

Table of Contents

LIST OF FIGURES	4
LIST OF TABLES	6
EXECUTIVE SUMMARY	7
1. INTRODUCTION.....	8
2. DIGITAL COMPONENT (DC) OVERVIEW	9
2.1. Overall DC architecture	9
2.1.1. Digital Component – Networked Component (NC) connection	9
2.1.2. Digital Component – Open Component (OC) connection	9
2.1.3. Digital Component – Industrial Component (IC) connection	9
2.2. Main Features	10
3. DIGITAL DESCRIPTION OF OPEN COMPONENTS	11
3.1. Overview	11
3.2. Overall methodology.....	11
3.2.1. Resource Description.....	11
3.2.2. Described Resources at Use Cases	12
3.2.3. Resource Catalogue Platform	13
3.3. Integration of digital descriptions with other ODIN tools.....	14
3.4. Prototype implementation	15
3.4.1. Status of Resource Description	15
3.4.2. Status of Resource Catalogue Platform	15
3.4.3. Integration to other ODIN tools	18
4. DIGITAL TWIN	19
4.1. Overview	19
4.2. Overall methodology.....	19
4.2.1. A multi-sensor data pre-processing and fusion model	19
4.2.2. A sensor data-driven decision making.....	20
4.2.3. A digital twin-based HRC system	21
4.3. Prototype implementation	23
4.3.1. Status of each module of digital twin enabled by sensor data fusion.....	23
5. DIGITAL SIMULATION.....	24
5.1. Overview	24
5.2. Overall methodology.....	25
5.2.1. Concept phase.....	25
5.2.2. Engineering phase	29

5.2.3. Virtual commissioning	29
5.3. Prototype implementation	30
6. AI TASK PLANNER FOR DECISION MAKING	31
6.1. Overall methodology.....	31
6.1.1. Task level work re-organization.....	31
6.1.2. Modelling of Human Robot Collaborative scenarios	33
6.1.3. ODIN Resource models characteristics.....	35
6.1.4. Task planning search problem.....	35
6.1.5. Planning algorithm description.....	36
6.2. Prototype implementation	37
6.2.1. AI Task Planner User Interface (UI)	37
6.2.2. AI Task Planner integration with OpenFlow.....	38
6.2.3. AI Task Planner integration with Digital Simulation.....	39
7. CONCLUSIONS.....	40
8. GLOSSARY.....	41
9. REFERENCES.....	42

LIST OF FIGURES

Figure 1: ODIN Digital Component	8
Figure 2. Digital Component modules data flow	9
Figure 3. Overview of Resource Description and Resource Catalogue Platform concepts in ODIN ...	11
Figure 4. Use cases for Resource Catalogue Platform [8]	14
Figure 5. RCP mock-ups for different user roles	17
Figure 6. User / Browsing Resource Descriptions	17
Figure 7. User / Viewing applied interface standards	17
Figure 8. Prototype tool for integration of TAU's matchmaking result to VC - Modified from [9]	18
Figure 9. Architecture of Digital Twin enabled by sensor data fusion	19
Figure 10. CNN-based voice recognition	20
Figure 11. Workflow of hand gesture recognition	21
Figure 12. Human operator's posture estimation results: a) raising left hand; b) raising right hand; c) and d) limb occlusion and overlap, and e) cross arms.....	21
Figure 13. A framework of digital twin-based HRC.....	22
Figure 14. A Java 3D scene graph of a robot.....	22
Figure 15. 3D model of the robot.....	23
Figure 16. Phases in the production system lifecycle	24
Figure 17. Detail of the phases concept, design, and virtual commissioning	25
Figure 18. Virtual component structure in VC 4.4.....	26
Figure 19. Screenshot of Visual Components 4.4, with an assembly layout for white goods using different levels of automation. The UI provides access to the virtual models through the eCat ..	26
Figure 20. Detail of the process modelling workflow, creating the products, defining the processes, and describing flows between processes.....	27
Figure 21. High level detail of the assembly of an engine motor assembly, showing the product, parts, and assemblies.....	27
Figure 22. Creation of the manufacturing process with VC 4.4	28
Figure 23. Screenshot showing the Process Flow editor which describe products workflow through processes and the resources used	28
Figure 24. Screenshot of a modification during the engineering phase.....	29
Figure 25. Screenshot of the Communication tab with signals mapping and connectivity.	29
Figure 26. Screenshot of the initial prototype communication between Visual Components 4.4 and the OpenFlow	30
Figure 27. ODIN AI task planner module methodology.....	31
Figure 28. AI task planner module process diagram.....	32
Figure 29. ODIN Workload hierarchical model	34
Figure 30. ODIN Resources hierarchical model	34
Figure 31. Assignment of Tasks and Primitive actions.....	35
Figure 32. ODIN task planning search problem formulation	36

Figure 33. Intelligent algorithm search space	36
Figure 34. ODIN AI Task Planner UI.....	38
Figure 35. AI Task Planner and OpenFlow modules' integration	39
Figure 36. Digital simulation environment initialization.....	39

LIST OF TABLES

Table 1. Digital Component modules' implementation per pilot	7
Table 2. Production resources currently modelled with RDs in ODIN	12
Table 3. Status of RDs per ODIN Use Case	15
Table 4. Status of Each Module for Digital Twin Enabled by Sensor Data Fusion.....	23
Table 5. Criteria for Task Plans evaluation.....	33

EXECUTIVE SUMMARY

The main focus of this document is the presentation of ODIN Work Package 3 (WP3) modules' first prototypes up to M18 of the project. These components' development is based on the ODIN pilot lines requirements as already presented in deliverable D1.1.

The following topics are covered by the initial prototypes of ODIN WP3 initial components:

- Digital description of Open Components.
- Digital Twin using sensor data fusion.
- Digital Simulation for adaptable process verification.
- Digital modules for AI based decision making.

Digital Component modules will be integrated in the ODIN pilot lines depending on the needs of each use case. The Digital Component modules which are considered to be implemented inside each pilot line of ODIN until the end of the project are presented in the following table.

Table 1. Digital Component modules' implementation per pilot

		ODIN Pilots		
		Automotive	Aeronautic	White Goods
Digital Component	Digital resource description	✓	✓	✓
	Digital Twin	✓		✓
	Digital simulation	✓	✓	✓
	AI Task Planner	✓		✓

Running the M18 of ODIN, the first prototypes of the aforementioned modules have been prepared by the partners and will be presented in this deliverable alongside with their first implementation.

LEGAL DISCLAIMER

The ODIN project is co-funded by the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 101017141. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

1. INTRODUCTION

In the last few years, great efforts have been made on the use of digital tools for a variety of process/execution steps such as the design, configuration, validation, commissioning and operation of robotic systems. However, these tools are used as standalone components failing to demonstrate their efficiency potential at large scale. The following objectives have been defined in ODIN in order to develop a set of digital tools able to interact with the physical system robustly and support the aforementioned steps for efficient operation and reconfiguration. In more details:

- Develop the digital resource description of ODIN Open Components including the required information to digitally represent the OC.
- Build the digital twin using sensor data fusion to be used in the HRC systems towards safe operation.
- Deploy the digital simulation environment for designing and validation of the pilot.
- Develop a decision-making AI module able to provide task planning and dynamic reconfiguration capabilities to adapt the pilot to the changing production requirements with maximum efficiency.
- Integrate a digital simulation environment to deploy the virtual control and commissioning of the pilot line.

Based on the previously presented objectives, ODIN DC consists of the following modules:

- Digital Resource description - Result of Task T3.1.
- Digital Twin - Result of Task T3.2.
- Digital Simulation - Result of Task T3.3.
- AI Task Planner - Result of Task T3.4.

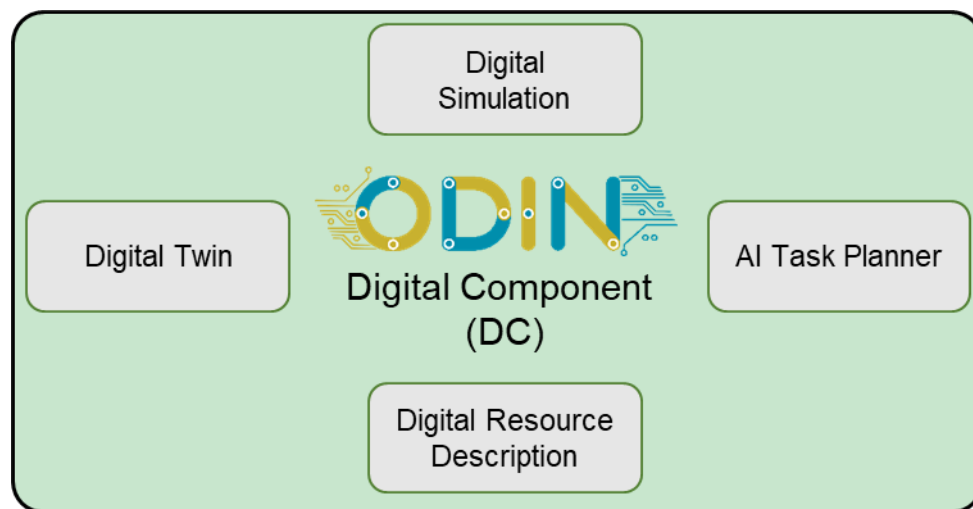


Figure 1: ODIN Digital Component

A more detailed overview of ODIN Digital Component will be presented in Section 2. Each DC modules' prototype version and a prototype implementation will be presented in the following sections of this document. Finally, conclusions and next steps for ODIN DC modules are documented in the last section of this document.

2. DIGITAL COMPONENT (DC) OVERVIEW

2.1. Overall DC architecture

ODIN focuses on the implementation of large scale pilots composed by 4 technical components namely a) Open, b) Digital, c) Network and d) Industrial components. ODIN components' modules will be demonstrated in three different industrial sectors:

- Automotive,
- Aeronautics,
- White Goods.

These components are connected for successful execution of the required operations in the selected industrial pilot lines.

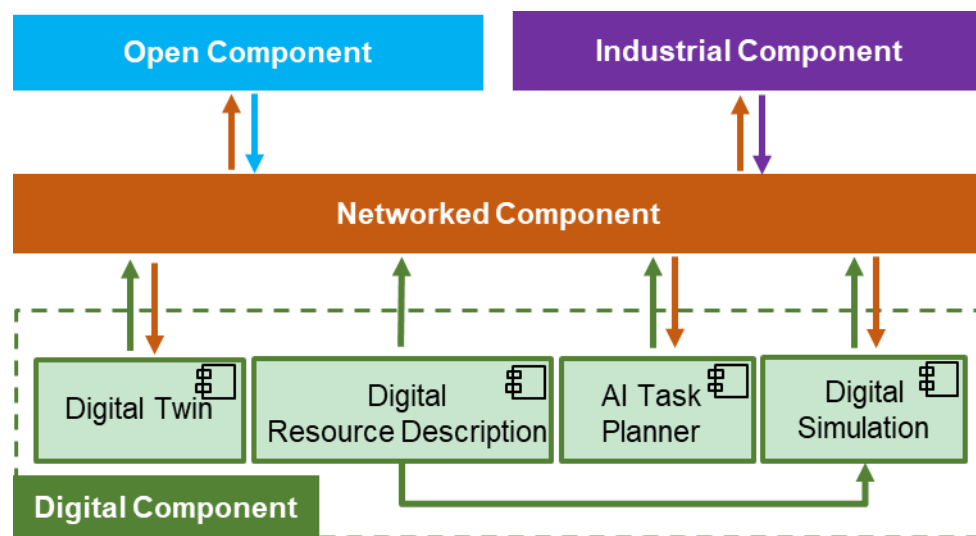


Figure 2. Digital Component modules data flow

2.1.1. Digital Component – Networked Component (NC) connection

The NC consists of an integration architecture with open interfaces allowing the safe and secure communication of all robotics hardware and controlling systems. The DC will be directly connected with the NC providing different kind of data (resource models description, assembly task plans and task assignments, robot motion commands, etc.) but also receiving information about digital simulation data, alternative task plans validation, etc.

2.1.2. Digital Component – Open Component (OC) connection

ODIN OC targets on the development, testing and integration of core robotics technologies such as mobile manipulators, reconfigurable tooling, perception systems and human interfaces using an open approach, before their deployment in industry. The digital simulation will be used for the virtual validation of the pilots. The OC also includes the Human Detection and Intention Estimation module which will provide data to ODIN Digital Twin through the OpenFlow module of NC.

2.1.3. Digital Component – Industrial Component (IC) connection

The IC focuses on the validation of the integrated solution at full scale under the actual production environments. It is responsible to provide robust data flow to the other ODIN Components for reconfiguration and optimization reasons. Despite the fact that the DC and IC of ODIN will not be directly connected, information derived from the DC will be transferred to the IC through the NC. The DC will also be connected with the IC through the virtual commissioning framework which will be described in D3.2.

2.2. Main Features

This deliverable is focused on the detailed definition of ODIN DC modules and their first prototypes development and implementation by the partners. ODIN DC consists of the following modules:

- Digital resource description
- Digital Twin
- Digital simulation
- AI Task Planner

The data flow between each Digital Component module with other ODIN components as it will be implemented in terms of ODIN is presented in Figure 2.

- **Digital resource description:** Information about ODIN resources such as size, interfaces, geometry/kinematics, business/design properties, and the resources' specific capabilities representation will be provided to the NC by the Digital resource description module. Information from this module will be also transferred to the digital simulation module visualizing the resources' geometry and their basic information.
- **Digital Twin:** This module is a digital twin of ODIN pilots including sensor data for safe HRC. It will be connected with the OC sending motion commands to ODIN robots. Additionally, the Digital Twin (DT) will provide data to the NC about the position of the operators inside the industrial shopfloor. Real time sensor data will be provided to the DT by the OpenFlow (NC).
- **Digital Simulation:** The digital simulation for pilot lines' simulation will be connected with the NC receiving the generated task plans of the AI task planner but also static and dynamic resources' location and status in order to initialize the simulation layout from the OpenFlow.
- **AI Task Planner:** This module is focused on the efficient tasks' planning and assignment to the available resources. Data transfer between the AI task planner module and the NC will be focused on the generated task plans provision to the OpenFlow and the receipt of these task plans' evaluation data by the OpenFlow after their simulated execution by the Digital Simulation.

The status of each module and its level of integration with other ODIN components will be presented in the following sections of this document.

3. DIGITAL DESCRIPTION OF OPEN COMPONENTS

3.1. Overview

Under Task 3.1, a formal comprehensive digital resource description for DCs will be developed along with a service (Resource Catalogue Platform) for publishing this information for the other ODIN tools. All this information target to serve the easy and quick system design, reconfiguration, and integration of system - first the DCs in digital world and finally OCs as a functioning system. By using the digital resource model, each individual OC is described to form a digital resource representation (DC) in catalogue of OCs. The catalogue will be used as a source of OC/DC information for other ODIN tools.

The mentioned concepts are illustrated in Figure 3 with Resource Description (RD) and Resource Catalogue Platform for storage, distribution, and processing of RDs. Integration to other ODIN tools, mainly the DCs and ICs, will happen through the interaction with the platform.

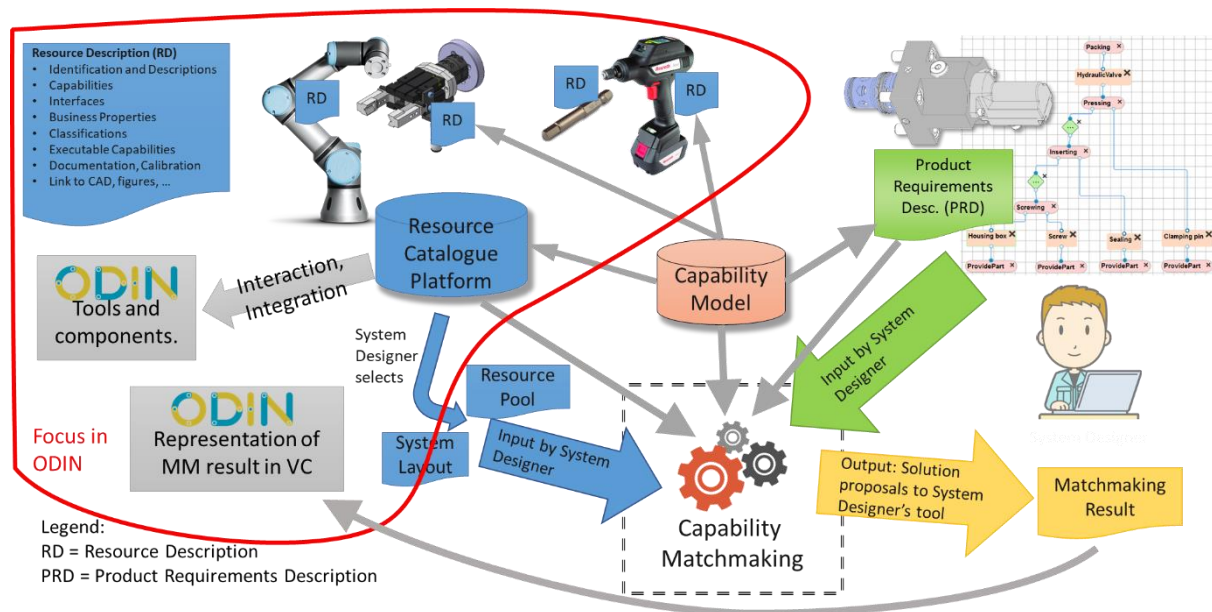


Figure 3. Overview of Resource Description and Resource Catalogue Platform concepts in ODIN

3.2. Overall methodology

The overall methodology for Resource description includes 1) Resource description data model; 2) RDs, which contain the content of production resources used in ODIN use cases; and 3) Resource Catalogue Platform.

3.2.1. Resource Description

3.2.1.1. Resource Description Concept

The resource description includes all relevant features such as size, interfaces, geometry/kinematics, business and design properties, in addition to representing the resource specific capabilities. Comprehensive mechatronic approach is followed, meaning that mechanical, electrical, control, and communication aspects are all thoroughly captured. Additionally, container model is followed, meaning that e.g. 3D models (CAD, URDF, ... etc.) are not replicated, but linked to the resource description with descriptive metadata. The main focus of resource descriptions is to comprehensively define the digital representation of production resources. This includes:

- Identification and Descriptions.
- Classifications.

- Capabilities/Skills.
- Interfaces (mechanical, supply, electrical, communication).
- Business, technical and environmental properties.
- Executable Capabilities i.e. abstraction of control interface.
- Documentation, Calibration.
- Link to CAD, figures etc.

The RD should provide the necessary information about production resources, which the other ODIN tools or e.g. system designers are needing (Figure 3). This information is however limited to data related to catalogue resource, meaning that the data which is shared among many instances of the same resource i.e. the product model or variant. Thus, data model will exclude data associated to the instances of resources or their real time data.

The existing XML-based resource description concept and associated XML schemas, developed by TAU, will be utilized as a basis for the development. The resource description concept was initially developed during EU FP6 project EUPASS [3] in the form of initial idea. Concept was defined in PhD thesis of Niko Siltala [4]. During the EU Hz project ReCaM [5] it has got its current form, which included merge of two PhD thesis (Siltala and Järvenpää) together and further developments of the data model from project needs. The data model is now assessed at the ODIN project and evaluated how the model corresponds the needs expressed here. So far, the indications show that only tiny additions are needed to the data model.

3.2.1.2. Resource Description Data Model

The data model of resource description bases on XML Schema [6]. It will define the names, quantity and relations between XML elements forming the data model. The attributes and their cardinality are also defined. In case of enumerated values, the enumerations are defined.

The capabilities follow ontology model in format of RDF/OWL. The capability model is shared online [7]. In case new capabilities are needed, they are required to be specified and added to the shared online model before they become active.

3.2.2. Described Resources at Use Cases

The tentative list of production resources to be modelled in ODIN and serving the different ODIN use cases are listed in the Table 2. The location column defines the ODIN use case, where the resource is primarily used. ODIN_All means the resource is used in two or in all three ODIN use cases. HRC Pilot means TAU's small scale pilot environment.

From the Table 2 it can be seen that totally 36 RDs are proposed. The number of modelled resources can change if some of the list are dropped out and more resources are identified at use cases and added to the list.

Table 2. Production resources currently modelled with RDs in ODIN

#	Location	Category	Manufacturer	Device
1	ODIN_Aero	AGV	Tecnalía	Robout mobile platform
2	ODIN_Aero	Gripper	TBD	Gripper for Aero
3	ODIN_Aero	ProcessTool	TBD	Drill for Aero
4	ODIN_Aero	Robot	KUKA	LBR iiwa
5	ODIN_Aero	Safety	PILZ	Safety Radar
6	ODIN_Aero	ToolBit	TBD	Drill bits for Aero
7	ODIN_All	Camera	Microsoft	Kinect 2

#	Location	Category	Manufacturer	Device
8	ODIN_All	Camera	Microsoft	Azure Kinect
9	ODIN_All	Camera	Roboception	RC Visard 65
10	ODIN_All	Camera	Roboception	RC Visard 160
11	ODIN_Automotive	AGV	AIC	Mobile platform
12	ODIN_Automotive	AGV	Comau	Agile 1500
13	ODIN_Automotive	Conveyor	DGH	Conveyor - curve
14	ODIN_Automotive	Conveyor	DGH	Conveyor - straight
15	ODIN_Automotive	Gripper	LMS	Gripper for motor
16	ODIN_Automotive	Gripper	LMS	Gripper for gearbox
17	ODIN_Automotive	Pallet	TBD	Pallet for engine
18	ODIN_Automotive	ProcessTool	TBD	Screwdriver for automotive
19	ODIN_Automotive	Robot	Comau	Aura Cobot
20	ODIN_Automotive	Robot	Comau	Racer5
21	ODIN_Automotive	ToolBit	TBD	Tool bits for screwing
22	ODIN_WhiteGoods	Fixture	LMS/Wingman	Tool changer holder
23	ODIN_WhiteGoods	Gripper	LMS/Schunk	Magnetic gripper for WhiteGoods
24	ODIN_WhiteGoods	Gripper	LMS	Vacuum gripper for WhiteGoods
25	ODIN_WhiteGoods	Gripper	LMS/Festo	Festo flexible gripper for WhiteGoods
26	ODIN_WhiteGoods	Robot	UR	UR 10 collaborative
27	ODIN_WhiteGoods	Safety	PILZ	Laser scanner
28	TAU Small-scale pilot	Gripper	TAU	TAU vacuum gripper for safety mat
29	TAU Small-scale pilot	Gripper	TAU/Schunk	TAU_Schunk_ThreeFinger gripper
30	TAU Small-scale pilot	Gripper	TAU/Schunk	TAU_Schunk_TwoFinger gripper
31	TAU Small-scale pilot	Other	TBD	Projector
32	TAU Small-scale pilot	ProcessTool	TAU	Virtual dispensing tool
33	TAU Small-scale pilot	Robot	ABB	IRB 4600
34	TAU Small-scale pilot	Robot	ABB	IRB 4600

3.2.3. Resource Catalogue Platform

The second part of the task is to create a platform for the catalogue to search and distribute the resource descriptions. The main identified use cases for Resource Catalogue Platform (RCP) are illustrated in Figure 4. The use cases are reflected through two main actors – *Resource user* and *Resource provider*. The examples of former are system designers, system re-configurators and system integrators. It can be also another software, which uses this service.

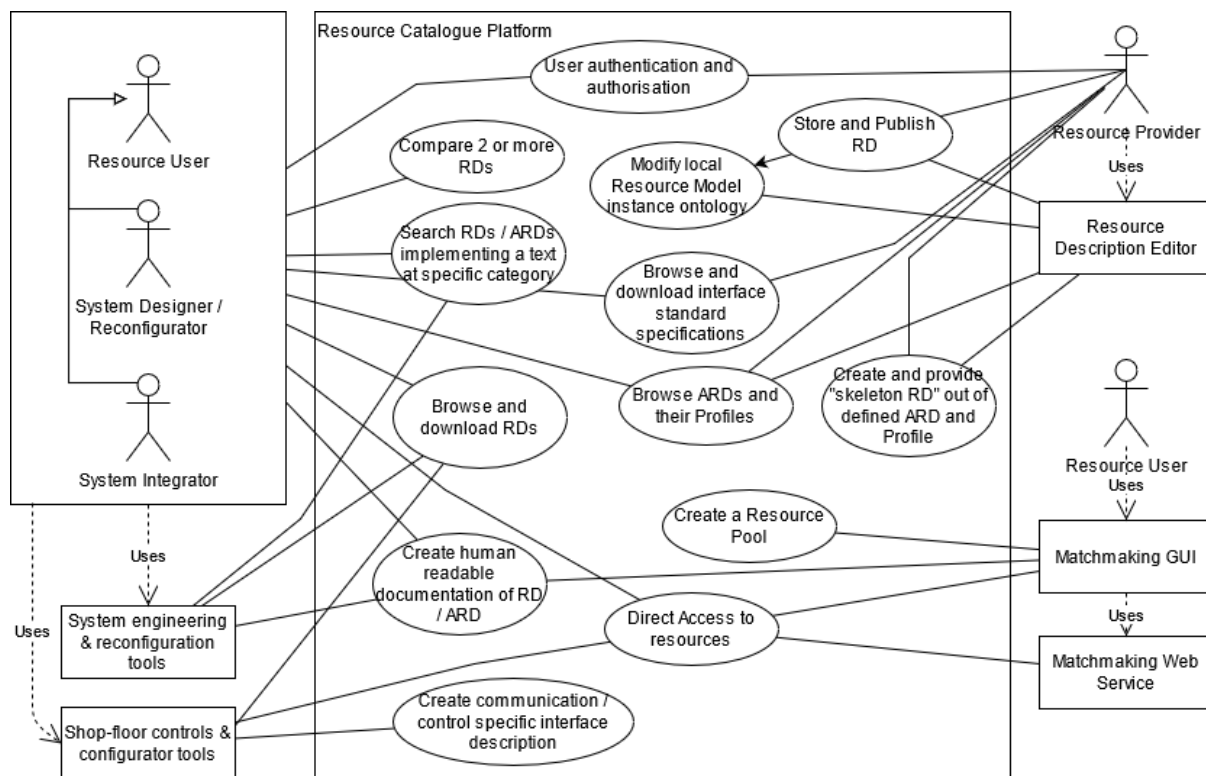


Figure 4. Use cases for Resource Catalogue Platform [8]

The main use case at RCP for Resource user is *Browse and download RDs*. The previous can be appended with search (*Search RDs / ARDs implementing a text at specific category*) and *Create human readable documentation of RD / ARD*. Interface standard information can be browsed and downloaded if needed, if those are not coming from international or company specifications, which are published already elsewhere.

The main use cases for the production resource provider are the creation of RD and *Store and Publish RD*. The former can be supported by *Browse ARDs and their Profiles*; *Browse and download interface standard specifications*; and *Create and provide "skeleton RD" out of defined ARD and Profile*. These all are helping the resource provider to make correct selections with the RD definition e.g. for the selection of right interface specification.

There exists an outdated version of Resource Catalogue Platform. It is monolithic implementation basing on Java Server Pages (JSP) and partial database solution. There are several issues: Terminology has got outdated through data model changes; DB solution is partially updated, but application is has not evolved at same pace and is not compatible anymore; service does not offer RESTful API for application integration; application and data are tightly connected and packaged (i.e. if RD data is changed, application recompilation is needed); features needed by resource provider does not exist e.g. for RD upload; and currently there exist a lot of libraries, which helps on implementation and maintenance of the application. But most of all, the past application is basic example of gradually accumulated features and code, without clear overall plan, ending up to intertwined "spaghetti" code, which is impossible to maintain. Thus, refactoring and/or rebuilding the whole application from scratch is proposed.

3.3. Integration of digital descriptions with other ODIN tools

The Resource Descriptions will work as source of resource information for other ODIN tools and components. The resource catalogue platform will provide a REST API for accessing this information. It will provide various end-points to interact with the platform and the provided data. Some end-points will provide specific data or view related to RD, but also the whole resource description data will be provided as complete RD. The implication for the receiver is that it needs to understand and parse the

full RD data model. The first specification version of API includes 56 end-points. Refinement will be done at next phases of ODIN project with the end-point parameters and how this interface is described in terms of OpenFlow.

Other tools can access resource catalogue platform and resource data through the OpenFlow. It is still a bit open will the OpenFlow provide or relay the whole RCP REST API, or only a part of it. Other ODIN tools will be connected to OpenFlow. Perhaps, it will be enough to relay some core end-points and functionalities, and an access point of the resource catalogue platform. Then other tools can use the access point information to connect at RCP and to directly retrieve all necessary information from the catalogue.

Integration to Visual Components: A prototype tool has been prepared for integration of TAU's matchmaking tool result (Figure 3) and resources (RDs) listed on it to Visual Components (VC). The tool is aimed to work as plugin in Visual Component's software. The tool gets a matchmaking result as input and after operator clicks the trigger button the process will start. First matchmaking result is processed through, then corresponding resource ids in VC are found out from mapping table, and finally resources are placed on canvas at VC according to the matchmaking results, grouped per each match. Results of this development are reported in [9].

3.4. Prototype implementation

3.4.1. Status of Resource Description

The status of RD definition is summarised in the Table 3. Most of the ODIN use cases related RDs are not started or in initial data collection phase. The resources at TAU's small scale pilot environment (HRC pilot) are prepared first. These are working as examples for the rest both at the data collection and as end result of RD.

Table 3. Status of RDs per ODIN Use Case

Row Labels	Count of Device
ODIN_Aero	6
00Not Started	6
ODIN_All	4
00Not Started	4
ODIN_Automotive	11
00Not Started	11
ODIN_WhiteGoods	6
00Not Started	5
15RD_Published	1
TAU Small-scale pilot	7
00Not Started	1
01DataCollection	2
05RD_XML_started	4
Grand Total	34

3.4.2. Status of Resource Catalogue Platform

The first version of Software Requirement Specification is completed.

System architecture is designed, through a few iterations. There might be still some changes on that. E.g. Database solution has been iterated from relational database (DB) MariaDB to MariaDB and BaseX (for storing XML data) to DB2 (which can store both). Currently main lines of architecture are RESTful

backend (BE) server solution on top of Spring Boot, Apache Tomcat; using DB2 as database solution. Front-end (FE) implemented with ReactJS as single page application.

3.4.2.1. Back-End

The REST API is on specification and development. First version including 56 end-points has been defined. Refinement will be done at next phase with the end-point parameters and how this interface is described in terms of OpenFlow. Initial pattern for description is provided by INTRA.

As the application is a migration case from old outdated application, the three phases migration is proposed.

1. Stage1: (on-going)
 - a. Refractor the BE (biggest effort)
 - b. Eliminate JSP by replacing that with Freemarker
 - c. The application is still the monolith application.
 - d. REST service is not ready
 - e. FE architecture and GUI mockup design
2. Stage2:
 - a. Eliminate Freemarker (zero effort)
 - b. Provide REST APIs for the FE and other application integration. (Mediocre effort)
 - c. FE implementation
3. Stage3:
 - a. The other requirements (TBD)

3.4.2.2. Front-End

Mock-ups of Resource Catalogue Platform web pages and browsing features are prepared on Figma online design tool. The service will have different user roles. Each role and actions they can make will be differentiated by different colour theme illustrated in Figure 5. The four themes / pages from left to right are public, user (logged in user), resource provider and resource manager (both will share the green theme), and administrator.

Figure 6 illustrate user view for browsing, searching and filtering of Resource Descriptions. User can set various criteria from predefined categories (Vendors, category, ... etc.) or open text search (filename, model & type) or applying open text search over whole data. Different actions can be made for the selected RD – viewing it in HTML or opening/downloading it in original format.

Figure 7 illustrates the view of applied interface standards at the RDs. The list can be filtered and viewed the specific RDs which are implementing the standard.

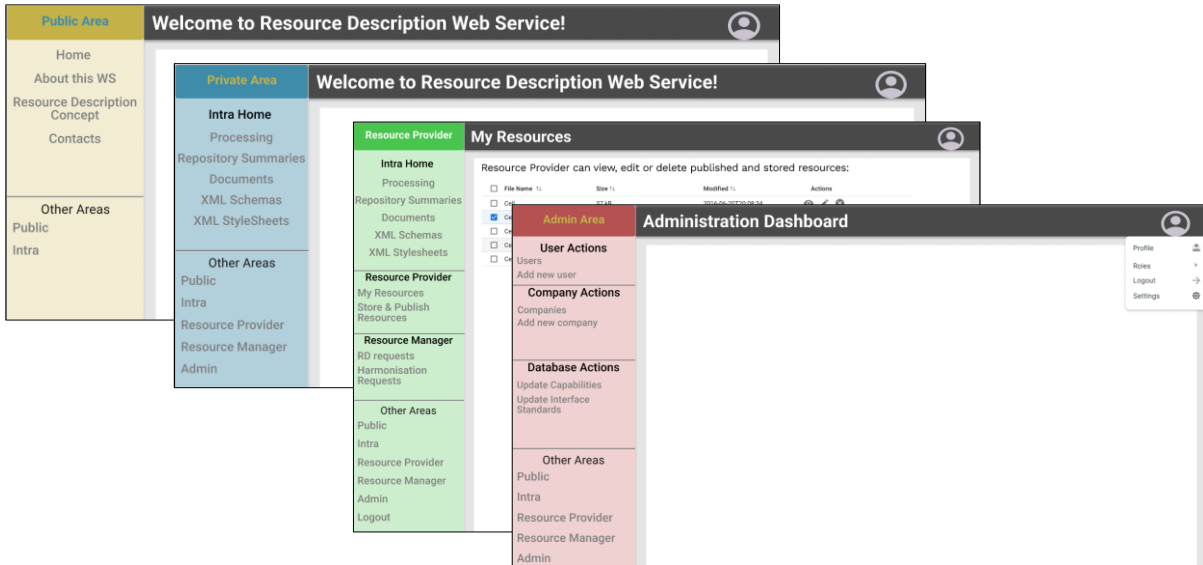


Figure 5. RCP mock-ups for different user roles

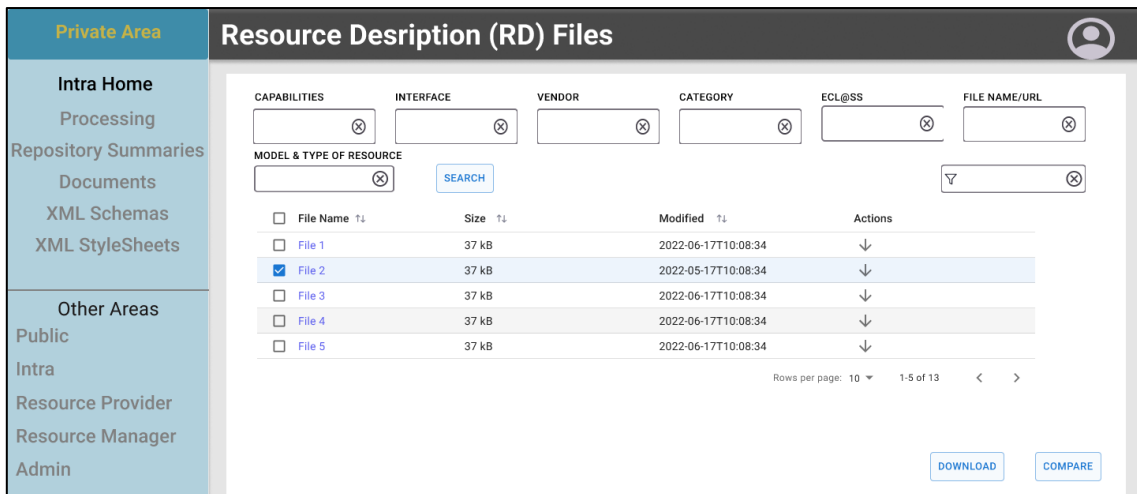


Figure 6. User / Browsing Resource Descriptions

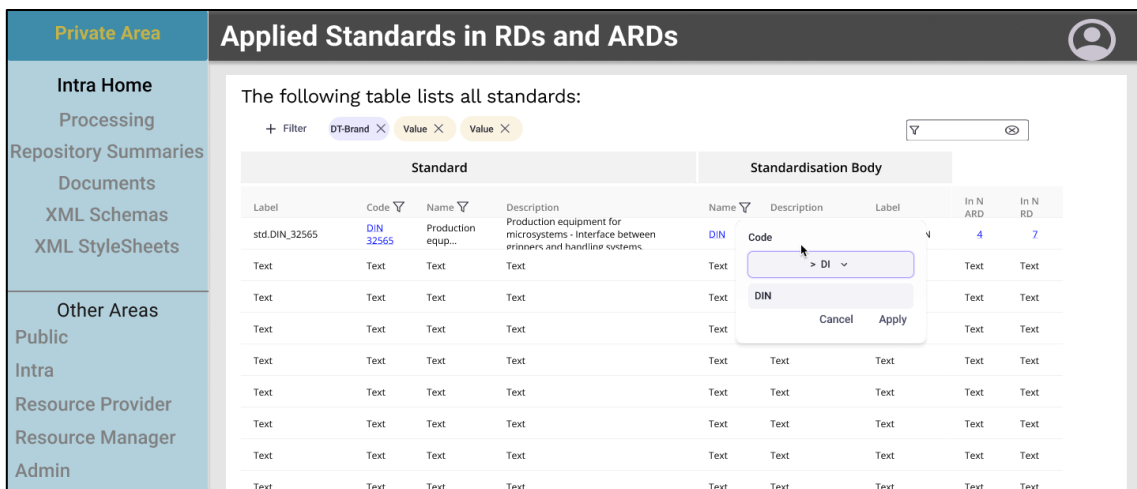


Figure 7. User / Viewing applied interface standards

3.4.3. Integration to other ODIN tools

Figure 8 illustrates the final result in Visual Components after reading in TAU's matchmaking result with the help of a prototype integration tool. This tool is aimed to work as a plugin added to VC's GUI. It is currently an executable file which starts the VC and performs the required actions. I.e. the implementation as plugin is pending with appropriate UI at VC side. Also, work is needed with the mapping and how it is performed. Eventually, the mapping data should be provided from individual RD, from where the mapping data should be retrieved. The mapping data could be made available via generated mapping file, database, or service endpoint at RCP.

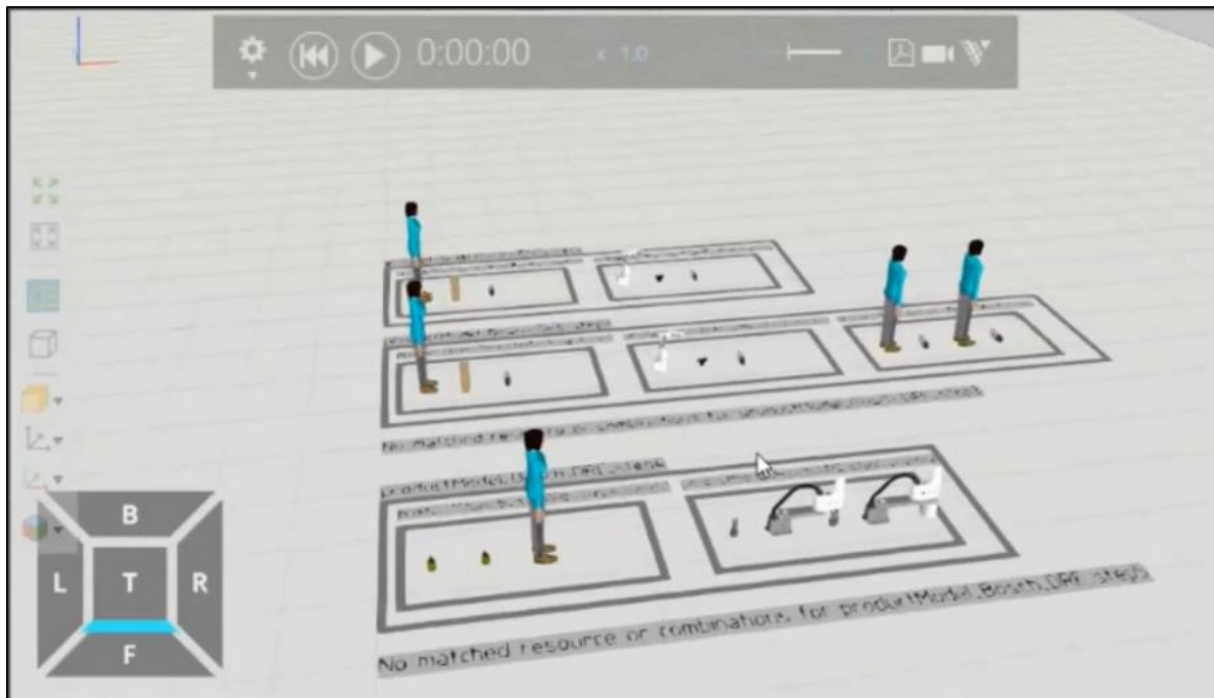


Figure 8. Prototype tool for integration of TAU's matchmaking result to VC - Modified from [9]

4. DIGITAL TWIN

4.1. Overview

The overview architecture of digital twin (DT) system enabled by sensor data fusion is presented in Figure 9, which is composed of four modules, namely: a) pre-processing module, b) a data fusion module, c) a decision-making module and d) a digital twin module. Firstly, a pre-processing module for multi-sensor data will be developed to extract key information from sensors respectively into a unified format. Second, a data fusion module, which fuses pre-processed data from various sensors, will be designed based on deep learning technology, Thirdly, a decision module using HRC status classification, recognition and prediction algorithms will be designed for generating commands for the HRC system towards safe operations. Finally, a digital twin of the HRC system will be presented by integrating the pre-processing module, the data fusion module and the decision module and tested in the corresponding ODIN pilot line.

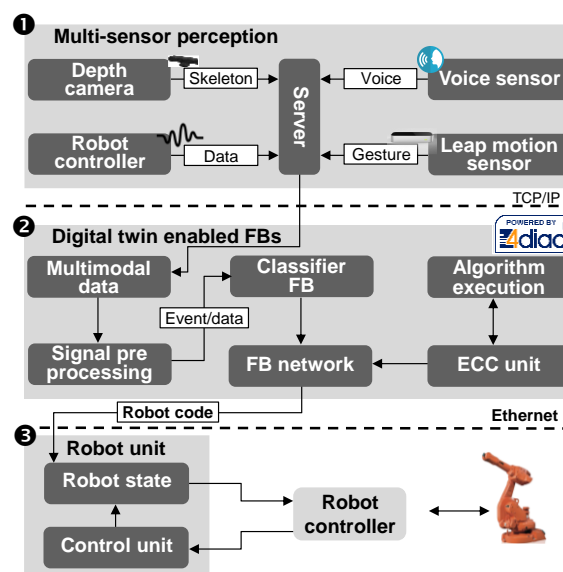


Figure 9. Architecture of Digital Twin enabled by sensor data fusion

In such a collaborative setup, hand body skeleton, gestures, voice commands, and are collected from a Leap-motion sensor, sensorless haptics, a sensor headset, and a microphone sensor, respectively. Then, event-driven FBs are used for the fusion and classification of the collected data, but also the encapsulation, invocation, and execution of the control algorithms and assembly operations. Within this context, the collected multimodal commands are served as triggering events to the execution of the FB network. Then, the outputs of FBs are sent to control the robot and execute the assembly tasks.

4.2. Overall methodology

The overall methodology of digital twin (DT) system enabled by sensor data fusion includes 1) a multi-sensor data pre-processing and fusion model; 2) a sensor data-driven decision making which consists of algorithm development of classification, recognition and prediction; 3) a digital twin-based HRC system.

4.2.1. A multi-sensor data pre-processing and fusion model

The interface design starts with data collection by diverse communication channels between a human and a robot. The voice instructions are collected from a microphone that is connected to an external computer via Bluetooth. The collected data are saved as a raw audio file. However, in a dynamic and acoustically noisy HRC environment, voice recognition is not always reliable. Therefore, gesture commands as a nonverbal communication can work as an alternative solution for robot control. Hand gestures are recorded from the Leap-motion sensor running a Python-based environment, and the

positions of fingertips and the palm centre in three-dimensional space are used to calculate the hand gestures.

4.2.2. A sensor data-driven decision making

A brief description of multi-modality recognition is introduced as follows.

Convolutional Neural Network (CNN) for voice recognition: CNN is a popular deep learning model that can be adopted for sequential data recognition tasks such as speech command recognition. Before using CNN to process the voice commands, the voice command dataset is transformed into 2-dimensional spectrograms by Fast Fourier transform. The convolution operation is formulated as a function of input map and feature map in which a ReLU activation function is adopted. In the formulated convolution function, the weights of the input map are shared among each convolution neuron, and Max-pooling outputs the maximum value of each of the local neighbour. Max-pooling makes each feature map invariant to local translations in the input map, which is also proven to be useful in CNN. In model training, the categorical cross-entropy is defined as the cost function. The voice commands used in the multimodal HRC are classified and labelled as left, right, on, off, up, and down. The recorded audio that is essentially a 1-D vector of strength signals is transformed into a 2-D matrix that can be treated as a single-channel image. The details of CNN-based voice recognition as shown in Figure 10 is described as follows: In a convolution layer with a ReLU activation function, a filter with a 3×3 image matrix slides over the input map to produce a feature map, and 25 filters, 1 stride length, $177 \times 98 \times 1$ input shape are adopted. Then, max-pooling operations with a 3×3 feature map and 2×2 strides output the maximum value of each of the local neighbour. The fully connected layer is to use the obtained features for classifying the input image into various classes based on the training dataset where the parameters of the dense operations are shown in Figure 10. Then, the categorical cross-entropy is selected as the loss function in the training model. In a noisy environment of HRC, the recognition accuracy is reliable.

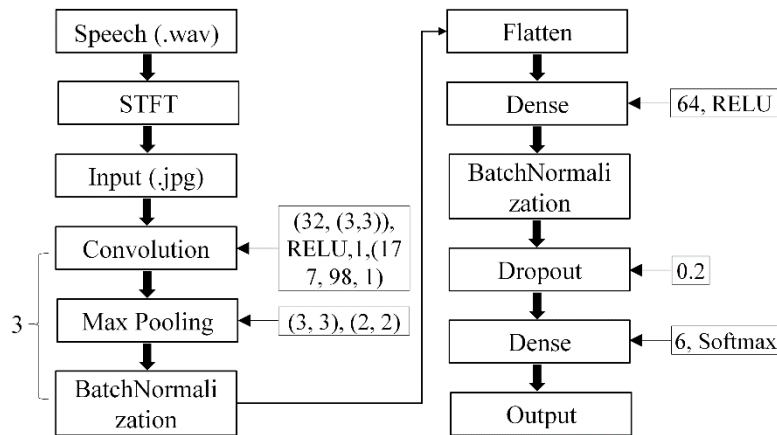


Figure 10. CNN-based voice recognition

Leap-Motion-based hand motion recognition: The Leap Motion can be connected with the mobile device (i.e. personal computer) for capturing hand motion and the data output of Leap motion offers a real-time representation of human hands with a series of timestamps, the finger positions, and hand position. To implement hand motion recognition in HRC, the Leap motion dataset built based on a number of hand-waving are labelled and multi-category classification is developed, while other random motion datasets collected from the sensors are labelled as negative. The algorithm that is to identify the hand-waving motion is used to train the labelled dataset. The hand motion dataset consists of a number of sequences of hand motions with six different categorical labels. The Leap Motion Controller captures the direction and orientation of key hand joints and bones frequency of 100 Hz. The hand skeleton models can be built by tracking and capturing hand bones and joints. In each time stamp of the hand motion, 64 hand motion features which are defined by the parameters from the skeletons are captured. In hand gesture recognition, long short-term memory (LSTM) as a recurrent neural network shows the capability of handling time-based sequence, as shown in Fig. 4b. At a time, an LSTM unit that is treated

as a memory cell contains an input gate, an output gate, and a forget gate. Through the control of the gates, long-time information can be realised, reserved, and updated. In the LSTM model, the input and output sizes of the LSTM unit with 1 stride length are 30 and 32, respectively, and the fully connected layer with a 6-D output size and a softmax activation function outputs a vector with the same dimensions. The workflow of hand gesture recognition is described in Figure 11.

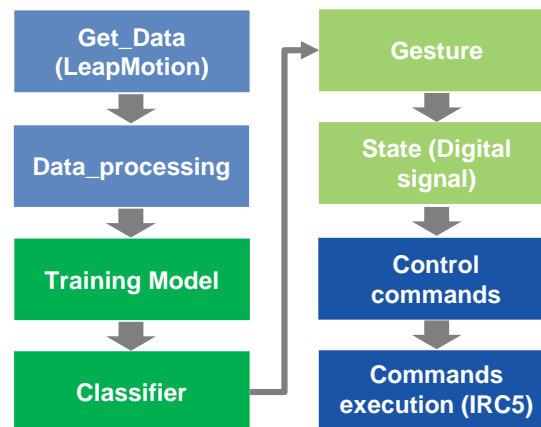


Figure 11. Workflow of hand gesture recognition

Human body skeleton and motion detection: A human body skeleton and motion detection algorithm is presented in ODIN for recognizing human postures and operators' motion inside a HRC environment. More details about the Human body skeleton and motion detection algorithm are presented in D2.2 submitted on M18 of the project.

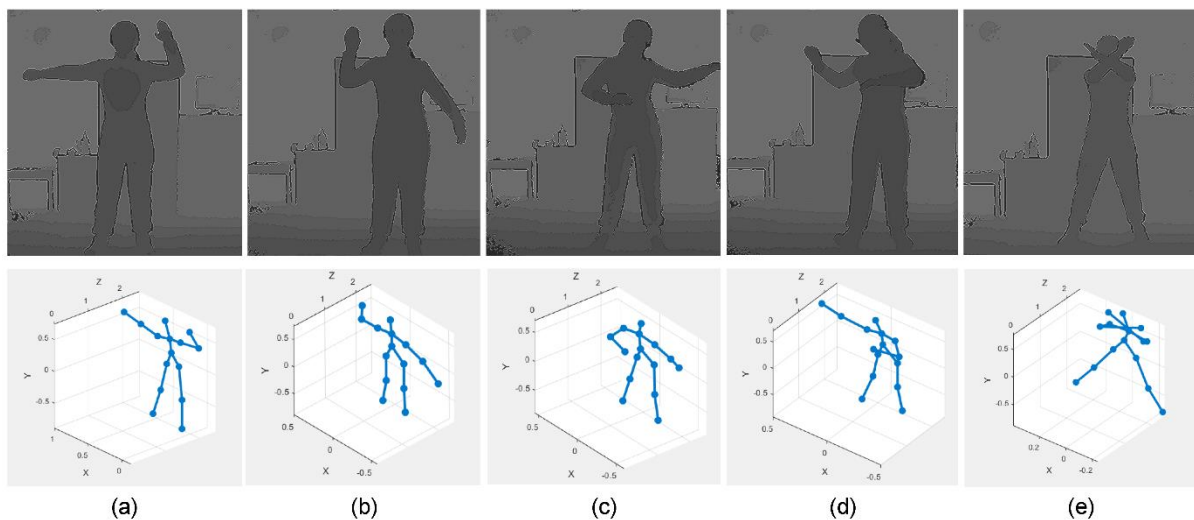


Figure 12. Human operator's posture estimation results: a) raising left hand; b) raising right hand; c) and d) limb occlusion and overlap, and e) cross arms.

4.2.3. A digital twin-based HRC system

This section describes the framework of the DT to connect virtual and real HRC cells as shown in Figure 13. DT works as an enabling technology used for creating a mock-up of HRC. In parallel, the components for robot operations are implemented by the use of function blocks that are an International Electrotechnical Commission (IEC) 61499 standard [14] for distributed industrial processes and control systems. The DT functionality of the HRC system built includes several components for sensor-driven 3D model, communication between physical and virtual HRC cells, robot control and task/command execution. The sensor-driven 3D model provides the digital presentation of objects in a HRC cell and it is fundamental to mirror replication of physical settings. To render a 3D model of a robot assembly station with geometry and kinematics, a scene graph Java 3D model embedded with behaviour control

nodes, as shown in Figure 14. It is adopted to represent a physical object of the HRC cell. The model's behaviour is controlled by sensors of its physical counterpart. It is represented and organised by a parent-child tree structure and contains a virtual universe (offers a grounding for the scene) and a set of branch graphs. Each branch is rooted by a BranchGroup node that specifies the position, orientation, and scale of the corresponding object. The scene graph is a "tree" structure that contains data arranged in a hierarchical manner. The scene graph consists of parent nodes, child nodes, and data objects. The parent nodes, called Group nodes, organize and, in some cases, control how Java 3D interprets their descendants. Group nodes serve as the glue that holds a scene graph together. Child nodes can be either Group nodes or Leaf nodes. Leaf nodes have no children. They encode the core semantic elements of a scene graph- for example, what to draw (geometry), what to play (audio), how to illuminate objects (lights), or what code to execute (behaviours). Leaf nodes refer to data objects, called NodeComponent objects. NodeComponent objects are not scene graph nodes, but they contain the data that Leaf nodes require, such as the geometry to draw or the sound sample to play. Using this approach, a 3D model of the robot combined with kinematics is generated in Figure 15. For network transport, the DT system uses a Transmission Control Protocol (TCP) for the communication between the robot and the server, while HTTP streams are used for data sharing from the server to remote users. Sensor data from the physical cell with the robot are passed through OpenFlow. The DT is considered to be integrated with the OpenFlow in order to retrieve any data required for visualization.

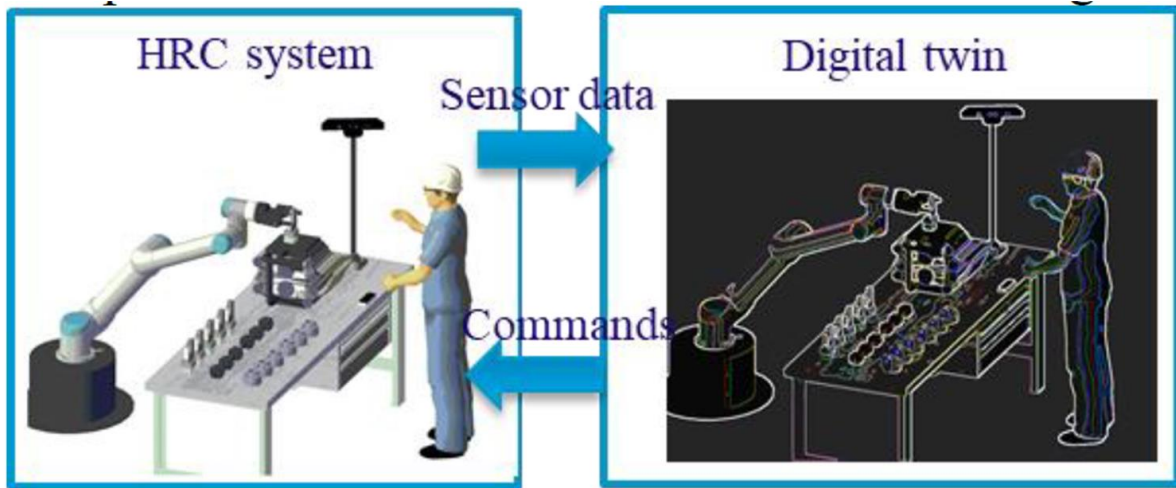


Figure 13. A framework of digital twin-based HRC

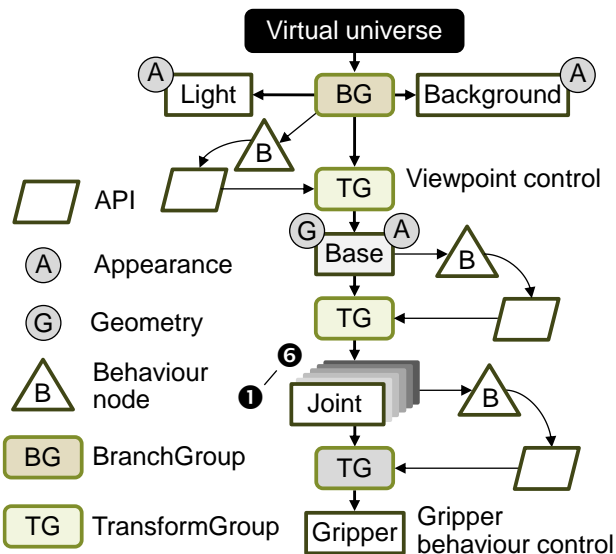


Figure 14. A Java 3D scene graph of a robot



Figure 15. 3D model of the robot

4.3. Prototype implementation

4.3.1. Status of each module of digital twin enabled by sensor data fusion

This section describes the status of each module used for digital twin enabled by sensor data fusion. The status of four modules that are a pre-processing module, a data fusion module, a decision-making module, and a digital twin module as well as related resources are summarised in Table 4. The resources at KTH's small scale pilot environment have been prepared as a first step. These are working as examples for the rest, both at the data collection and as end result of the digital twin-based HRC system.

Table 4. Status of Each Module for Digital Twin Enabled by Sensor Data Fusion

Module	Status
Data Preprocessing	Started
Data Fusion	Started
Decision-Making	Started
Digital twin of the HRC system	Started

4.3.2. Status of digital twin enabled by sensor data fusion

The first version of Software Requirement Specification is completed. A system architecture is designed through a few iterations. There might be still some changes on that and some technical details might be modified in the future during real execution. For example, human body skeleton and human motion detection in the white goods case will focus on the perspective of human operators. For the data pre-processing module, multi-sensor devices are deployed to collect sensor data. For example, the Kinect sensor is used to capture the raw data of human body skeleton, followed by a processing method for building human body skeletons. The programs used for data pre-processing is mainly developed in Python language. The data fusion is enabled by function blocks. Event-driven FBs embedded with smart algorithms can be applied as the solution to the needs of the human-robot interface. Algorithms used for command processing and classification, as well as execution of control commands can be encapsulated in the FBs. Then, these high-level multimodal commands serve as triggering events to the FBs for robot control. The decision module is supported by deep learning technology. Finally, the digital twin of the HRC system will integrate the previous modules together used for ODIN pilot cases. During this process, the digital components and digital simulation for pilots' simulation and digital twin development are provided by TAU and VIS, respectively.

5. DIGITAL SIMULATION

5.1. Overview

The utilization of digital simulations in the manufacturing domain has been growing continuously since the decade of 1980s [10]. Nowadays, the utilization of digital simulation is even more relevant in a manufacturing scenario influenced by trends such as globalization, mass production, mass customization, repurposing (which has considerably increased, due to pandemic situations such as the current COVID-19), etc.

Digital simulation comprises a set of technological tools and methods for the successful implementation of digital manufacturing. Its correct deployment along the life cycle of product and production systems, enables the validation of products, processes and systems design and configuration [10, 12]. Despite the fact that the digital simulation started in the aerospace and automotive manufacturing domains, the advance of the computer technology extended this application to other areas and the economic viability of the solutions accelerated the extension to other manufacturing domains [12]. The development of the ICT technologies and its combination extended the digital simulation scope seamlessly be integrated in solutions that have been reached the whole manufacturing industry.

The manufacturing system lifecycle comprises the phases described in Figure 16. The decommissioning phase is not going to be comprised in the research of ODIN. However, it is kept in the overall view as the availability of digital models and the acquisition of historical data during the production that will be maintained and facilitated for future reuse of the digital information in case of reconfiguration and reusability of the manufacturing systems that the production line contains. This deliverable presents the current work in the digital simulation, covering the concept phase (section 5.2.1), the engineering phase (section 5.2.2), and the virtual commissioning (section 5.2.3) which will be briefly introduced as it will be detailed presented in the upcoming deliverables D3.3 and D3.4.

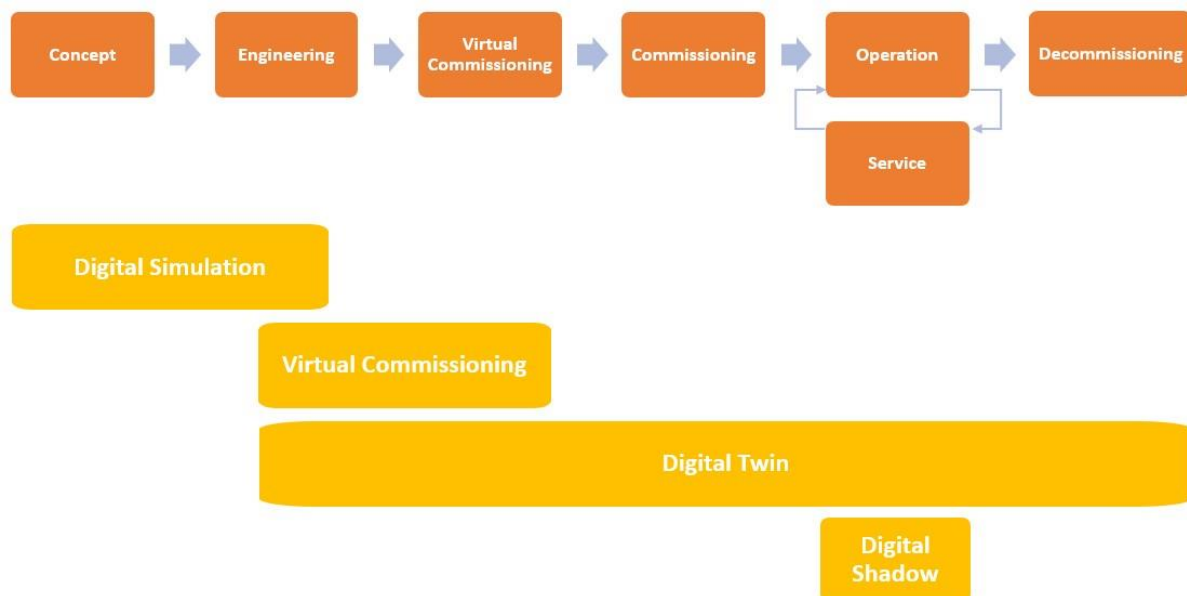


Figure 16. Phases in the production system lifecycle

Two scenarios are devised within ODIN pilots at the moment of creating the digital simulation, greenfield and brownfield scenarios. In the greenfield scenario the production system is already deployed in the factory requiring modelling the systems as-is. In the brownfield scenario the production system is not deployed, and the digital simulation is built based on existing systems in the facilities and is more open to incorporate new technologies.

The digital simulation within ODIN pilots has used process modelling feature. Process modelling is a feature included in Visual Components 4.4 (VC 4.4) which is further developed within ODIN with simulation components to reach the ODIN simulation requirements. The digital simulation of the three

pilots deployed within ODIN (Aeronautics, Automotive and White Goods) are using process modelling, as starting point for creating the simulation, building the respective simulation models.

5.2. Overall methodology

The approach proposed in the development of the digital simulation within ODIN is following the PDCA (Plan-Do-Check-Adjust) methodology. The PDCA is defined at [11] as an iterative design and management method used for the control and continual improvement of processes and products. The utilization of this approach while creating the pilots' digital simulation facilitates its implementation as it provides an incremental way of work and can be used during the three phases considered within ODIN (Figure 17), of the system life cycle for the digital simulation.

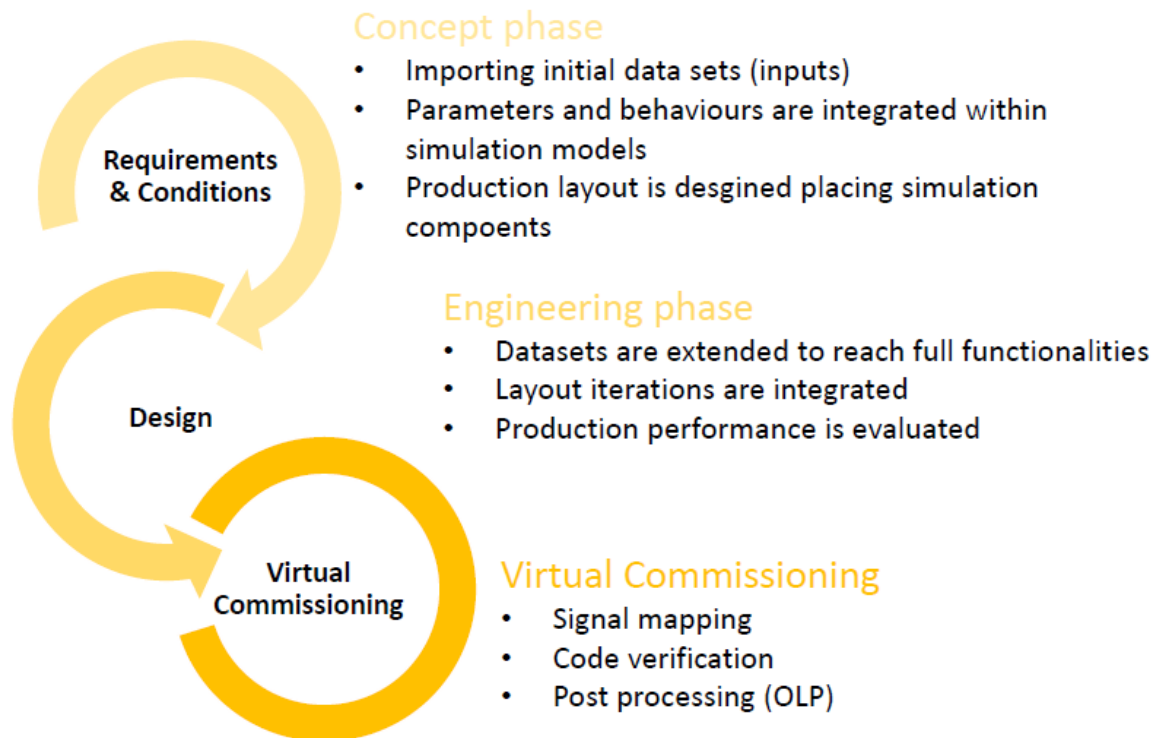


Figure 17. Detail of the phases concept, design, and virtual commissioning

5.2.1. Concept phase

During the concept phase the pilot requirements are compiled, including functional and operational ones. Existing equipment is modelled within VC 4.4 in case that no virtual components are available in the electronic catalogue (eCat). The virtual component in VC 4.4 has the structure described in Figure 18. During the concept phase, the component is modelled starting from the parameter which defines it, CAD data is imported to add behaviours and define features which conforms the virtual component to match functional and operation requirements. The manufacturing systems can have different levels of complexity in the virtual environment. They can be represented as a virtual component or a set of virtual components which provide a determined functionality. Once the manufacturing systems are created within the virtual environment, the layout is built by adding the virtual systems to match production requirements.

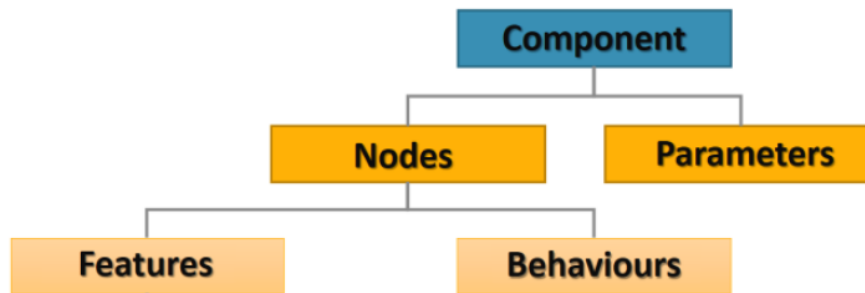


Figure 18. Virtual component structure in VC 4.4

The user interface provided in VC 4.4 is very intuitive and the virtual components created in the first step can be arranged within the virtual space as in the real factory (Figure 19), to build the virtual plant layout.

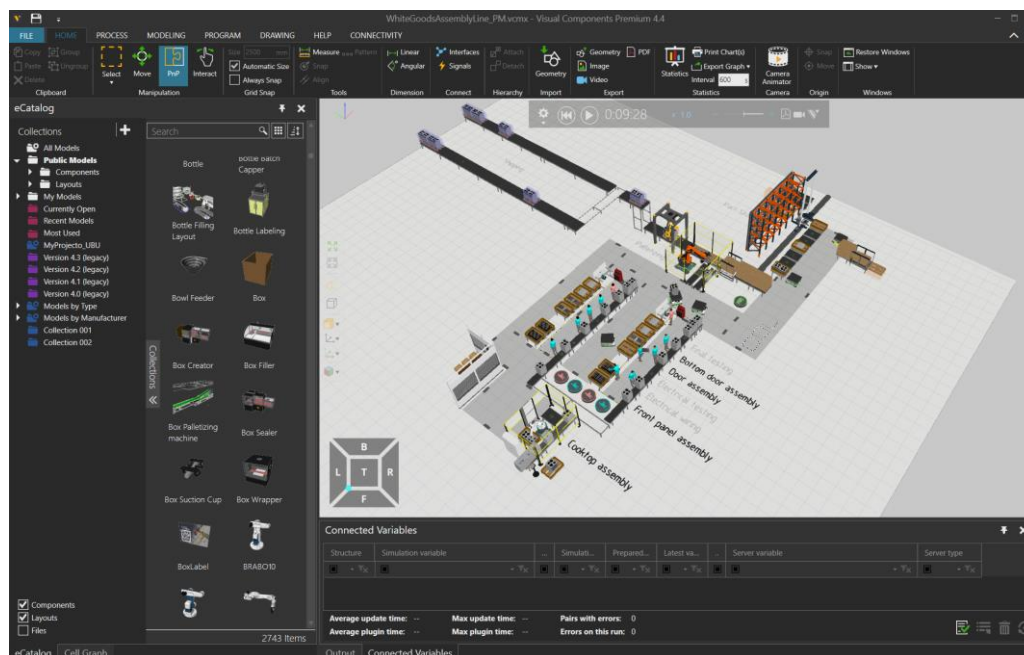


Figure 19. Screenshot of Visual Components 4.4, with an assembly layout for white goods using different levels of automation. The UI provides access to the virtual models through the eCat

Once the layout has been created the next step is to define the **production workflow** using the process modelling feature provided by VC 4.4 which follows the Product-Process-Resource (PPR) perspective. Process modelling feature is accessible within the *process* tab of VC 4.4 which it is extended with ODIN virtual components adapted to each pilot (Figure 20).

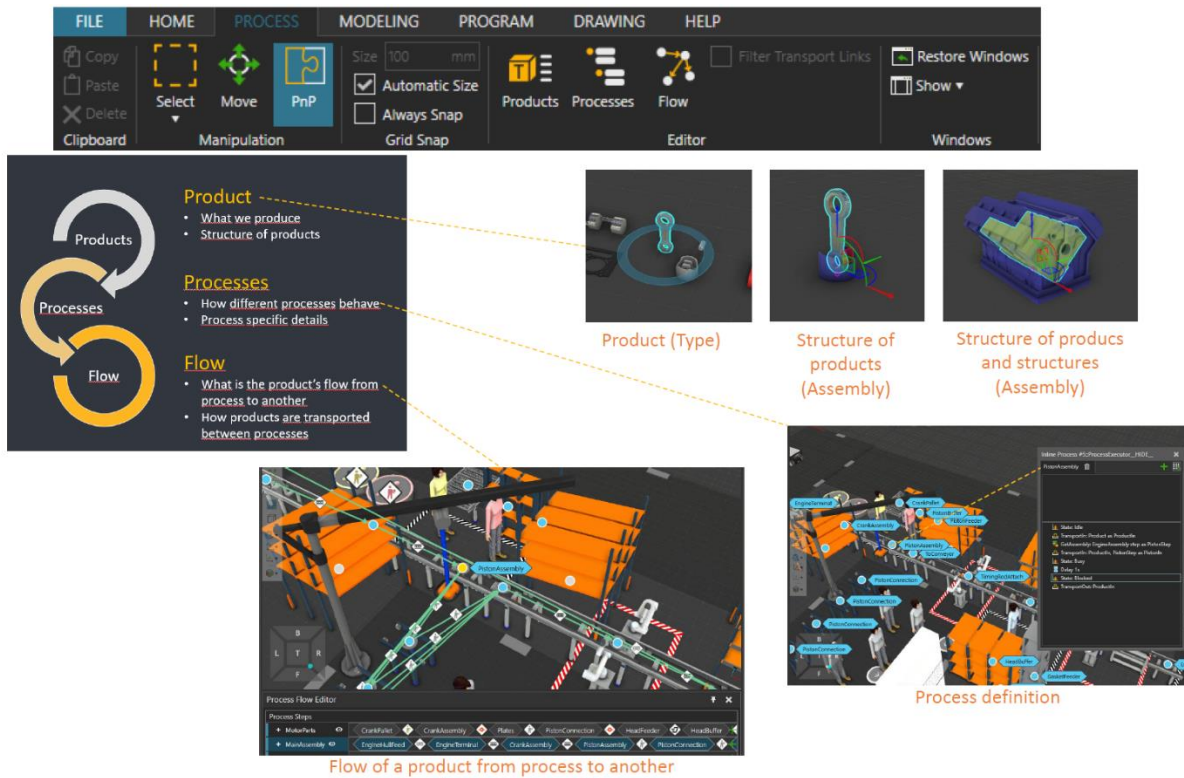


Figure 20. Detail of the process modelling workflow, creating the products, defining the processes, and describing flows between processes.

First, **the product must be created** in the digital simulation. The product is the outcome of the manufacturing process which is the result of the different manufacturing processes. It is required to create the digital product according to its structure, defining parts, assemblies, sub-assemblies and resulting products after performing any kind of processes such as machining, drilling, polish, etc. (Figure 21).

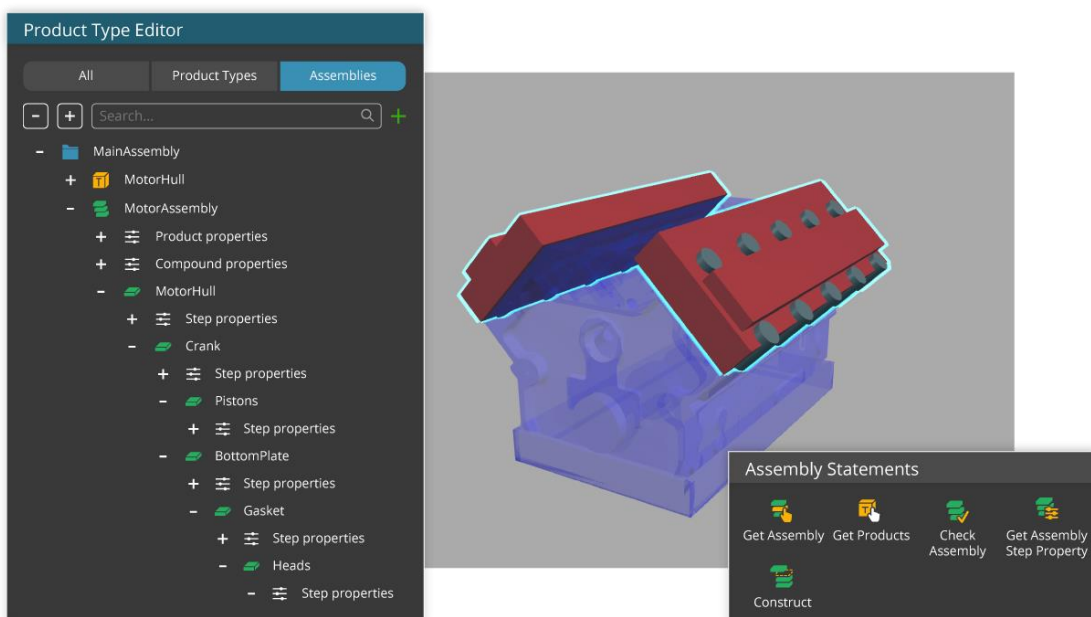


Figure 21. High level detail of the assembly of an engine motor assembly, showing the product, parts, and assemblies.

The manufacturing operations are defined within the **processes**. The default operations included within the VC4.4 are extended through the open interfaces (Python and .Net) to add the ODIN process requirements (Figure 22).

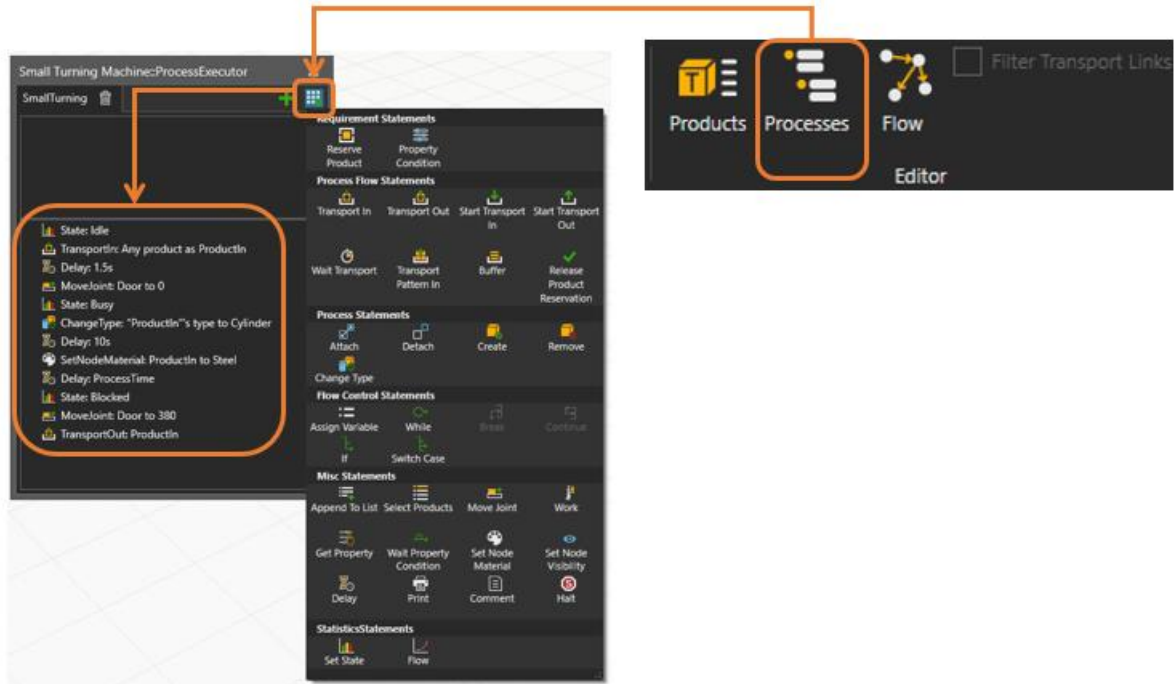


Figure 22. Creation of the manufacturing process with VC 4.4

Finally, the **production flows are added**, setting the resources (operators, machinery, robots, etc.) which will perform the required processes (Figure 23).

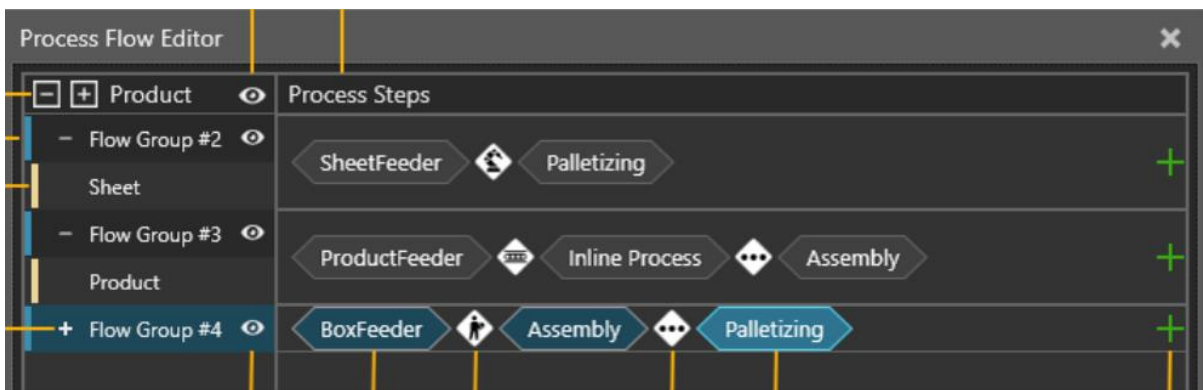


Figure 23. Screenshot showing the Process Flow editor which describe products workflow through processes and the resources used

At this point the digital simulation can be generated and the first results can be retrieved. This starts the iterative process where the simulation can be extended with more details and requirements to achieve the pilot functional and operational requirements.

It is important to mention that the digital models developed during the concept phase are used along the entire product life cycle extending them and maintaining the digital continuity. Data sets are added along the different phases ensuring digital continuity.

5.2.2. Engineering phase

The engineering phase starts once the functional and operational requirements have been achieved during the concept phase. The engineering phase extends the information contained in the virtual components by adding additional data sets and including design information. The interactions between the systems within the layout are integrated to mirror the real world within the digital simulation. Communication interfaces are defined, considering protocols and signals which will enable the digital twin functionality within the simulation environment provided by VC 4.4 (Figure 24).

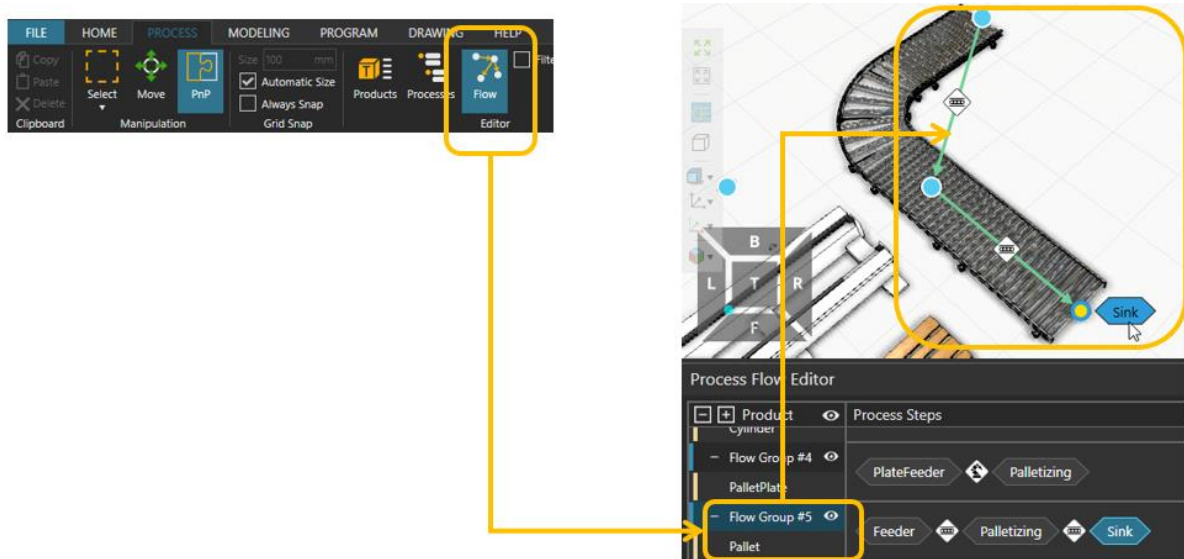


Figure 24. Screenshot of a modification during the engineering phase.

5.2.3. Virtual commissioning

Virtual commissioning completes the workflow considered within the digital simulation in ODIN. In this phase the signals are mapped between the digital environment and the real system to verify the code. Post processing plug-ins are currently being developed targeting pilots’ requirements, for example the integration with the robot manipulator used in the White Goods pilot (UR 10) (Figure 25).

Virtual commissioning requirements within ODIN and the solutions developed within the project will be covered in dedicated deliverables D3.3 and D3.4.

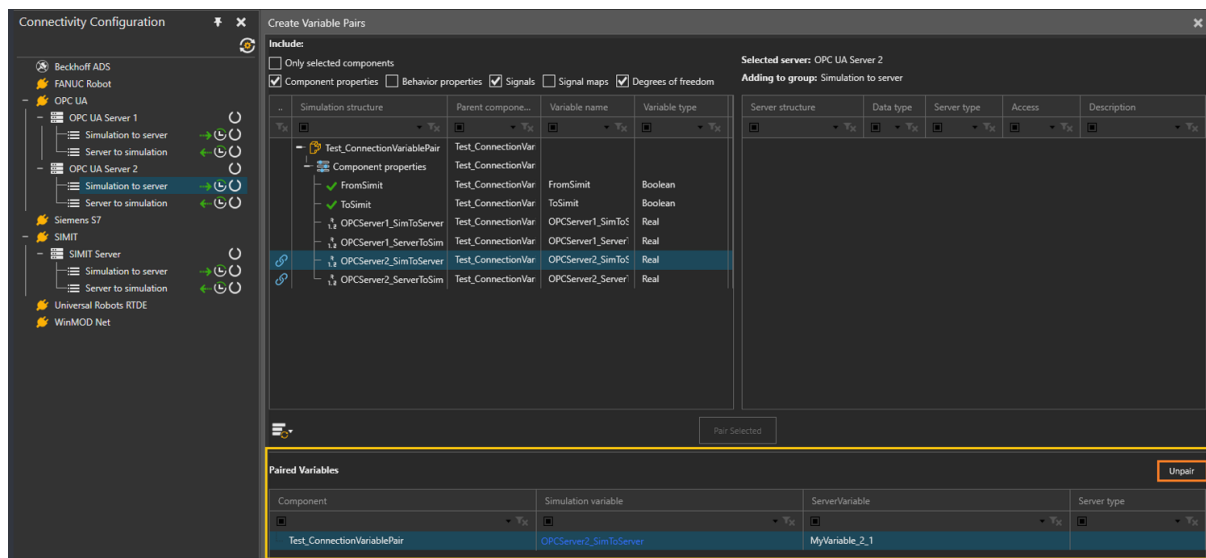


Figure 25. Screenshot of the Communication tab with signals mapping and connectivity.

5.3. Prototype implementation

Up to M18 of ODIN, partners' efforts have been centralized on the requirements for the three ODIN pilots. More detailed information has been included in the deliverable 5.1. The interaction to the open component is done through a plug-in using the open interfaces provided by VC 4.4.

The integration with the AI task planner module has started by defining the data interfaces to be used and the current status is targeting the communication between the digital simulation and the OpenFlow modules (Figure 26).

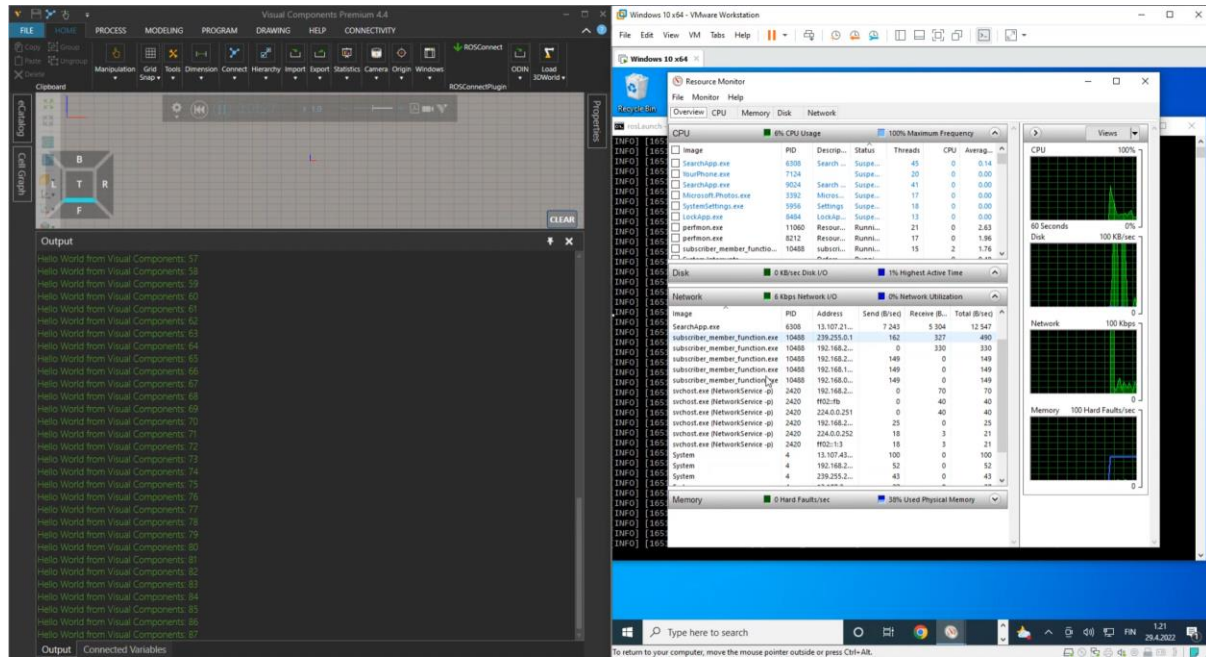


Figure 26. Screenshot of the initial prototype communication between Visual Components 4.4 and the OpenFlow

6. AI TASK PLANNER FOR DECISION MAKING

6.1. Overall methodology

The AI task planner module of ODIN focuses on the online and offline task planning process for a selected assembly operation. ODIN task planner is able to be triggered during assembly operation's execution in case of unexpected events. The task planner tool is able to automatically analyse and assess the production tasks and rearrange the available production resources/modules in real-time.

Firstly, a set of assembly tasks are inserted in the AI task planner module by the production manager in order to for scenario's modelling. Then, the scenario is added in the ODIN world model including the resource and workload hierarchical modelling of the pilot. The search engine of ODIN AI task planner module generates different alternative task plans based on tasks' model and resources' capabilities. These task plans are stored in the ODIN database for validation and evaluation through the ODIN Digital Simulation. The Digital Simulation module is connected with the ODIN AI task planner module for the simulated validation of each generated task plan. Information about the simulated execution of each task plan are stored in ODIN database. Digital Simulation layout initialization is based on data exported from the Digital Twin and the OpenFlow modules of ODIN. In order to avoid multiple execution of the same task by a specific resource when the initial conditions for the resource are the same, ODIN AI task planner module will check if task's evaluation data are already available in the database. After the simulated execution of each task plan, the AI based task planner module is responsible to retrieve data about each task plans' execution from the database and evaluate them. Based on the evaluation criterias' weight defined by the production manager, the best alternative task plans will be visualized through the UI to the production manager. He/She is able to evaluate them based on his/her knowledge and select the best alternative to be sent for physical execution through the OpenFlow. This exchange of information is also visualized in Figure 27.

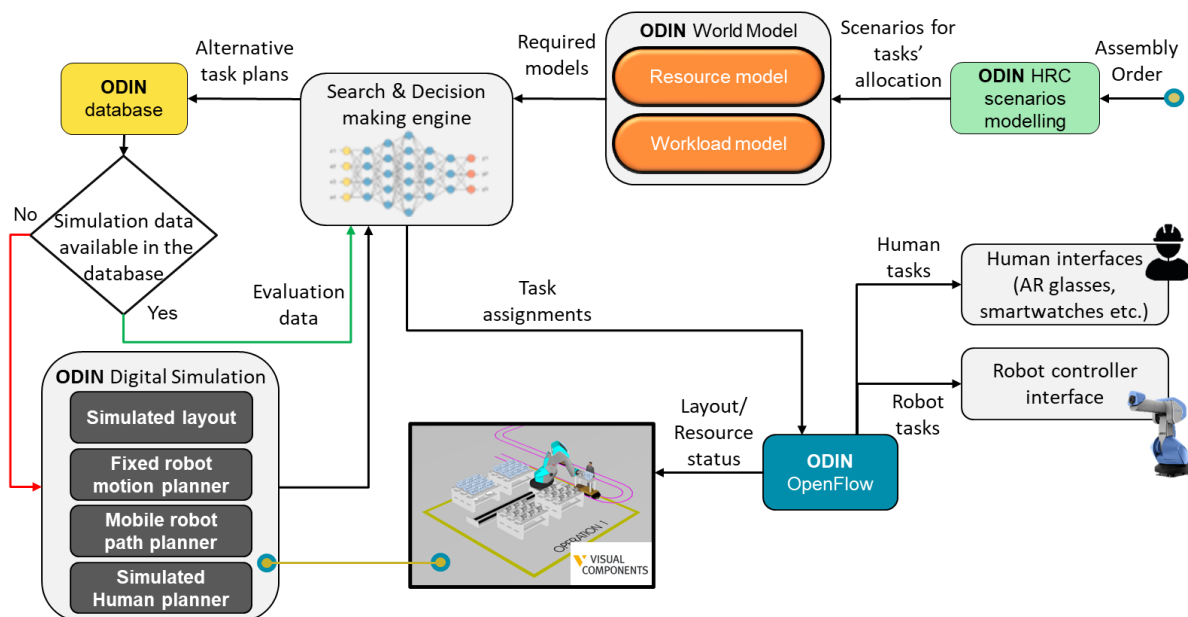


Figure 27. ODIN AI task planner module methodology

6.1.1. Task level work re-organization

6.1.1.1. Overall concept

ODIN task planning algorithm consists of 4 steps:

- **Receive request for Task Planning:** The task planner algorithm receives a request for execution either by the production manager or dynamically by the OpenFlow in case of unexpected events in the shopfloor forcing the production to stop.

- Initiate and trigger Task Planner through the developed UI:** The production manager initializes the search parameters, the evaluation parameters' weights, the filtering criteria and the number of plans to be calculated for the task planning algorithm. Then the task planner algorithm is triggered to start calculating task plans and sends the generated task plans for validation in the Digital Simulation.
- Task Plan validation and evaluation:** In this step, the AI task planner module's connection with the Digital Simulation but also the Digital Twin is presented. The generated task plans from the Task Planner algorithm are sent for simulated execution in the Digital Simulation of ODIN. Digital environment's layout initialization is based on data received from the OpenFlow (robot's joint values, fixed resources' location etc.) and the Digital Twin (operator's location). OpenFlow is responsible to receive the tasks' allocation from the AI task planner module and translate them to primitive actions for simulated execution in the 3D simulation. The Digital Simulation executes the primitive actions and stores information about these actions' execution in ODIN database to be used by the OpenFlow during task plans' evaluation.
- Generated Task Plan evaluation and best alternative selection for execution:** The generated task plans are automatically evaluated based on the pre-defined evaluation metrics and their selected weights. Then the top 3 task plans are visualized through this modules' UI. The production manager is able to check the AI task planner module's result through the UI and save the optimal task plan for execution in the physical environment.

Finally, the production manager is able to send the selected task plan for execution in the physical shopfloor through the developed UI of Task Planner module.

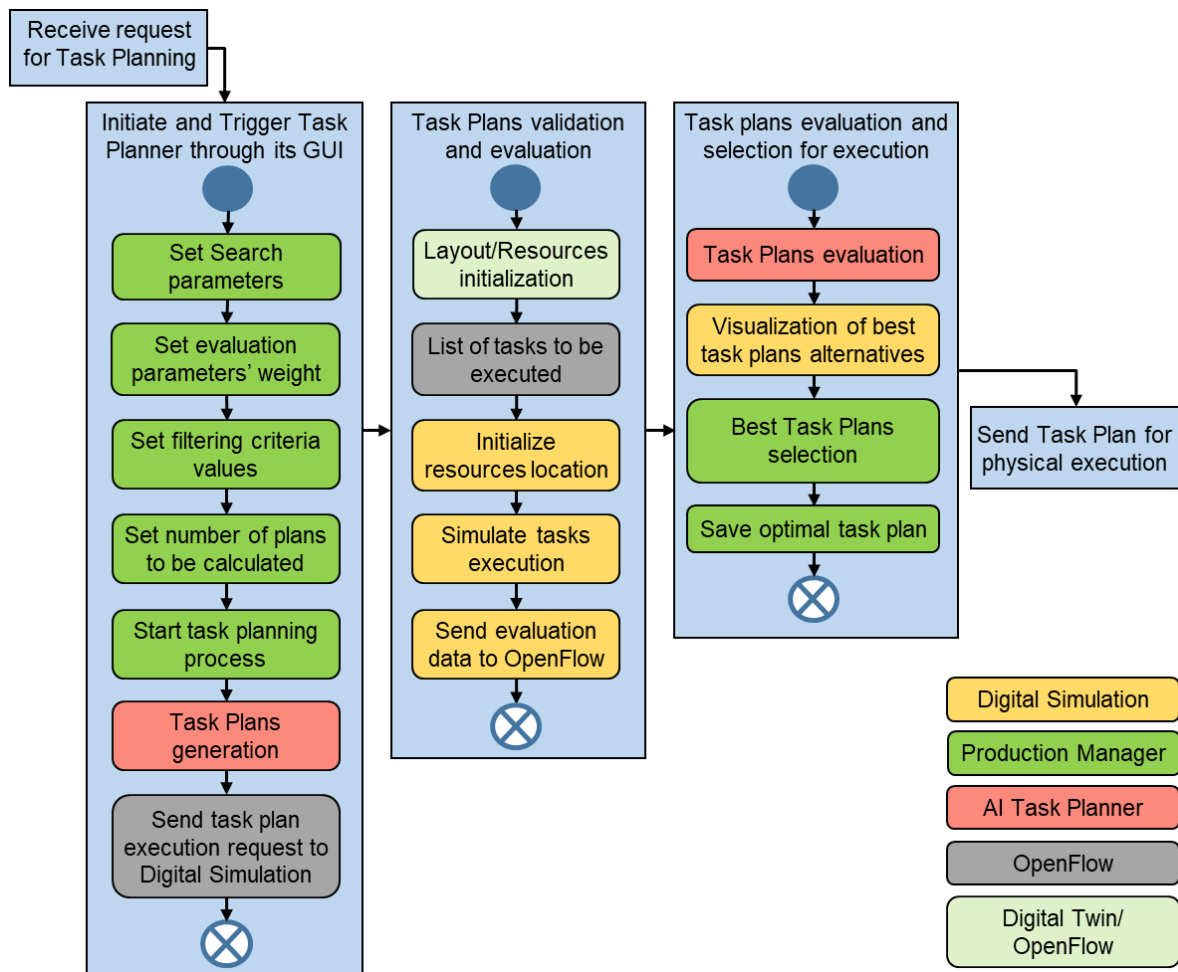


Figure 28. AI task planner module process diagram

6.1.1.2. Task planer evaluation criteria

A set of evaluation criteria have been defined by LMS and ODIN industrial partners for the evaluation of different task plan generated by the AI task planner module. These criteria are presented in the flowing table (Table 5) and will be used for alternatives' task plans evaluation.

Table 5. Criteria for Task Plans evaluation

Title	Description	Measurement unit
Flowtime	Total time duration of a generated task plan execution	Seconds
Human busy-time	Total time duration that a human execute tasks during a generated task plan execution	Seconds
Distance Covered	Distance covered by each resource during a generated task plan execution	Millimetres
Ergonomics	How much stressful are the human tasks for the operator based on the generated task plan	Integer number calculated using an Ergonomic analysis method (OCRA, RULA etc.)
Non-Adding values activities time	Time duration that no tasks are executed by any resource	Seconds
Utilization of Resources	Utilization of Resources during a generated task plan execution	(Time contributing to the execution of assembly) / (Overall cycle time) %
Safety – HR distance	Avoid any plans that do not conform to the minimum distance constraints set between humans and robots	Boolean

6.1.2. Modelling of Human Robot Collaborative scenarios

A hierarchical modelling approach has been implemented in ODIN for assembly processes' representation. This modelling consists of 4 levels of information for the description of a process plan. The first two levels are the top levels of the assembly process, the third one is the level of information used by the AI task planner module while the fourth one is the level of information sent to the different resources for motion, navigation, tools' enabling/disabling actions etc. In more detail, the four semantically different levels of activities are defined as follows:

- **Order:** The root element of the entire process plan which should be executed for the assembly of a product (e.g. oven assembly, motor assembly).
- **Job:** It is the second level of activities need to be performed for the assembly of the individual parts of the final product (e.g. transformer installation in the oven, gearbox installation on the motor).

- **Task:** This information is related to the work that a resource can handle and is being used by the AI task planner module for task plans generation (e.g. remove transformer from kitting area, pick gearbox from its kitting pallet).
- **Primitive action:** It is the lowest level of activities that the actual hardware will interpret to control the actual Mobile Robot Platforms (both the platforms and the manipulators on top of them), static robots, operators and any other type of resource (e.g. enable/disable gripper, set robot payload).

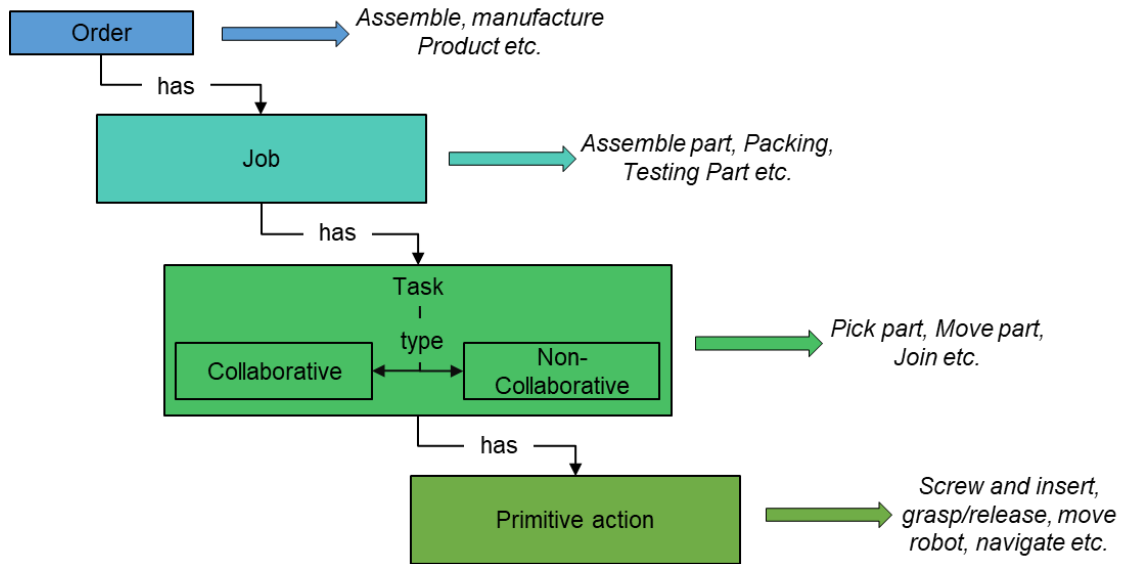


Figure 29. ODIN Workload hierarchical model

Following the same methodology with the workload model, a hierarchical modelling approach has been implemented for ODIN available resources. This model is critical for the AI task planner module as it includes information about the existing resources and their capabilities. Additionally, information about each resource’s suitability for each assembly operation are included in this model. An example of the ODIN resource hierarchical model for dynamic and static resource is visualized in Figure 30.

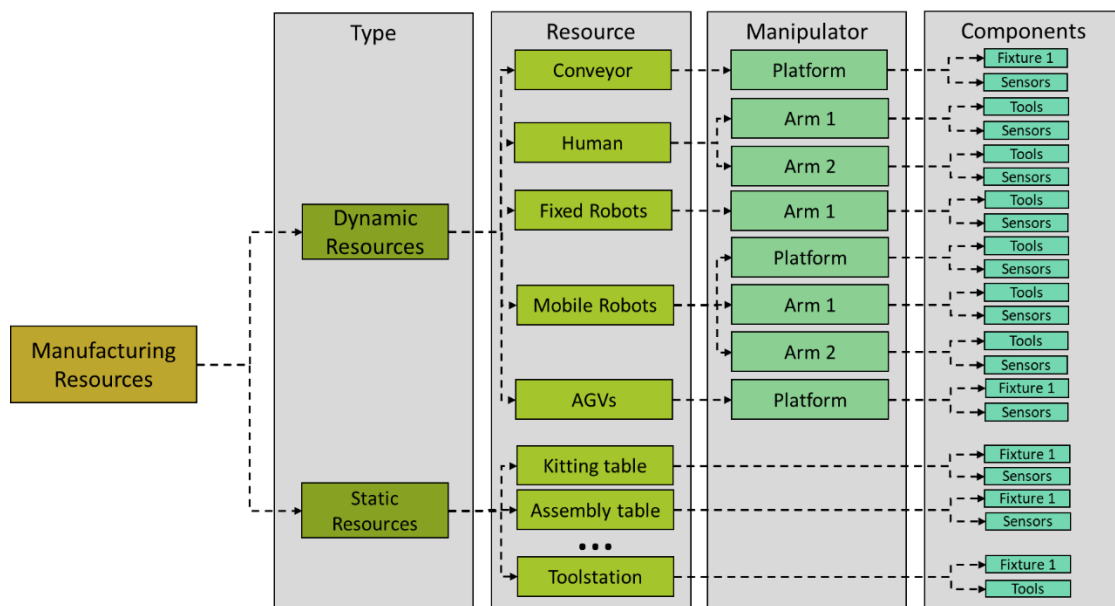


Figure 30. ODIN Resources hierarchical model

ODIN AI task planner module will define task plans where tasks will be assigned to manipulators and through the OpenFlow assembly primitive actions will be assigned to manipulators' components for assembly tasks' execution.

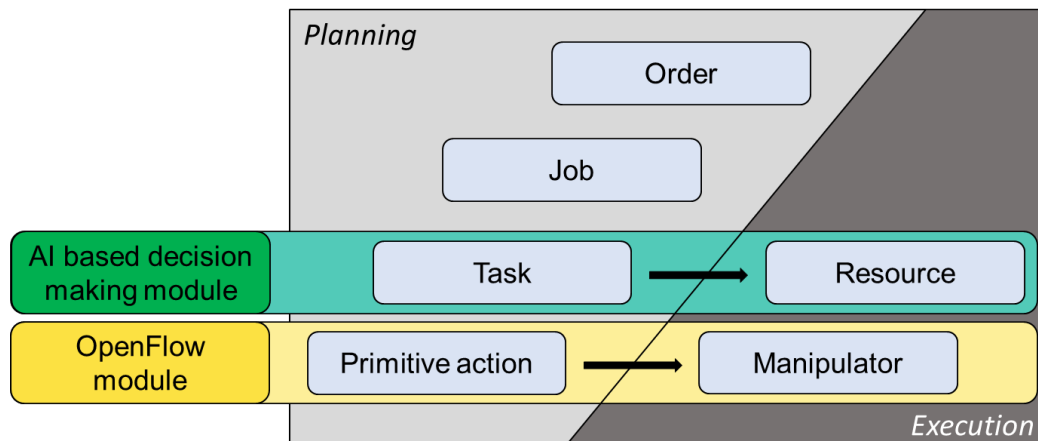


Figure 31. Assignment of Tasks and Primitive actions

6.1.3. ODIN Resource models characteristics

Task plans' creation for HRC processes including human and mobile but also fixed robotic resources requires some specific rules and constraints to be defined in order to differentiate the resources and allow the task planning algorithm to create the final allocation based on as many of the system variables as possible. ODIN resources will be modelled with certain characteristics as listed below:

- **ID:** The identification number of resource which will be automatically defined by the AI task planning algorithm and will be used to define each resource.
- **Title:** The user can define a specific title for each resource.
- **Type:** Resource's type will be defined by the user. The available options will be:
 - Operator.
 - Mobile Robot.
 - Static Robot.
 - Static Resource.
- **Strength:** This value corresponds to the maximum value of weight each resource is able to handle. Human operators' strength will be initially set to 1 kg but the user will be able to change this value based on end user's requirements. For ODIN robots, the weight of the part should be less than its payload based on robots' specifications.
- **Speed:** This is the speed of robot end effector. It will be used to make an initial calculations for the time that the robot needs for the execution of an assembly task in case the digital simulation will not be used for task plans' evaluation.
- **Flexibility:** This number quantify the level of flexibility of the parts' material. Human operators are set having the ability to manipulate flexible parts and the robot's manipulability is by default set to 0 unless the user define that they are able handle this type of parts.
- **Envelope:** This number corresponds to the floor space required by a resource in order to execute the assigned task. It is more critical in case of the mobile resources to consider suitability depending on the area where the task needs to be executed in.
- **Reachability:** This value depicts the maximum reach of the robot arm in meters and will be used to evaluate if the position of a robot is correct to execute the assigned task.

6.1.4. Task planning search problem

In ODIN, each task allocation problem is being treated as a search problem. The task allocation search tree is being created based on resources' capabilities to perform each assembly task but also the pre and

post conditions for each task (Figure 32). The blue boxes of the search tree represents the assembly tasks which are required to be executed for the assembly of the investigated product, while yellow boxes represent the assigned resource. Each branch of the tree represents a complete task plan alternative.

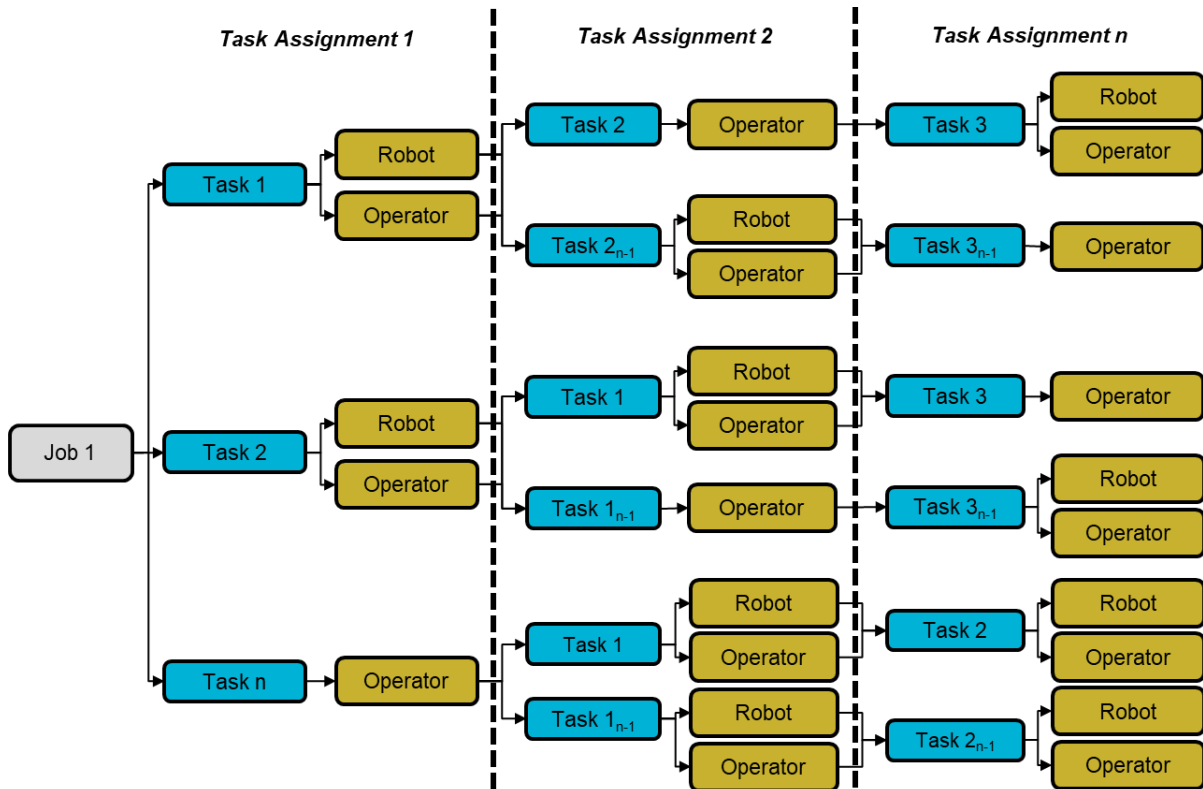


Figure 32. ODIN task planning search problem formulation

6.1.5. Planning algorithm description

ODIN search space algorithm consists of 3 configuration parameters (Integers) as presented below:

- DH - Decision Horizon: It is the number of that will be examined in each searching step.
- SR - Sampling Rate: This parameter depicts the number of samples that will be defined for each of the decision points of the decision horizon.
- MNA - Maximum Number of Alternatives: This parameter defines the maximum number of alternatives that will be formed at each step of the algorithm.

Figure 14 presents an example of the ODIN search space algorithm where the search parameters (DH, SR, MNA) are initialized with the values (2, 2, 2).

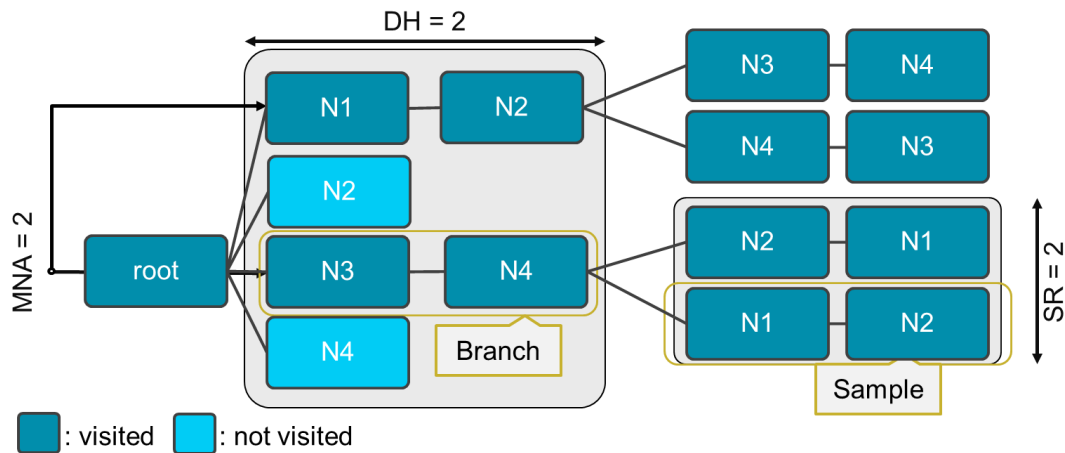


Figure 33. Intelligent algorithm search space

The following steps are used to search the tree for a good solution:

- 1) Alternatives' creation by randomly creating assignments for all layers included in the DH. This creation takes place until MNA parameter is equal to DH parameter.
- 2) For each alternative of Step 1, create SR sample alternatives by randomly assigning resources to operations until all nodes of the specific branch are searched.
- 3) Calculate the criteria scores for all samples.
- 4) Calculate the alternative scores as the average of scores achieved by its samples.
- 5) Calculate the utility values of all alternatives.
- 6) Select the optimal alternative.
- 7) Save the assignments of the investigated alternatives in ODIN data repository.
- 8) Repeat steps 1 to 7 until an assignment has been performed for all the nodes/operations of the selected branch.

Based on the analysis of how the search tree is generated, the following key points can be highlighted:

- MNA controls the breadth of search process.
- DH controls the depth of search process.
- SR directs the search towards branches which will provide more efficient solutions.

A defining method for selecting the [MNA, DH, SR] values is presented in [13]. For all parameters, good quality alternative might be easier generated (i.e. an task plan with a utility value within a range Δ with respect to the alternative with the highest utility value) with increased parameters' value and Δ following an almost negative exponential distribution and finally levelling off at the alternatives with the highest utility values. A careful selection of the MNA, DH and SR will result to a good quality solution by investigating a very limited set of the search space and effectively reducing the computational time. This method has been selected based on its customizability of the decision parameters enabling the calculation of speed and the good quality of solutions for different sizes of the problem.

6.2. Prototype implementation

6.2.1. AI Task Planner User Interface (UI)

Production managers are able to interact with the AI Task Planner module of ODIN through the developed User Interface (UI). Through this UI, the production manager is able to initialize the search and evaluation parameters of task planning algorithm but also check the generated task plans and evaluation data. In more details production managers are able to:

- Trigger the Task Planning algorithm.
- Check the resource and workload hierarchical model for the selected pilot.
- Initialize the task planning algorithm's DH, MNA and SR parameters.
- Initialize the evaluation criterias' weight to be used during the evaluation of the generated task plans.
- Check the results of generated task plans' evaluation process and select the optimal solution for sending to physical execution.

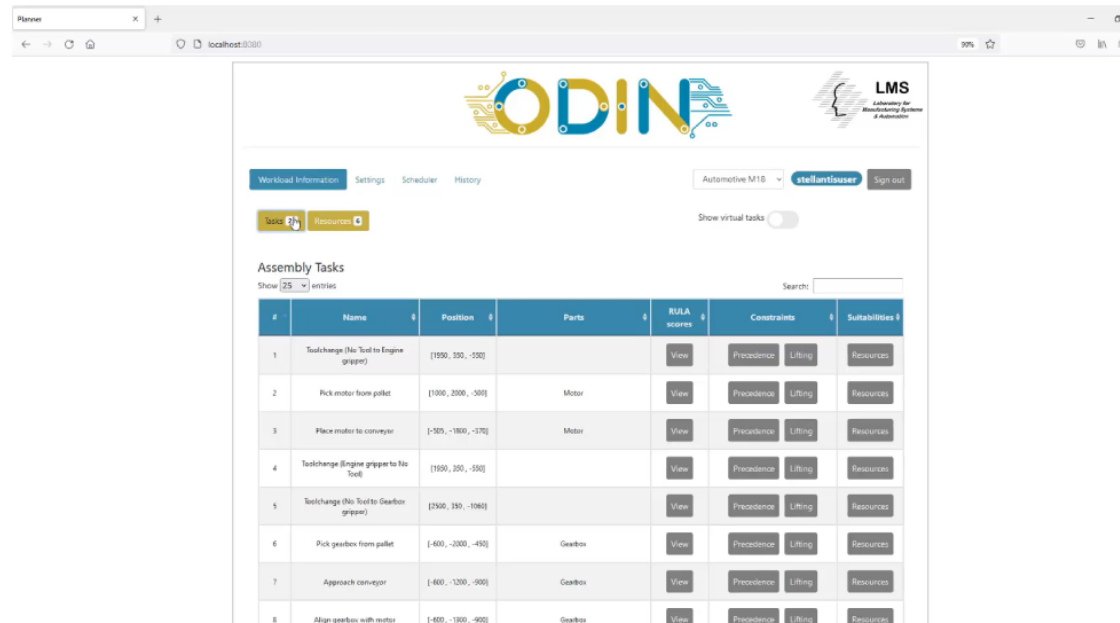


Figure 34. ODIN AI Task Planner UI

6.2.2. AI Task Planner integration with OpenFlow

The AI Task Planner (included in DC) and OpenFlow (included in NC) modules integration is presented in Figure 35. In more details:

➤ AI Task Planner to OpenFlow

On the one hand, The AI Task Planner module's connection with the OpenFlow is focused on the generated task plans data transfer from the Task Planner to the OpenFlow. Then the OpenFlow is responsible to analyse the received information and assign primitive actions to the available resources for simulated or physical execution. Until M18 of ODIN, this connection have been completed and during the next period, contributing in this task partners' efforts will be focused on the dynamic re-scheduling capabilities of the AI Task Planner.

➤ OpenFlow to AI Task Planner

On the other hand, OpenFlow module's connection with the AI Task Planner is focused on simulated results and evaluation data exchange. This information is used by the AI Task Planner for task plans evaluation.

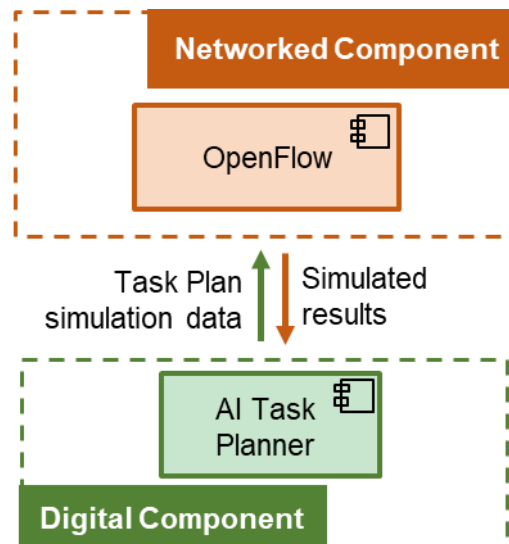


Figure 35. AI Task Planner and OpenFlow modules' integration

6.2.3. AI Task Planner integration with Digital Simulation

The AI Task Planner and Digital Simulation modules of the DC will be indirectly connected as the OpenFlow module will be used as the communication and orchestrator channel. The Digital Simulation module will be used for the simulated execution, validation and evaluation metrics calculation for each task plan generated by the Task Planner and broken down to primitive actions by the OpenFlow. Until M18 of the project, the main efforts for these modules' connection have been centralized in simulated layout's initialization. The first version of ODIN White Goods small scale pilot setup has been chosen to test this modules' connection. As visualized in Figure 36, the scene of the selected small-scale pilot is able to be automatically initialized in the digital environment using a custom button in the VIS simulation software. Information about resources' initial position and status (Robot and Human pose) are provided by the OpenFlow module to the Digital Simulation.

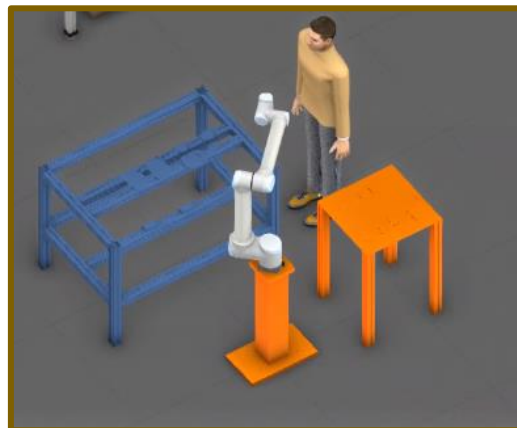


Figure 36. Digital simulation environment initialization

7. CONCLUSIONS

This document presents the initial prototypes of ODIN DC modules and their level of implementation with other modules of the project. In more detail the following components' implementation has been discussed in this deliverable:

- Digital description of ODIN Open Components including information about ODIN resources such as size, interfaces, geometry/kinematics, business/design properties, and the specific capabilities.
- Digital Twin of ODIN HRC system integrated with a) pre-processing, b) data fusion and c) decision technologies including real time sensor data for safe HRC.
- Digital Simulation for pilot lines simulation execution and validation.
- AI based task planner module for the generation of different task plans and assignments to available resources generation but also their validation using data from the Digital Simulation module.

A communication interface inside the simulation platform for the virtual control and commissioning of ODIN pilots will be presented in D3.2. This communication interface will allow the communication of the DC with the OC to enable virtual control of ODIN pilots in the simulation environment.

Following the initial time plan of the project, the next version of ODIN DC modules and their final integration in the project will be delivered under D3.3. Internal milestones have been defined from the partners focusing on these modules' validation and refinement against end user requirements and feedback.

8. GLOSSARY

API	Application Program Interface
ARD	Abstract Resource Description
BE	Back-end
DB	Database
DC	Digital Component
DT	Digital Twin
EC	European Commission
EU	European Union
FB	Function Block
FE	Front-end
GUI	Graphical User interface
HRC	Human Robot Collaboration
HRI	Human Robot Interaction
IC	Industrial Component
ICT	Information and communications technology
JSP	Java Server Pages
NC	Network Component
OC	Open Component
RCP	Resource Catalogue Platform
RD	Resource Description
UI	User Interface
VC	Visual Components
WP	Work Package
XML	Extensible Markup Language

9. REFERENCES

1. Chryssolouris, G., *Manufacturing Systems: Theory and Practice*, 2nd Edition, Springer-Verlag, New York, New York, (2006)
2. G. Michalos, S. Makris, N. Papakostas, D. Mourtzis, G. Chryssolouris, "Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach", *CIRP Journal of Manufacturing Science and Technology*, Volume 2, Issue 2, pp. 81-91 (2010)
3. Anonymous, "Evolvable Ultra-Precision Assembly Systems (EUPASS)," Portal, Cordis EU, 2009. Grant agreement ID: 507978. [Online]. Available: <https://cordis.europa.eu/project/id/507978>
4. N. Siltala, "Formal Digital Description of Production Equipment Modules for supporting System Design and Deployment," Tampere University of Technology, 2016. Available: <http://urn.fi/URN:ISBN:978-952-15-3783-7>
5. Anonymous, "Rapid Reconfiguration of Flexible Production Systems through Capability-based Adaptation, Auto-configuration and Integrated tools for Production Planning (ReCaM)," Portal, Cordis EU, 2018. Grant agreement ID: 680759. [Online]. Available: <https://cordis.europa.eu/project/id/680759>
6. N. Siltala, "Resource Description, XML Schema,," 2018. [Online]. Available: https://resourcedescription.rd.tuni.fi/resourcedesc/schemas/ResourceDesc_v2-0-1.xsd [Accessed: 30-May-2022].
7. E. Järvenpää, "Capability Model, Ontology," 2018. [Online]. Available: <https://resourcedescription.rd.tuni.fi/ontology/capabilityModel.owl> [Accessed: 30-May-2022].
8. N. Siltala, "Software Requirements Specification for Resource Catalogue Platform," Tampere, p. 27, 2022.
9. R. Ma, M. Lanz, and N. Siltala, "Visualization concept for representing capability matchmaking results in a virtual environment," in 31st International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), 2022, p. 9. (*in press*)
10. D. Mourtzis, M. Doukas, and D. Bernidaki, "Simulation in Manufacturing: Review and Challenges," *Procedia {CIRP}*, vol. 25, no. 0, pp. 213–229, 2014, doi: <http://dx.doi.org/10.1016/j.procir.2014.10.032>.
11. <https://en.wikipedia.org/wiki/PDCA>
12. H. Sechsrhön, G. Fritz, R. Holom, V. Stanzl and F. Ubis, *Big Data Machine Tool Cybernetics, BOOST 4.0 (Big Data Value Spaces for Competitiveness of European Connected Smart Factories 4.0)*, Final Conference, On-Line event, 10th December 2020.
13. Chryssolouris, G., Dicke, K., and Lee, M., 1992. On the resources allocation problem. *International Journal of Production Research*, 30 (12), 2773–2795.
14. IEC Standard, Available. [Online]:<https://iec61499.com/>