which are used in various human activity recognition studies: accuracy and macro F1 score.

Table 5 Comparison results with Apple Watch and the designed sensors

	Apple Watch		**Designed Sensor**	
	MC-CNN	DeepConvLSTM	MC-CNN	DeepConvLSTM
Accuracy	72.80	69.75	67.60	63.87
Macro F1 Score	48.50	45.70	46.90	39.51

Table 5 shows the comparison results with Apple Watch and the designed sensors by using MC-CNN and DeepConvLSTM models. The results show that Apple Watch data provide better performance than the designed sensor data. To compare two hardware devices fairly, we also just used two Apple Watch devices without iPhone Mini on left arm. The accuracy is 61.33% and macro F1 score is 40.70%. The results were worse than the designed sensor data. Also, we used only IMU sensors without the capacitive sensor data to check the impact of the body capacitance data. The accuracy is 63.87% and macro F1 score is 39.50%. The capacitive sensor improved the performance of the activity recognition 4% of accuracy and 7% of macro F1 score.

To further improve the performance of the activity recognition, we plan to design the sensor with high sampling frequency and sensors on other place of worker's body. In addition, we plan to apply active learning and domain adaptation techniques.

5 Conclusions

The HDT Core Infrastructure and its components are available at https://github.com/star-eu/human-digital-twin-core-infrastructure (only for the project's partners actually involved in the pilots).

In the next period, all the solutions delivered and developed in T5.1 will be validated in the STAR pilots. These will allow also to provide relevant success-stories supporting end-users in identifying the benefits that human digital twins can provide to existing production systems, fostering the shift to Industry 5.0 paradigm.

References

[REF-01] Montini, E., Cutrona, V., Bonomi, N., Landolfi, G., Bettoni, A., Rocco, P., & Carpanzano, E. (2022). An IIoT Platform For Human-Aware Factory Digital Twins. Procedia CIRP, 107, 661-667.

[REF-02] Bettoni, A., Montini, E., Righi, M., Villani, V., Tsvetanov, R., Borgia, S., ... & Carpanzano, E. (2020). Mutualistic and adaptive human-machine collaboration based on machine learning in an injection moulding manufacturing line. Procedia CIRP, 93, 395-400.

[REF-03] Montini Elias, Bettoni Andrea, Ciavotta Michele, Carpanzano Emanuele, Pedrazzoli Paolo. (2021). A meta-model for modular composition of tailored human digital twins in production. Procedia CIRP 104C (2021) pp. 688-694