



DIMOFAC

Digital & Intelligent MOdular FACtories

DT-FOF-08-2019 Pilot lines for modular factories

Grant Agreement n° 870092

D3.1: Module hardware and software interfaces

Deliverable nature	OTHER
Dissemination level	PUBLIC
Contractual delivery date	31/05/2022
Actual delivery	31/05/2022
Version	V2
Lead Author	ALONSO Lucía (AIM)
Contributors	ALONSO Lucía (AIM), MOHAMMED Asif (AIM), SEDANO Iñigo (TCN), LOPEZ Pedro (TCN), MISURACA Antonio (COM), MONTAGUD Santiago (ESI), BELLE Sabine (IPC), PAPAIOANNOU Christos (LMS), COSTIN Marius (CEA), WENDLING Michael (SSF), MEDEN Boris (CEA)
Reviewer	QUADRINI Walter (POL), Cécile GIRARDOT (CEA)



1 Version history

Version	Date	Editors	Description
0	08/10/2021	Asif Mohammed	Outline for the deliverable
	05/11/2021	Iñigo Sedano, Pedro López [TCN]	Added content to 5.2. Laser joining
	14/02/2022	Antonio Misuraca [COM]	Added content to 7.1 Comau mobile platform and 7.2 OnRobot gripper
1	28/02/2022	Lucía Alonso	Changes in ToC
1	15/03/2022	Santiago Montagud	Software for M1 and M4
1	30/03/2022	Sabine BELLE	Section Interface & AAS for M1 and M2
1	11/04/2022	Christos PAPAIOANNOU [LMS]	Milling Module Contribution
1	12/04/2022	Christos PAPAIOANNOU [LMS]	Draft Laser Welding Contribution
1	15/04/2022	Marius Costin [CEA]	Added content related to modules 6 and 18
1	20/04/2022	Lucía Alonso [AIM]	Added content related to modules 5 and 8.
1	26/04/2022	Michael Wendling (SSF)	Flexible Module and Robotic Module description
1	28/04/2022	Iñigo Sedano, Pedro López [TCN]	Added content to 5.2. Laser joining
1	02/05/2022	Lucía Alonso (AIM)	Introduction, conclusions and some missing titles
1	03/05/2022	Walter Quadrini (POLIMI)	Internal revision
2	31/05/2022	Cécile GIRARDOT (CEA)	Final review and submission

Important remarks

The contributors quoted in this table and on the front page are the main editing authors of this report. It is important to note that all pilot line members and technology providers supplied essential technical contributions to the software architecture.

A glossary of technical terms is given in section 6.1 Glossary of Technical Terms (Annex) of this report



DIMOFAC



2 Abstract

This deliverable focuses on the development of hardware and software interfaces for the integration of the different physical assets which compromise a production module, connecting heterogeneous devices and field buses with the middleware based on Asset Administration Shell (AAS).

It describes the architecture of the different modules, as well as the interfaces and the software developed for achieving plug and produce capabilities. This deliverable also shows the data modelling for each asset and the relevant information shared by the module.

One of the main objectives of this task is the integration of a master controller (i.e., IoT gateway) at pre-industrial lab scale. This was affected by the chip crisis and the shortage of material. Indeed, the development of the AAS was affected by the delays of the definition of specifications of a Common Information Model from WP1. For minimising the delay, meanwhile the equipment did not arrive. The focus of this task was to develop the models. On this purpose, several meetings for discussing the AAS structure and modelling were scheduled.

This public version of the deliverable is published in the DIMOFAC webpage (<https://dimofac.eu/project-deliverables/>)



3 Table of contents

1	Version history.....	2
2	Abstract	3
3	Table of contents.....	4
3.1	Table of figures	5
3.2	Table of Table	7
4	Modules.....	8
4.1	MODULE 1 – Injection moulding	9
4.2	MODULE 2 – Modular quality control for injection moulding	22
4.3	MODULE 3 – Parts preparation module: laser texturing	26
4.4	MODULE 4 – Processing module: induction joining	29
4.5	MODULE 5 – Processing module: laser joining	36
4.6	MODULE 6 – Quality Control for fibre reinforced panels.....	40
4.7	MODULE 7 – Small parts assembly.....	43
4.8	MODULE 8 – Automated Tape Laying.....	47
4.9	MODULE 9 – AGV-Robot	52
4.10	MODULE 10 – Milling	54
4.11	MODULE 11 – Laser quality control / sheet-metal bending and additive manufacturing.....	61
4.12	MODULE 12 – AR assembly instructions	66
4.13	MODULE 13 – Laser welding.....	70
4.14	MODULE 14 – Flexible module (assembly, cabling quality control)	72
4.15	MODULE 15 – Robotic handling	75
4.16	MODULE 16 – Parts storage	77
4.17	MODULE 17 – Bin picking handling	78
4.18	MODULE 18 – Quality control – X-ray for additive manufacturing	82
5	Conclusions	88
6	Appendices.....	89
6.1	Glossary of Technical Terms.....	89
6.2	Result's disclosure plan.....	90



3.1 Table of figures

Figure 1 : Architecture of Injection Moulding Module	10
Figure 2 : AASX of Injection Moulding Machine AAS	11
Figure 3 : AASX of Thermal Controller Device AAS	11
Figure 4 : AASX of Mould AAS.....	12
Figure 5 : Material AAS for IM module.....	13
Figure 6 : AAS of the Plastic Part	14
Figure 7 : AAS of a Moulded Part.....	14
Figure 8 : AASX model of Job_AAS	15
Figure 9 : the 3 AASs for DT of Injection Moulding Module	16
Figure 10 : ESI ROMs AASs for both mould temperature and pressure.....	16
Figure 11 : AAS for DT based on Simcenter Amesim tool	17
Figure 12 : IM Module configuration from new PartCode	19
Figure 13 : starting up of the process from Job_AAS	19
Figure 14 : Real Time data from Module 1 and Module 2 on Grafana dashboard	20
Figure 15 : data recording at each cycle	21
Figure 16: Connection of ESI software to DIMOFAC Platform by Custom connectors	21
Figure 17 : architecture of Modular Quality Control Module	22
Figure 18 : PlasticPart_AAS related to Module 2.....	23
Figure 19 : MouldedPart_AAS including Quality Control Results and Traceability.....	24
Figure 20 : Quality Control and Traceability in Job_AAS	24
Figure 21 : Module 2 configuration from PartCode	25
Figure 22: Architecture of the module 3 (texturing).....	26
Figure 23: Linear axis AAS model	27
Figure 24: Scanner and laser AAS model	27
Figure 25: Texturing module AAS	28
Figure 26: Product AAS.....	28
Figure 27: Induction joining architecture	29
Figure 28: Grafana panel for the Induction Joining module	30
Figure 29: Robot AAS	31
Figure 30: Generator AAS	31



Figure 31: Pyrometer AAS.....	32
Figure 32: Induction Welding application AAS (ESI).....	35
Figure 33: Architecture of the laser joining module	36
Figure 34: Laser joining AAS submodel.....	39
Figure 35: Architecture of the UT QC module (module 6)	41
Figure 36: AAS model for UT QC module (module 6)	42
Figure 37: Main GUI of UT QC module (module 6)	43
Figure 38: Architecture of Small parts assembly specifics Module	44
Figure 39: Robots' flange (ISO 9409 - 1 - A 25)	47
Figure 40: ATL module: architecture	48
Figure 41: DT in Blender and Grafana dashboard	49
Figure 42: Robot AAS	50
Figure 43: ATL Head AAS	50
Figure 44: Heat system (Laser and IR) AAS	51
Figure 45 : Comau mobile platform	53
Figure 46. ROS communication scheme	54
Figure 47: Milling Module (ID: 10) Architecture	55
Figure 48: Reference Architecture of QC module 11 and its main interaction with other modules and components.....	61
Figure 49: Reference flow diagram of Laser QC module.	63
Figure 50: Scanner AAS.	64
Figure 51: Measurement AAS.	64
Figure 52: Colour deviation map SCL sample part.	66
Figure 53: AR application functionalities	66
Figure 54: Overall system architecture.....	67
Figure 55: AR application AAS	68
Figure 56: Unity3D graphical interface	69
Figure 57: Laser Welding Module Architecture.....	70
Figure 58: Laser Welding Simulation Tool Screenshot.....	72
Figure 59: Electro-mechanical interfaces between the flexible module and robotic module	73
Figure 60: Sensors of the modular module	73



Figure 61: Pilot line 5 flexible module infrastructure	74
Figure 62: Pilot line 5 flexible module AAS ModelPnP capabilities.....	74
Figure 63: Fanuc CRX10iA with Schunk EGP-C-64 cobotic gripper	75
Figure 64: Pilot line 5 robotic module infrastructure	76
Figure 65: Pilot line 5 robotic module AAS Model	76
Figure 66: Parts storage infrastructure	77
Figure 67: Parts storage AAS Model	78
Figure 68: CAD model (top left), 3D scene provided by a 3D sensor, with different instances of the object (top right) and multiple registration of the CAD in blue over the scene (middle below)	79
Figure 69: Bin picking module architecture	79
Figure 70: Global Architecture of the Bin Picking (module ID 17)	80
Figure 71: Bin Picking AAS model.....	81
Figure 72 - CAD design of the inspection cabin of ID18 module.....	82
Figure 73 – Global architecture of the ID18 module	83
Figure 74 - AAS model of ID18 module	85
Figure 75: GUI of software for the ID18 module	87

3.2 Table of Table

Table 1: Portfolio of Modules, Leader and PL	8
Table 2: EXOM laser head parameters stored into the AAS server.....	38
Table 3: Specifications of accelerometer Kistler.	56
Table 4: Specifications of accelerometer Micromega.	56
Table 5: Specifications of Labjack T7-Pro	57
Table 6: Specifications of Dell Latitude 5501 laptop.....	58
Table 7: Specifications of AD-Link DLAP-401-Xavier	58
Table 8: IIoT Gateway Specifications	59



4 Modules

The developments done in this task are specific for each module defined in DIMOFAC portfolio (D3.10). Even these developments are specific for each module, the development procedure is common and is the following one:

- Definition of the architecture and protocols between assets and between assets and DIMOFAC Platform.
- Development of interfaces using the specific protocols for each asset.
- Modelling in AAS using Package Explorer tool.
- Software development and dataflow definition for managing AAS information and for achieving Plug-and-Produce (PnP) capabilities.

Module		Leader	PL
1	Injection moulding	IPC	ALB / PCL
2	Modular quality control	IPC	ALB / PCL
3	Parts preparation module : laser texturing	AIM	EIRE
4	Processing module : induction joining	EIRE	EIRE
5	Processing module: laser joining	TCN	EIRE
6	Quality evaluation & control (NDT)	CEA	EIRE
7	Small parts assembly specifics	LMS	PCL
8	Automated Tape Laying	AIM	OPL AIM
9	AGV-Robot	LMS	VDL
10	Milling	LMS	VDL
11	Laser quality control / Sheet-metal bending& additive manufacturing	DPX	VDL / SCL
12	AR assembly instructions	LMS	VDL
13	Laser welding	LMS	VDL
14	Flexible module (assembly, cabling, quality control)	SSF	SLT
15	Robotic handling	SSF	SLT
16	Parts storage	SSF	SLT
17	Bin picking handling	CEA	SCL
18	Quality control - X-ray	CEA	SCL

Table 1: Portfolio of Modules, Leader and PL



4.1 MODULE 1 – Injection moulding

4.1.1 Description

The Injection Moulding (IM) module (ID 1) is being developed for Albea pilot line (PL1) and Philips pilot line (PL3). It aims to produce a plastic part by the injection moulding process. The configuration of the module, as described in the job, consists of combining a mould, a material, an injection moulding machine, and a thermal controller. Setting up and monitoring the process are based on the exchange of data among the different Asset Administration Shells of the module's assets, which also include a Digital Twin (DT) of the injection process.

The IM module aims to allow a fast starting up of the production, to maximize product quality and process efficiency as well as to record for each cycle the key parameters for traceability purpose.

At the time of writing this deliverable, a subpart in charge of monitoring the process is being developed and will be added in this module in the near future. This subpart will consist in a multivariate analysis aimed at sending an alert in the event of process deviation and will contribute to an early detection of quality defect.

4.1.2 Architecture

The hardware components of the module are:

- The Injection Moulding Machine (IMM)
- The Thermal Controller Device (TCD)
- The sensorized Mould
- The PC All in One which plays the role of central unit and user interface through its touch screen. As an edge device, it contains the DT of the module.

Depending on the technicality of the plastic material and the technology of the mould, the IM module can be enriched by other physical assets such as a hot runner controller for the mould, a material desiccator and a robot to pick plastic parts from the mould. At this stage of development, these assets are not integrated in the module.

The physical assets are equipped with an OPC UA server to provide the requested data to the related OPC UA clients.

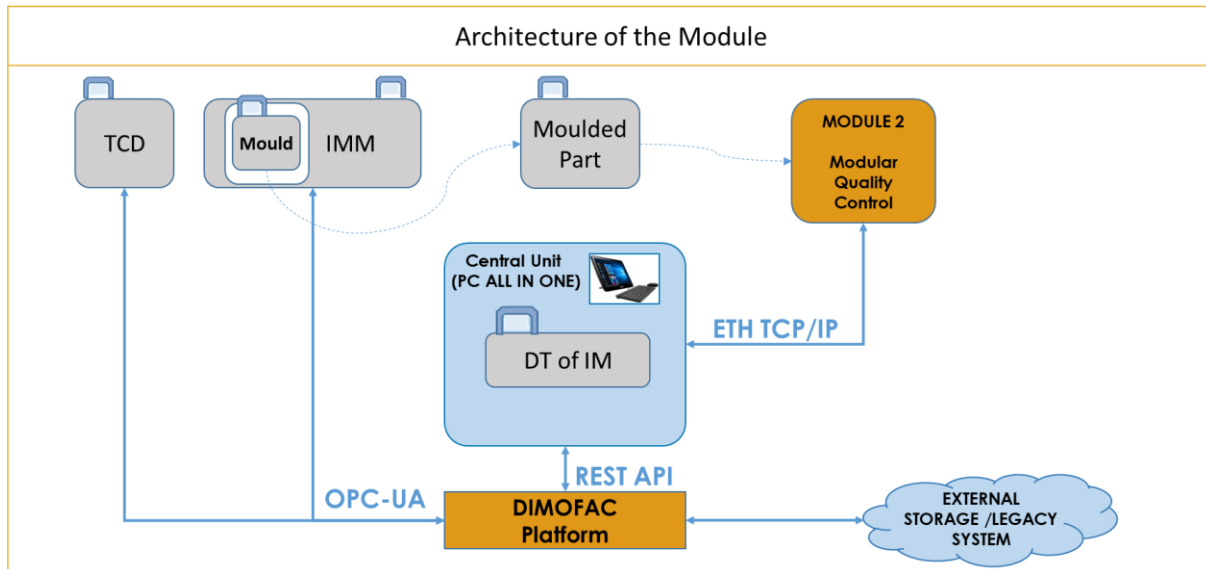


Figure 1 : Architecture of Injection Moulding Module

4.1.3 Interfaces

4.1.3.1 Physical assets-DIMOFAC Platform

The main interfaces between physical assets and the DIMOFAC digital platform are connectors based on the OPC UA protocol. The physical connections are standard Ethernet cable.

4.1.4 AAS modelling

At the current stage of development, the IM module comprises 6 different AASs as listed below:

- IMM_AAS
- TCD_AAS
- Mould_AAS
- Material_AAS
- PlasticPart_AAS
- MouldedPart_AAS
- Job_AAS
- DT_IM_AAS (Digital Twin_Injection Moulding_ AAS)

IMM_AAS, *TCD_AAS*, *Mould_AAS* and *Material_AAS* are the resources needed for the injection moulding process. The *PlasticPart_AAS* is the product definition of the plastic part to be produced whereas the *MouldedPart_AAS* is the instance of the actual physical part and provides information on what was made during the manufacturing stage. The “job” stands for the production order. *DT_IM_AAS* is the AAS of the DT of injection moulding module.

The building of the AASs and their respective submodels has started with the analysis of the use case and the identification of the data to be exchanged between all the assets: for each batch of plastic parts to be moulded, the IM module has to be



DIMOFAC



configured according to the information given by the job. A new configuration of the IM module consists of a combination of a mould, an injection machine, a thermal controller and a material.

For the purpose of interoperability within the plastics industry, the modelling of the AASs was based on the OPC UA Euromap standard of this sector, that is to say Euromap 77, Euromap 83, Euromap 82.1 and Euromap 82.2¹.

Data and objects not described in the Euromap standard has been modelled according to the Common Information Model in development within DIMOFAC.

The AASs were built with the AASX Package Explorer and a brief description of their submodels is detailed below.

4.1.4.1 IMM_AAS and TCD_AAS

The AAS are shown Figure 2 and Figure 3.

AAS "InjectionMouldingMachine_AAS" [IRDI, 1007] of [IRI, https://example.com/ids/asset/6121_6151_1112_8287, Instance]	
SM "Identification" [IRDI, 5001]	
Prop "IMM_ID" = 70T	
MLP "DeviceClass" -> Injection Moulding Machine	
MLP "Manufacturer" -> ARBURG	
MLP "Model" -> ALLROUNDER 370S 700 170A Series5	
Prop "SerialNumber" = 226628	
Prop "SoftwareRevision" = version 5.200 CPU820	
SM "TechnicalData" [IRI, https://example.com/ids/sm/1091_6132_3022_6029]	
SMC "ClampUnit" (6 elements)	
SMC "InjectionUnit" (12 elements)	
SM "InputOperationalData" [IRI, https://example.com/ids/sm/7055_7151_1112_1781]	
Prop "PlastificationVolume" = 20 [cm3]	
Prop "PlastificationCircumferentialSpeed" = 30 [m/min]	
Prop "BackPressure" = 30 [bar]	
Prop "DecompressionVolumeAfterPlastification" = 5 [cm3]	
Prop "ScrewDecompressionSpeed" = 5 [cm3/s]	
Prop "BarrelNominalZoneTemperature_1" = 210 [°C]	
Prop "BarrelNominalZoneTemperature_2" = 220 [°C]	
Prop "BarrelNominalZoneTemperature_3" = 240 [°C]	
Prop "BarrelNominalZoneTemperature_4" = 240 [°C]	
Prop "BarrelNominalZoneTemperature_5" = 240 [°C]	
Prop "BarrelNominalZoneTemperature_6" = 240 [°C]	
Prop "InjectionSpeedProfileNumber" = 2	
Prop "InjectionSpeedProfile1" [cm3/s]	
Prop "InjectionPressureProfile1" [bar]	
Prop "EndOfInjectionProfile1" [cm3]	
Prop "InjectionSpeedLastProfile" = 20 [cm3/s]	
Prop "InjectionPressureLastProfile" = 2000 [bar]	
Prop "VPChangeOverMode" = Volume	
Prop "EndOfInjectionLastProfile" = 6.5 [cm3]	
Prop "HoldPressureProfilePointsNumber" = 2	
Prop "HoldPressurePoint1" = 600 [bar]	
Prop "HoldPressurePoint2" = 600 [bar]	
Prop "HoldPressureRampDuration1" = 5 [s]	
Prop "CoolingTime" = 15 [s]	
SM "OutputOperationalData" [IRI, https://example.com/ids/sm/9555_7151_1112_3062]	

Figure 2 : AASX of Injection Moulding Machine AAS

AAS "ThermalController_AAS" [IRDI, 6745] of [IRI, https://example.com/ids/asset/8420_6170_0112_4915, Instance]	
SM "Identification" [IRDI, 5047]	
Prop "TCD_ID" = Thermo_OPCUA_2019.1	
MLP "DeviceClass" -> Temperature Control Device	
MLP "ManufacturerName" -> HB-THERM AG	
Prop "SoftwareRevision" = SW51-2 2017	
Prop "SerialNumber" = 1935-161	
Prop "YearOfConstruction" = 2019	
Prop "Model" = HB-180Z2-16-4M-A2-405	
SM "TechnicalData" [IRI, https://example.com/ids/sm/0370_8031_4022_1472]	
Prop "CoolingCapacity" = 30 [kW]	
Prop "PowerValue" = 16 [kW]	
Prop "ConnectedLoad" = 17 [kW]	
Prop "NominalFlowRate" = 45 [L/mn]	
Prop "MaxTemperature" = 180 [°C]	
SM "InputOperationalData" [IRDI, https://example.com/ids/sm/1080_6170_0112_3556]	
Prop "PumpControlMode" = 1	
Prop "TemperatureMainLineS" = 40 [°C]	
SM "OutputOperationalData" [IRDI, https://example.com/ids/sm/6234_7080_0112_5214]	
Prop "FlowRateZone1" = 13 [L/mn]	
Prop "TemperatureReturnLineZone1" = 46 [°C]	
Prop "FlowRateZone2" = 12 [L/mn]	
Prop "TemperatureReturnLineZone2" = 45.5 [°C]	
Prop "PressureMainLine" = 4 [Bar]	
Prop "PressureReturnLine" = 2 [bar]	
Prop "TemperatureReturnLine" = 45.7 [°C]	

Figure 3 : AASX of Thermal Controller Device AAS

¹ <https://www.euromap.org/i40/introduction>



DIMOFAC



The unique identifier of an instance of the asset (injection machine or TCD) is the property “*IMM_ID*” or “*TCD_ID*” which is contained in the submodel “*Identification*” of the related asset.

The submodels *TechnicalData* describe the main technical characteristics of the asset.

The *InputOperationalData* submodel contains the parameters for setting the main steps of the process. For the injection moulding machine, the parameters are related to the plasticisation of the polymer, the injection, the holding and the time of cooling. For the Thermal Controller Device, the setting values apply to the cooling parameters of the mould.

The purpose of the *OutputOperationalData* submodel is to gather all the relevant actual values which are measured during a cycle of injection moulding. They are provided by the IMM and the TCD as well as by the sensors in the mould.

4.1.4.2 Mould_AAS

The AAS of the mould is shown in Figure 4.

AAS "Mould_IPC01_aas" [IRI, https://example.com/ids/aas/7530_7141_3022_8940] of [IRI, https://example.com/ids/aas/7530_7141_3022_8940]	
SM "Identification"	[IRI, https://example.com/ids/sm/3471_7141_3022_1907]
Prop "Mould_ID"	= IPC01.002.002
Prop "MouldDescription"	= 1 cavity mould "COQ Box Cover"
MLP "MouldManufacturer"	-> IPC
File "Mould_CADfile"	
SM "MouldTechnicalData"	[IRI, https://example.com/ids/sm/9391_8051_3022_3441]
SMC "Dimensions"	(3 elements)
Prop "Weight"	= 300 [kg]
Prop "ShotVolume"	= 17 [cm3]
SMC "HotRunner"	(2 elements)
SMC "ThermalControl"	(8 elements)
SMC "Cavity"	(6 elements)
SMC "Sensing"	(3 elements)
SMC "Kinematics"	(0 elements)
SM "Mould_Recipe"	[IRI, https://example.com/ids/sm/3113_5151_3022_2249]
SMC "BarrelTemperature"	(3 elements)
SMC "Plastification"	(5 elements)
SMC "Injection"	(5 elements)
SMC "Holding"	(4 elements)
SMC "Cooling"	(2 elements)
SMC "Kinematics"	(0 elements)
SM "Mould_Signature"	[IRI, https://example.com/ids/sm/6554_3132_3022_2447]
SMC "Pressure"	(9 elements)
SMC "Time"	(3 elements)
Prop "MeltCushion"	[cm3]
SMC "MaterialTemperature"	(1 elements)
SMC "MouldCooling"	(4 elements)

Figure 4 : AASX of Mould AAS

The data modeled in the *Mould_AAS* are related to 4 submodels :

1. **Identification**, including notably the property “*Mould_ID*” which defines the unique identifier of a mould including the version in case of interchangeable cavities.



2. **MouldTechnicalData:** it contains the main technical requirements of the mould for choosing the appropriate IMM and TCD for the process. The *Cavity* submodel collection makes the link with the plastic parts that can be produced with the mould by referencing each cavity to CAD file and name of the plastic part. The *Sensing* submodel provides information about the sensors embedded in the mould and which can be used for the monitoring of the process.
3. **Mould_Recipe:** it groups the main parameters for setting the injection moulding machine and the thermal controller. This dataset is built upstream thanks to process simulations which allow to optimize the setting parameters for maximise the quality of the moulded part. Thus, *Mould_Recipe* contains only around twenty quality-relevant setting data whereas a complete dataset for injection moulding process can include hundreds of parameters. It is noted that parameters of *Mould_Recipe* are described using physical units independent of the IMM and TCD characteristics.
4. **Mould_Signature:** it has for purpose to provide the process window able to ensure the good quality of the moulded part. The target values it contains will be compared to some output operational data of injection process within the monitoring model.

4.1.4.3 Material_AAS

Each polymer and colorant required for the moulding of a plastic part will be modelled with its own AAS and can be called within the DIMOFAC platform by a specific *Material_ID* contained in the submodel "*Identification*" (see *Material_AAS* in Figure 5).

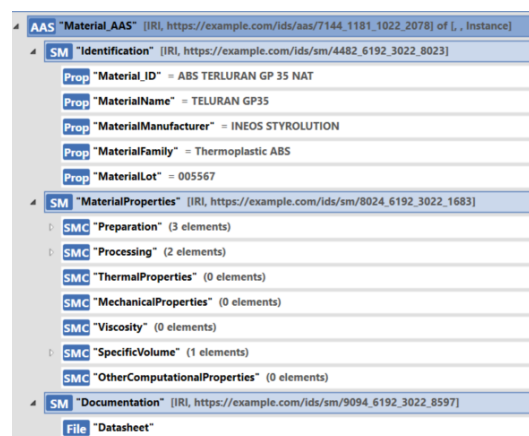


Figure 5 : Material AAS for IM module

The *MaterialProperties* submodel gathers the properties which can be required at different stages of the lifecycle of the product, e.g. recommendation for preparation and processing during the manufacturing stage or viscosity laws for the simulation.

The manufacturer's datasheet file can be recovered in the *Documentation* submodel.



DIMOFAC



4.1.4.4 PlasticPart_AAS and MouldedPart_AAS

The *PlasticPart_AAS* records all the information generated during the engineering stage. If the part can be produced in different colours, there is a specific *PartCode* for each colour. Besides the *PartIdentification* submodel which aims to describe the main properties of the part, the other submodels *PartMaterial* and *BillOfProcess* are used for defining which specific resources are needed to produce the part, that is concerning injection process, the material references (polymer and colorant) and the list of mould_IDs able to form this part.

The *MouldedPart_AAS* records all the information generated during the manufacturing stage, that is the actual resources used for produce the part (thanks to the reference to the *Job_ID*) as well as the actual values of the process parameters. For each injection cycle, a *MouldedPart_AAS* is created (with a unique *Idshort*) and the process information related to this physical part are recorded. If the mould has several cavities, an AAS and *Part_ID* is generated for each moulded part of the shot.

The *MouldedPart_AAS* is linked to the *PlasticPart_AAS* thanks to a reference to the *PartCode* included in the *Design* submodel collection (see Figure 6 and Figure 7).

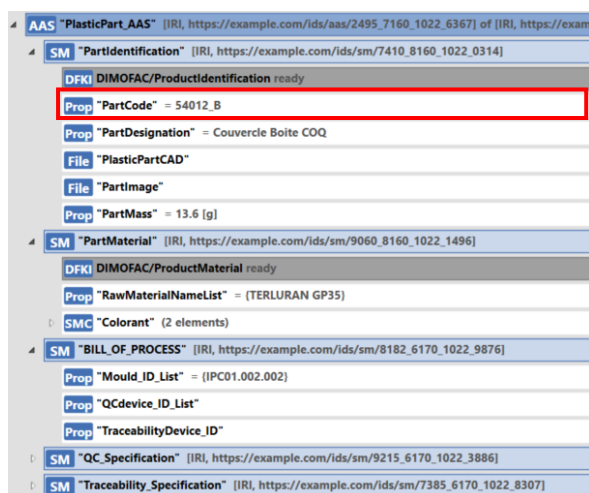


Figure 6 : AAS of the Plastic Part

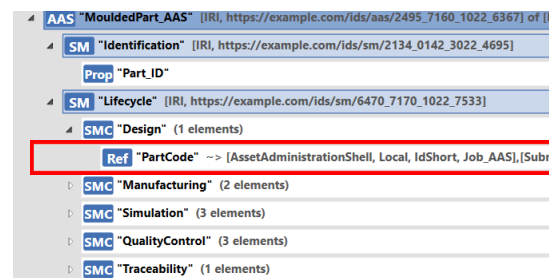


Figure 7 : AAS of a Moulded Part

4.1.4.5 Job_AAS

The *Job_AAS* is built at the top of the module and upstream of the manufacturing process. As the injection moulding is a cyclic process able to produce a batch of thousands of parts in a job, the *Job_AAS* aims to define the plastic part code to be moulded, the quantity of moulded parts, the resources needed and the expected time. As shown in Figure 8, the *Job_AAS* contains three submodels whose main elements are described below.

Job_Information aims to define which plastic part has to be moulded by referencing to the *PartCode* in the related AAS of the plastic part. The physical assets needed for the process are also listed: the asset IDs related to injection moulding are in the



submodel collection *InjectionMoulding* and are referenced to each asset's AAS (see the detail in the red box of Figure 8).

4	AAS "Job_AAS" [IRI, https://example.com/ids/aas/4203_7160_1022_1584] of [IRI, https://example.com/ids/asset/3272_7160_1022_6685]
4	SM "Job_Information" [IRI, https://example.com/ids/sm/5274_6111_1022_4913]
	Prop "Job_ID"
	Prop "PartCode" = 54012_B
	Ref "PartCode" -> [AssetAdministrationShell, Local, IdShort, PlasticPart_AAS],[Property, Local, IdShort, ManufacturerPartCode]
	Prop "PartDesignation" = Couvercle Boite COQ
	Ref "PartDesignation" -> [AssetAdministrationShell, Local, IdShort, PlasticPart_AAS],[Property, Local, IdShort, PartDesignation]
4	SMC "InjectionMoulding" (8 elements)
	Prop "Material_ID" = (Terluran GP35)
	Ref "Material_ID" -> [AssetAdministrationShell, Local, IdShort, PlasticPart_AAS],[Property, Local, IdShort, Material_ID]
	Prop "IMM_ID" = Arburg70T
	Ref "IMM_ID" -> [AssetAdministrationShell, Local, IdShort, IMM_AAS],[Property, Local, IdShort, IMM_Assetid]
	Prop "Mould_ID" = IPC01.002.002
	Ref "Mould_ID" -> [AssetAdministrationShell, Local, IdShort, Mould_AAS],[Property, Local, IdShort, Mould_ID]
	Prop "TCD_ID" = Thermo_OPCUA1
	Ref "TCD_ID" -> [AssetAdministrationShell, Local, IdShort, TCD_AAS],[Property, Local, IdShort, TCD_AssetID]
1	SMC "QualityControl" (4 elements)
1	SMC "Traceability" (3 elements)
1	SM "Job_Schedule" [IRI, https://example.com/ids/sm/9022_6112_1022_9515]
1	SM "Job_Status" [IRI, https://example.com/ids/sm/4011_1131_1022_8002]

Figure 8 : AASX model of Job_AAS

The *Job_Schedule* and *Job_status* provide the required information for plan and monitor the job at production level.

4.1.4.6 DT_IM_AAS

Injection moulding is a complex process comprising several stages where physico-chemical phenomena occur. Many setting parameters can impact the final properties of the plastic part. As part of the DIMOFAC project, several DT have been developed for the injection moulding module thanks to the collaborative work of Siemens, IPC and ESI. Each DT is built on physics-based models and can simulate a specific stage of the process. The ones based on 3D models provide a deep analysis of the mould filling and cooling but have a long computing time which can be afford for engineering stage of the product and virtual commissioning. For supporting the manufacturing stage, several DTs based on Reduced Order Modules (ROMs) and thus able to run in real time have been developed and integrated into the Injection Moulding Module thanks to their respective AAS (see in Figure 9). As each DT provides different information related to the process, the choice can be given to use one or more DT, depending on the need and availability of the model.



DIMOFAC

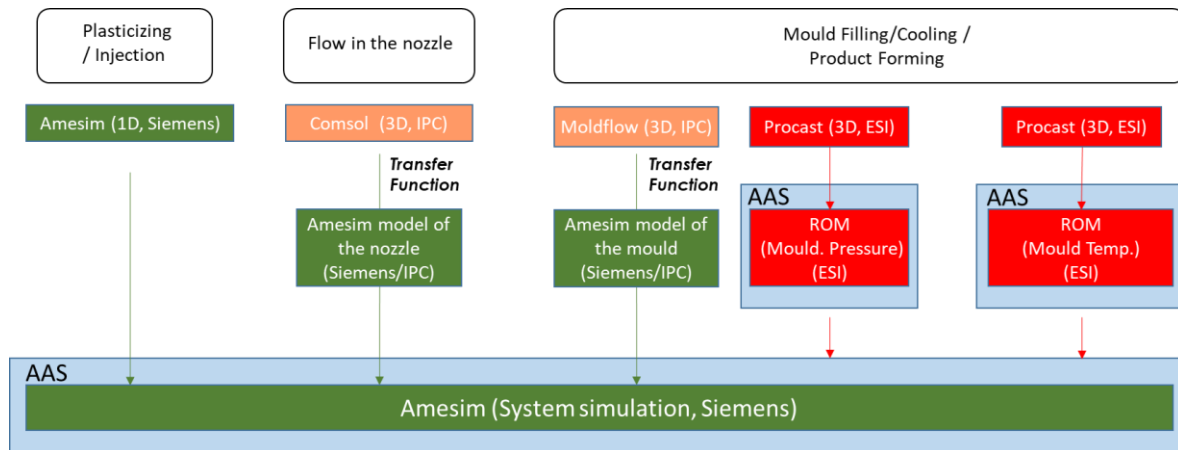


Figure 9 : the 3 AASs for DT of Injection Moulding Module

The AAS aims to connect the DT to the physical assets and to enable real-time data exchange. Some operational data coming from the machines is sent as input data to the DT. This then acts as a virtual sensor by providing output results related to certain physical process parameters such as temperature, pressure and shear rate.

The AAS modelling was based on a common framework. This latter consists of three submodels which are *Identification*, *Capabilities* and *RunSimulation*.

➤ ESI Digital Twins.

Two ROMs have been computed by ESI for IM module (IPC use case). These ROMs are physics-based and can provide both fast and accurate information of the physical asset to the shopfloor. This information is stored in a parametric form, where the user needs to provide the value of some required parameters to get the desired output.

The two ROMs are:

- Temperature of the mould.
- Pressure evolution at the entrance of the mould.

Figure 10 shows the AAS of the mould temperature ROM (left) and the AAS of the pressure evolution (right).

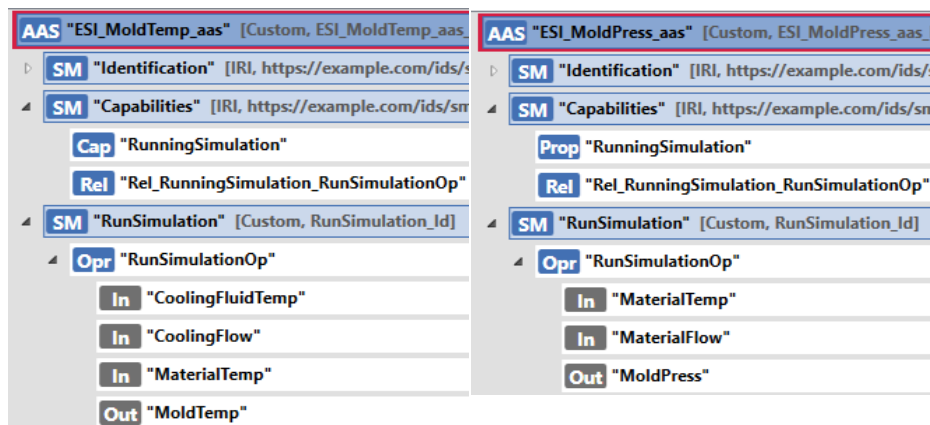


Figure 10 : ESI ROMs AASs for both mould temperature and pressure



The data to be communicated are:

- Input data for the temperature of the mould. Three values are required to predict the mould temperature: the temperature and the flow of the cooling fluid and the temperature of the injected material. It is planned to read these parameters from other AAS as *IMM_AAS* or *TCD_AAS*.
- Output data for temperature. The current implementation provides the maximal temperature achieved inside the mould, which is a single value that can be stored as virtual sensor inside the AAS of the *MouldedPart*.
- Input data for pressure evolution. Sensor data is required to feed the simulation, as well as several parameters required for the simulation: material parameters, process parameters, etc.
- Output data for pressure evolution. The output data is a 'pressure vs time' curve. It is planned to communicate using DIMOFAC platform.

➤ IPC+SIE Digital Twins.

The same structure of AAS can be seen in Figure 11 for the DT co-developed by Siemens and IPC. This DT is based on ROMs of Moldflow and Comsol softwares which are integrated in the Simcenter Amesim tool.

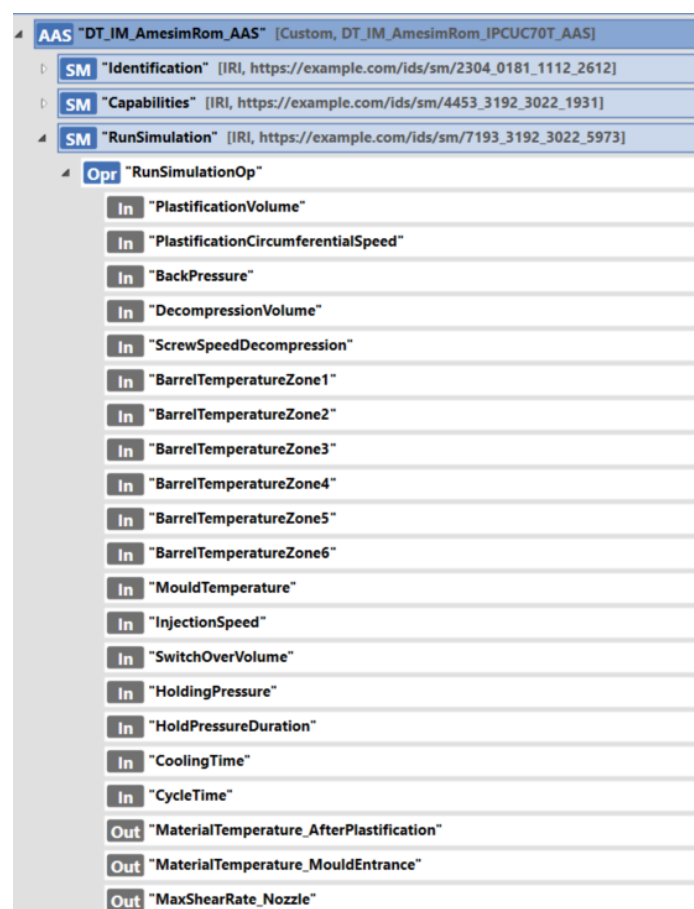


Figure 11 : AAS for DT based on Simcenter Amesim tool



DIMOFAC



The data to be communicated are:

- **Input data :**

Around twenty process parameters from the injection machine and the thermal controller are required to predict the material temperature in the barrel after plasticisation and during the filling stage, and the shear rate in the nozzle. It is planned to read these parameters from the AAS as *IMM_AAS* and *TCD_AAS*.

- **Output data :**

There are three output data : the first is the average value of the material temperature in front of the screw after the plasticisation. The second is the average value of the material temperature at the entrance of the mould during the filling stage. The third is the maximum shear rate in the nozzle during the filling stage.

4.1.5 PnP capabilities

The PPR model followed for building the AASs of the injection moulding module allows to speed up the reconfiguration of an injection moulding cell in function of the plastic part which have to be produced. As it is exposed below, the configuration and monitoring of the module is driven by the *PartCode*.

Thanks to the using of plastics industry Euromap standard for describe the resources and the process, the IM Module can be implemented in most of the current injection cells, whatever the brand of the machine. The independency to the manufacturer is also ensured thanks to the generic modelling of the *Mould_Recipe* as explained before.

4.1.6 Software

For the demonstrator at IPC premises, the orchestration of the dataflow between the different assets of the module is being developed inside the DIMOFAC Platform with the support of Intrasoft. The different steps planned to be demonstrated in this pre-industrial use case are detailed below.

➤ IM Module configuration: from *PartCode* to *Job*

All the information contained in the *PlasticPart_AAS* allows to select the appropriate resources for produce the part. They are gathered as elements of the *JobInformation* submodel contained in the *Job_AAS* as illustrated in Figure 12.

It should be noted that this step occurs upstream the IPC Use Case which begins with the provision of a *Job_AAS_ID*.



DIMOFAC

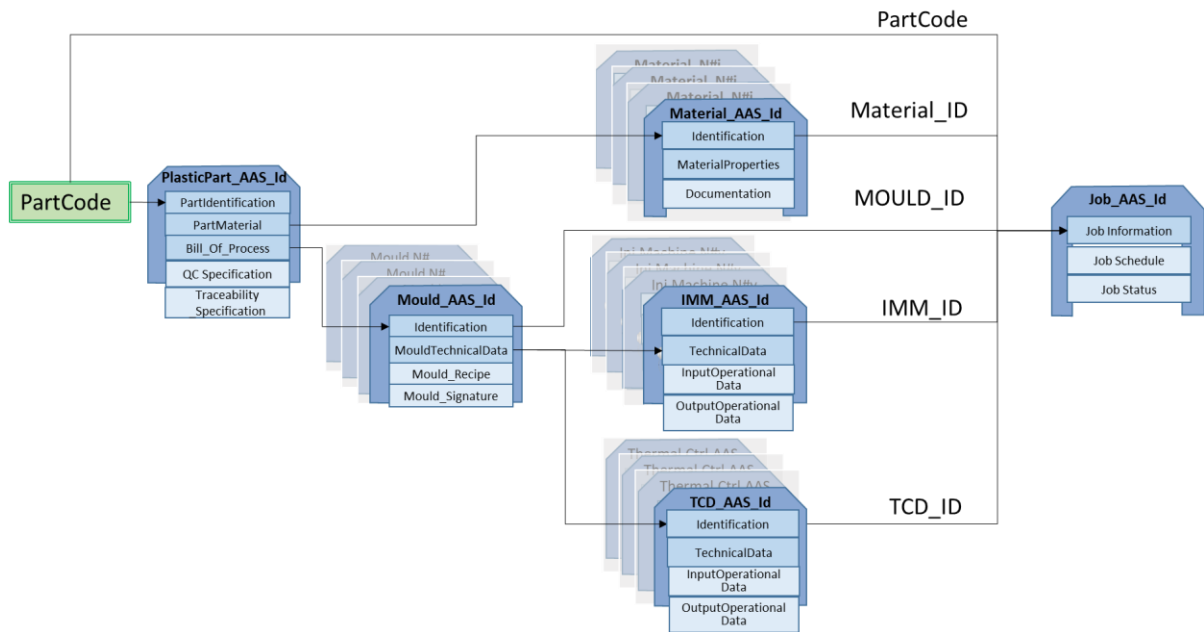


Figure 12 : IM Module configuration from new PartCode

➤ Start-up of production

A user interface connected to the DIMOFAC platform displays the information needed to start the process namely the list of equipment to be connected and set, and the process recipe. The data exchange at this stage is shown in Figure 13. The operator enters the setting parameters provided by the *MouldRecipe* into the control panel of each piece of equipment.

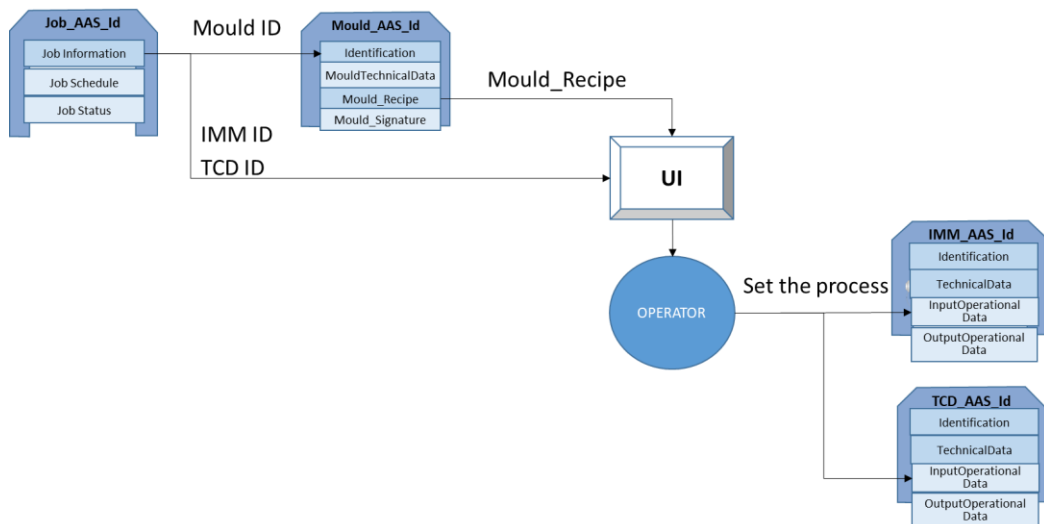


Figure 13 : starting up of the process from Job_AAS

It has to be mentioned that this step should be performed only if the mould is mounted on the machine for the first time: for the other times, a complete dataset file will be already available in the machine and could be retrieved and downloaded for start-up.



DIMOFAC



The operator validates via the UI that the Injection Module is ready for the production and switches the injection moulding machine into automatic mode.

➤ In production: process monitoring and traceability

Once the production is started in automatic mode, the machine sends a digital signal to DIMOFAC platform to start recording the data of the process. Data exchange between equipment and their AAS in the DIMOFAC platform is done via the OPC UA protocol. At the end of each injection cycle an event is sent to the DIMOFAC Platform to trigger the following actions:

- To create a new AAS of the moulded part, to give it a Part_ID and then to copy the actual process values of the submodels *OutputOperationnalData* into the *MouldedPart_AAS*, as illustrated in Figure 15.
- To display the actual process values on the Grafana dashboard providing a real-time monitoring (see Figure 14).
- To run the DT simulation, copy the *Out* results into the *MouldedPart_AAS* (see Figure 15) and display the result on the Grafana dashboard.

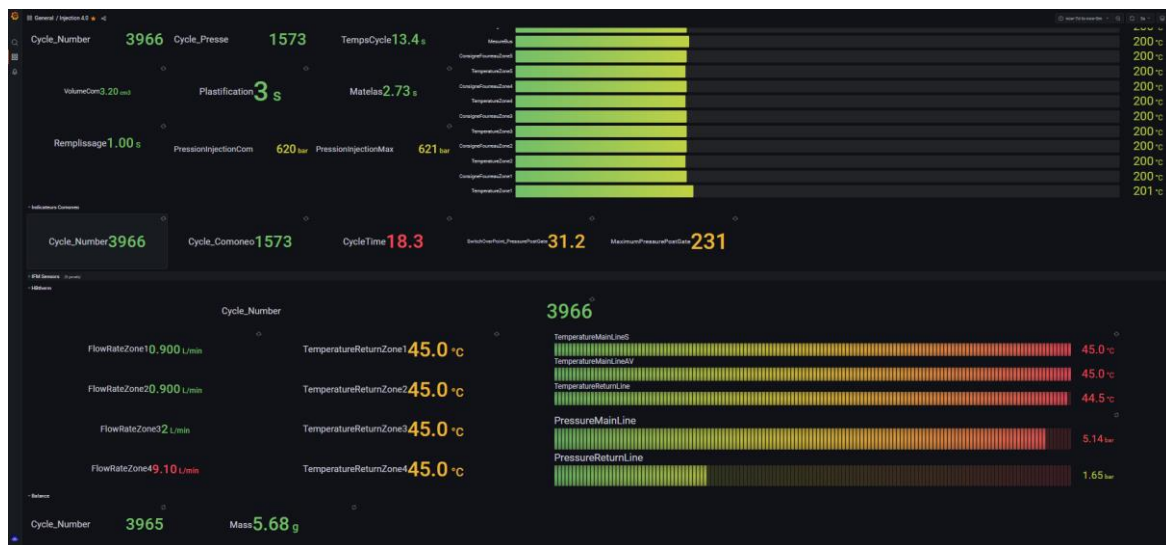


Figure 14 : Real Time data from Module 1 and Module 2 on Grafana dashboard

If the Module 2 is also used for this production, the moulded part is then sent to the Quality Control Module and the *MouldedPart_AAS* will be fed with QC results as it is described in section 4.2.6.

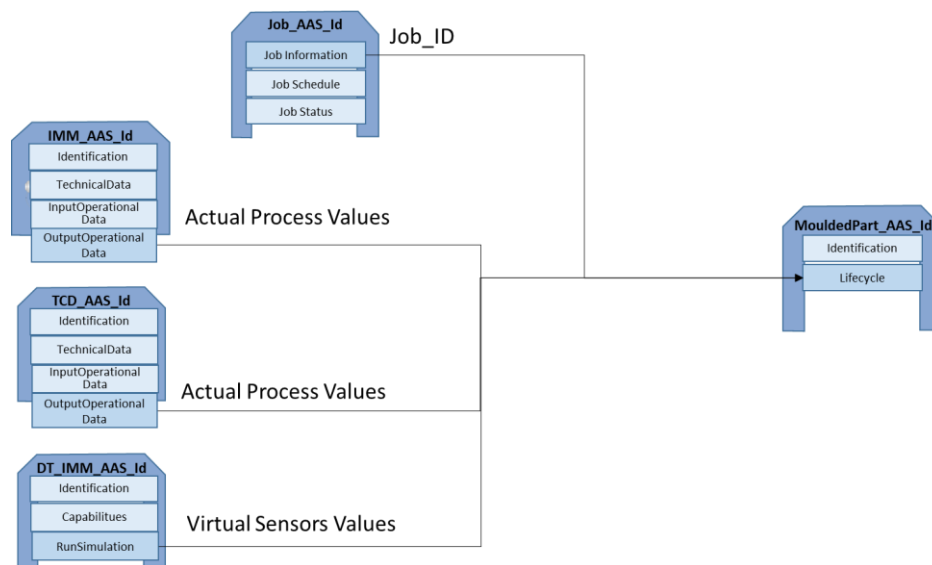


Figure 15 : data recording at each cycle

➤ Software for ESI Digital Twin

The capability of the ROMs is exposed to the rest of the module by REST API connection. This connection allows to call system specific commands on ESI level. In order to connect ESI software with the DIMOFAC platform, a python-based API is used. This API allows to manage many of the software capabilities through python commands. By REST connection, the operations defined in the AAS model are linked with the correspondent python scripts that execute the actions on ESI software. Figure 16 shows a schema of the connection.

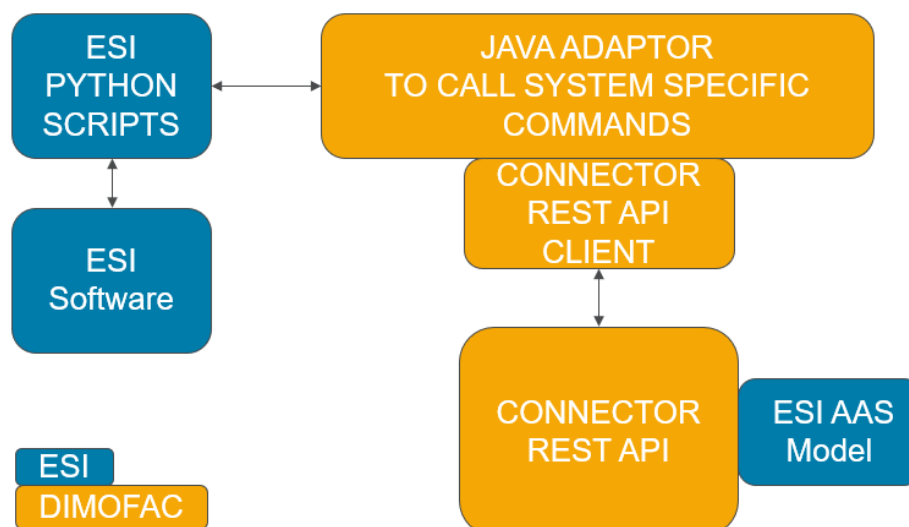


Figure 16: Connection of ESI software to DIMOFAC Platform by Custom connectors

4.2 MODULE 2 – Modular quality control for injection moulding

4.2.1 Description

This module aims to do the online quality inspection of plastic parts just after their moulding. It includes three different quality control stations based on non-destructive technologies and one traceability station.

The Quality Control (QC) stations implemented in the IPC demonstrator are an Infra-red camera cabin, a scale and a vision station composed of three different vision cameras and a various lighting system. The traceability station is a dot peen marking device, but the design of the module allows to replace it by other type of device such as a NFC tag writer.

The AAS modelling makes it possible to exchange data to configure the workcell, save the QC results and traceability of the parts.

4.2.2 Architecture

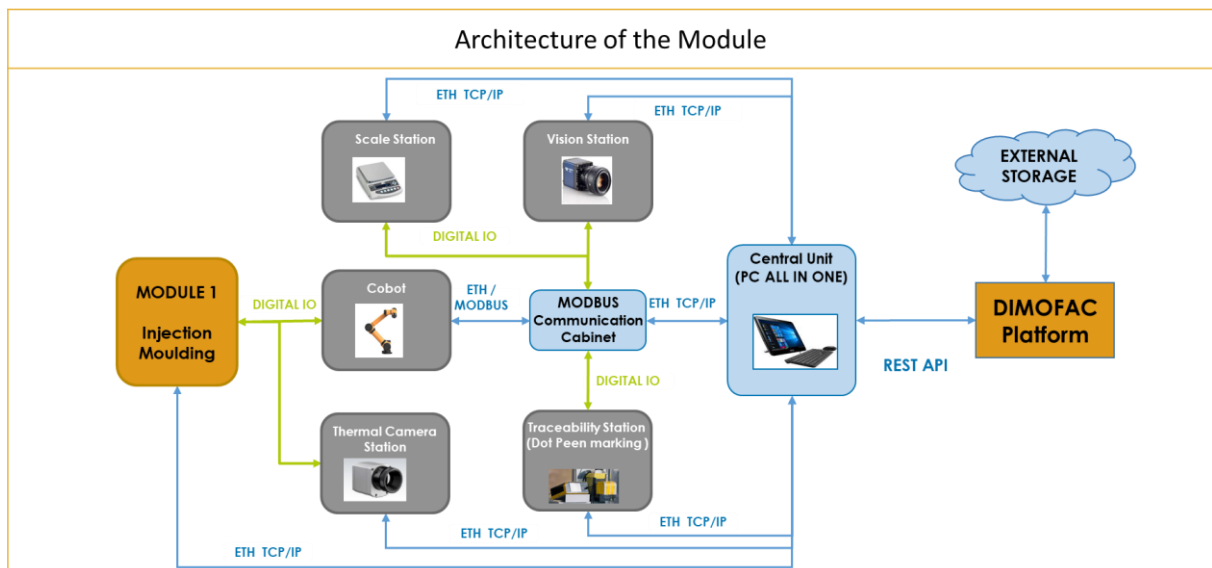


Figure 17 : architecture of Modular Quality Control Module

4.2.3 Interfaces

The communication between the piece of equipment included in the module is based on Modbus connection and ethernet TCP/IP exchanges. The central unit, which is all-in-one PC, communicates with the DIMOFAC Platform thanks to REST API.

4.2.4 AAS modelling

Module 2 is intended to be linked to the IM Module as it aims to perform the inspection of the plastic part sequentially to each injection cycle.

- The AAS modelling of the Module 2 consists of supplementing the AAS modelling of the IM Module by submodels and properties dedicated to quality control as exposed below:
- The PlasticPart AAS



DIMOFAC

During the engineering stage, the product development team defined the online quality control to be performed on the moulded part and the technology for trace the part e.g. Dot Peen Marking or NFC Tag. The list of the quality stations needed for the inspection of the part and the traceability device defined for tracing the part are recorded in the *Bill_Of_Material* submodel of the *PlasticPart_AAS* (see properties *QCDevice_ID_List* and *TraceabilityDevice_ID* in Figure 18).

For the vision inspection, the specific program which has been developed for each part is recorded in the *Vision_Specification* submodel collection (see the property *Path_to_VisionInspectionProgram* in Figure 18).

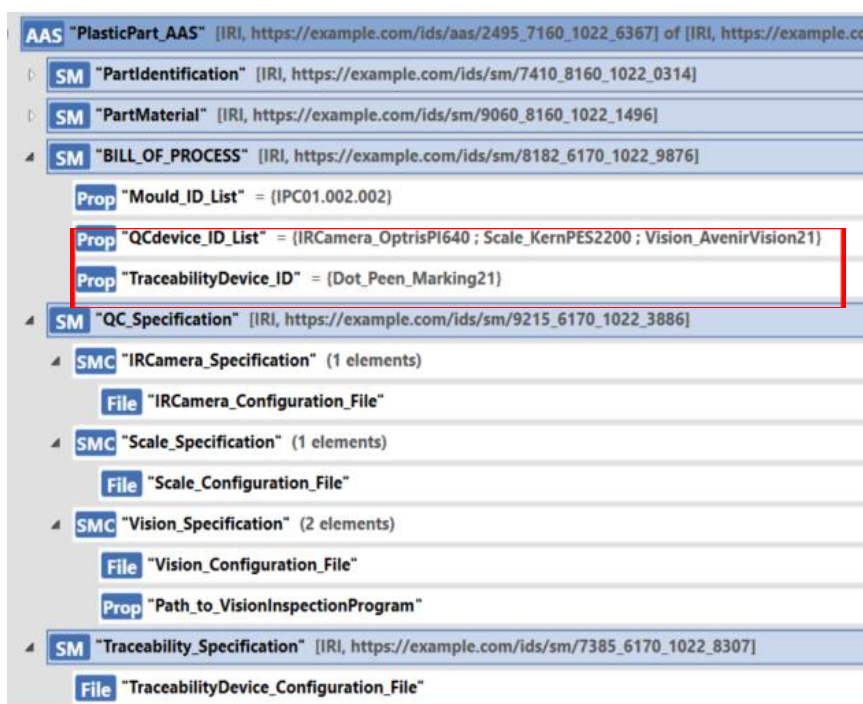


Figure 18 : PlasticPart_AAS related to Module 2

The *QC_Specification* submodel contains the configuration file of each QC device: these files aim to be displayed on the UI and to provide instructions for setting the physical devices e.g. adjusting the position of the camera within its cabin or mounting the correct jig related to the plastic part.

- The MouldedPart AAS

The results of the quality inspection and the traceability code are recorded in the *MouldedPart_AAS*, respectively in the submodel collection *QualityControl* and *Traceability* (see the red box in Figure 19).

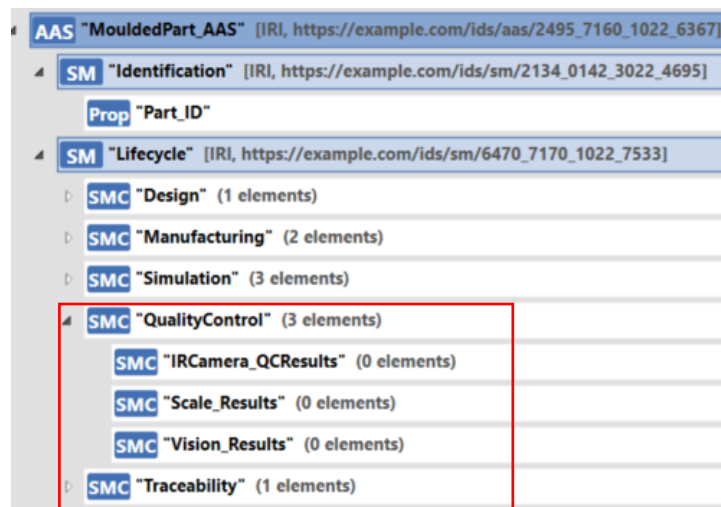


Figure 19 : MouldedPart_AAS including Quality Control Results and Traceability

- The Job_AAS

As quality control and traceability writing cannot be performed independently from the injection process, it seems relevant to integrate this process step into the job of injection moulding and to complete it with the related information.

Thus, the *Job_Information* is supplemented by a submodel collection *QualityControl* and *Traceability* for define which physical device will be used for the QC and the traceability of the part (see the red box in Figure 20).

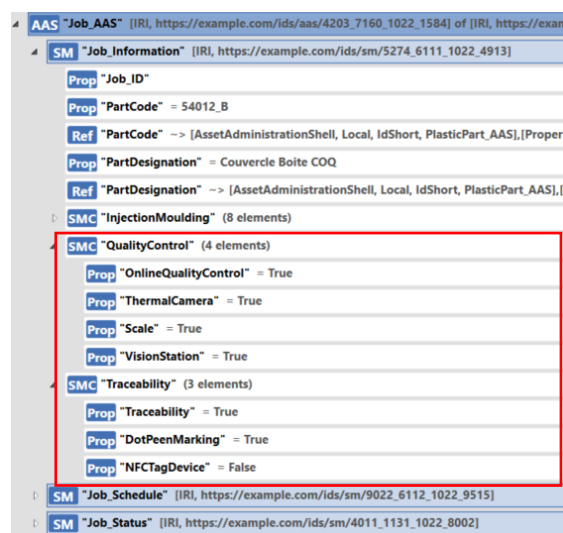


Figure 20 : Quality Control and Traceability in Job_AAS

4.2.5 PnP capabilities

The module has been built for being able to change a QC station or a traceability station by another or to exchange the position of each QC and traceability station within the module.



However, the first version of the demonstrator at IPC will consist in choosing to use or not each station whereas the localisation and connection of each will remain unchanged.

- The information contained in the Job_AAS will allow to configure the quality control cell in accordance to the plastic part code, as illustrated in Figure 21.

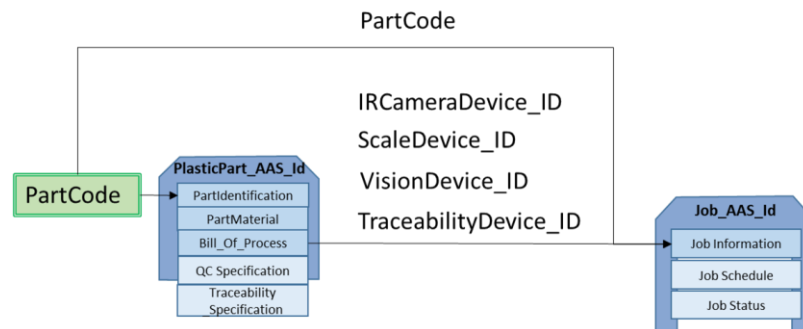


Figure 21 : Module 2 configuration from PartCode

- For the software PnP capabilities, the software will read in the *job_AAS* which QC station and traceability station is needed. The configuration files of each station will be sequentially displayed on the user interface by reading the information from the submodel *QC_specification* of the *PlasticPart_AAS*. This will allow the operator to manually set each station.

The automatic reconfiguration of the Modular Quality Control module for any type of part is feasible but would imply much more effort and it is out of scope of the planned work.

4.2.6 Software

A Python script was implemented for the Modular Quality Control module. This script ensures the communication between the equipment and their software, their AAS and the connection to the DIMOFAC platform through the REST client.

Once the plastic part enters into the quality module, the *Part_ID* created for this part is read from the DIMOFAC Platform and then the quality control process is started. The part goes through the several stations (thermal camera, scale, vision camera and traceability). Each equipment detects the presence of the part and starts processing it. When the control is finished the result is updated in the AAS of Moulded part and displayed on the Grafana dashboard to have a real-time monitoring. From the historical data stored in the *MouldedPart_AAS*, a traceability code is generated and sent via the all-in-one PC to the traceability device for marking the part. Then the part is stored in a buffer. If the quality control result was not correct, the part is evacuated in the scrap.



DIMOFAC

4.3 MODULE 3 – Parts preparation module: laser texturing

4.3.1 Description

The Laser Texturing module will be part of the EIRE PL, but it will not be integrated in the production line. Laser texturing is a non-conventional technique used for creating specific periodic microstructures with positive topographies by molten metal displacement technique. This technique can improve the performance of metallic materials by providing special properties as improving the adhesion by changing the surface roughness. This module is compound by a movement system (linear axis), scanner and laser. In this module the engineer will register the information related with the geometry of the texturing pattern used and the process parameters used to manufacture the parts.

4.3.2 Architecture

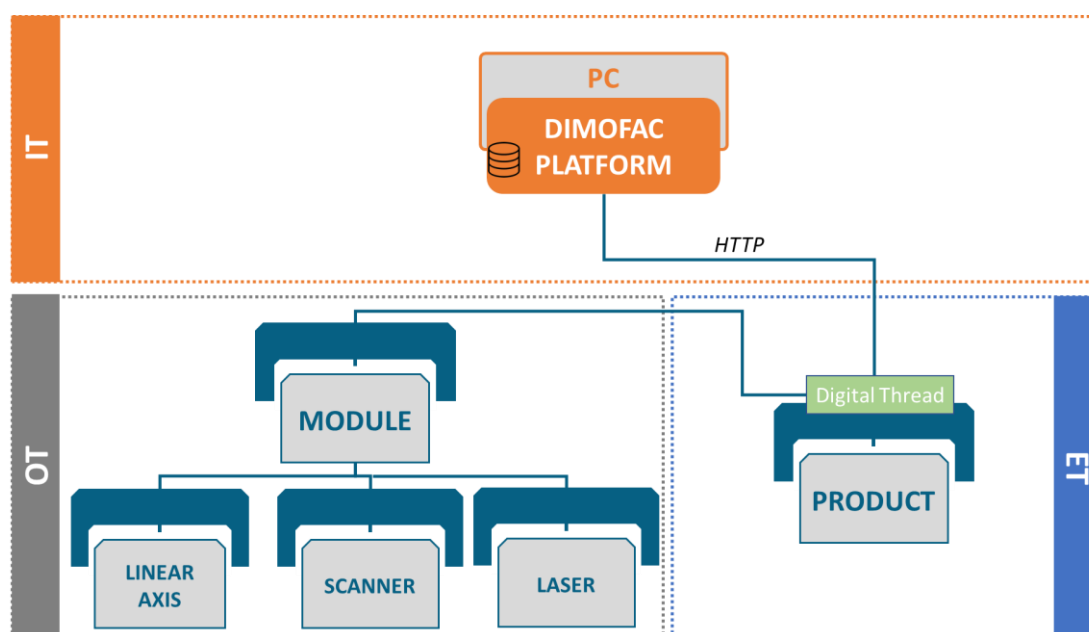


Figure 22: Architecture of the module 3 (texturing)

4.3.3 Interfaces

The first step to communicate with DIMOFAC platform is the register of the different assets. For doing this, it is necessary to export the AAS and the different Submodels (described in the next section) in JSON format. The JSON information will be registered using HTTP methods on the DIMOFAC Platform and will be necessary to describe what type of format is expected. Once the assets are register, the user using the application Digital Thread can register/trace all the information associated to a specific product. The Digital Thread application will communicate with the platform through HTTP REST API protocols.

4.3.4 AAS modelling

The AAS modelling of the different assets follows the specifications described in T1.4 where the Common Information Model is defined. Due to the legacy systems used in



DIMOFAC

this module (laser and scanner) and the speed of the process (i.e., 9 inserts for EIRE PL in 15 seconds), the assets modelled in this module are the Type 1² (serialized files that contain static information and may be distributed as a files). These are the assets that compound the texturing module:

- **Linear axis.**

This model (Figure 23) will store the position of the linear axis when the part is being manufactured.

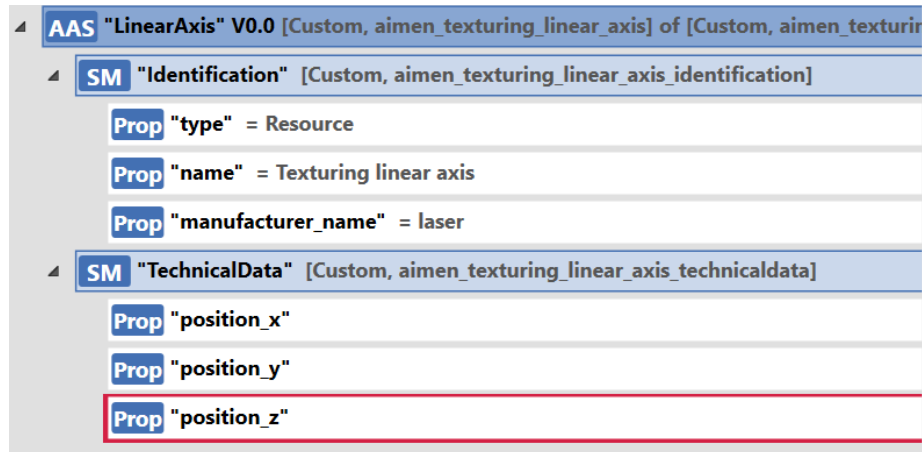


Figure 23: Linear axis AAS model

- **Scanner and Laser.**

These models (Figure 24) will register the information related with the process parameters that the operator used for manufacturing the parts.

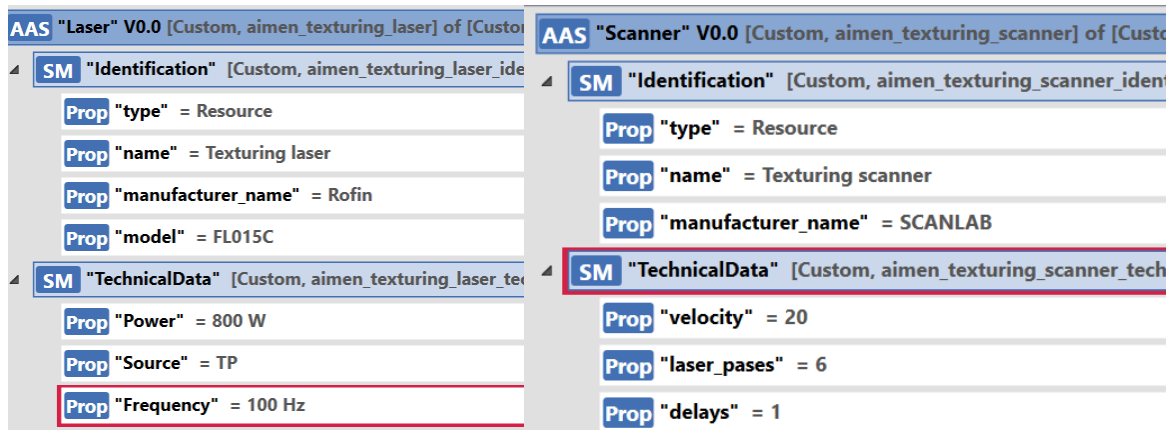


Figure 24: Scanner and laser AAS model

- **Module.**

This asset (Figure 25) represents the whole physical texturing cell. The submodel *AasComposition* references to the different assets that compound this module.

² [BaSyx / Documentation / AssetAdministrationShell - Eclipsepedia](#)

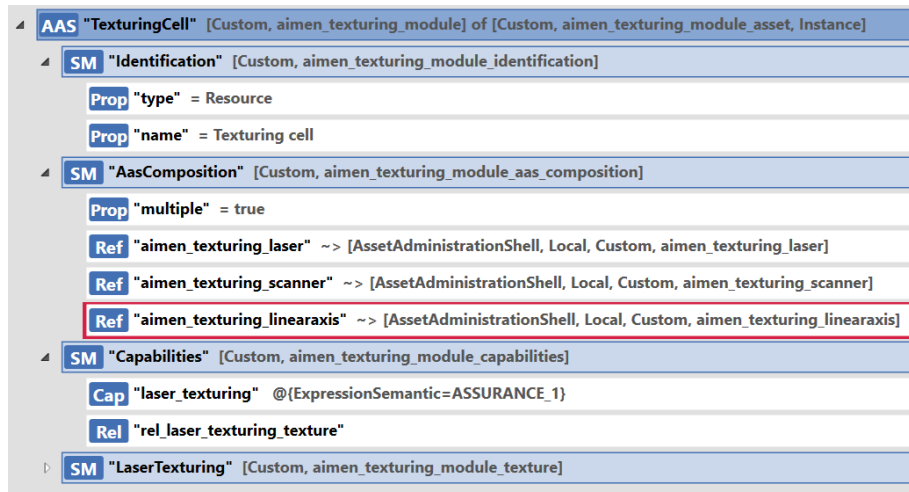


Figure 25: Texturing module AAS

- **Product.**

In this model (Figure 26) the user, through the Digital Thread application, will trace all the information related with the product (i.e., design, physical assets used during manufacturing, and files generated during manufacturing)

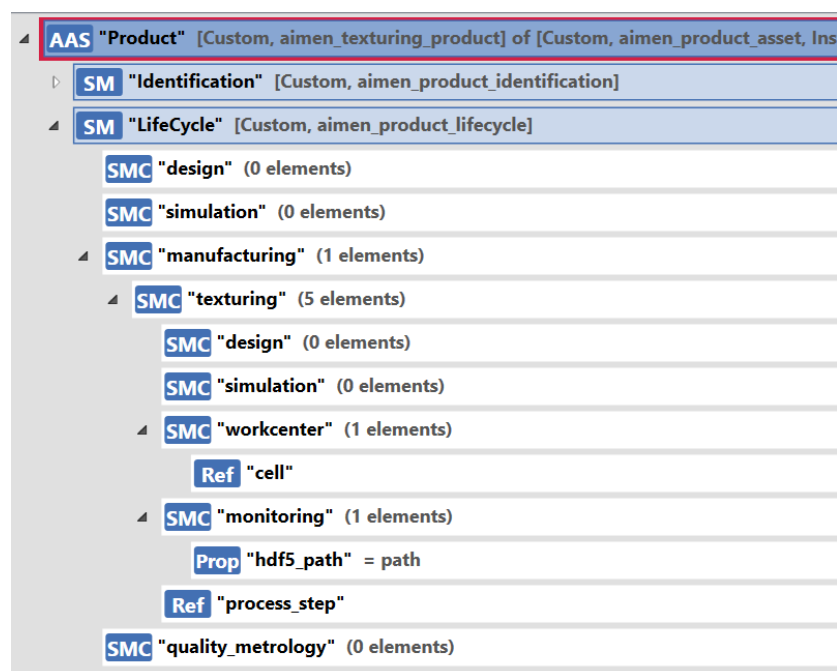


Figure 26: Product AAS

4.3.5 PnP capabilities

Due to this module is managed as AAS Type 1, this module will not have PnP capabilities for orchestrating the different physical assets that compound the module.

4.3.6 Software

The software used for managing the data generated in this module will only manage product information. This software is module independent and will be also used in Module 5 and Module 8. The user will generate a Product AAS and save the



information related with the manufacturing stage in DIMOFAC Platform. The user will also check the information related with the a specific product (i.e., files used for Simulation, the monitoring data,...) in an intuitive way.

4.4 MODULE 4 – Processing module: induction joining

4.4.1 Description

The induction joining module is part of EIRE PL and consists of welding composite parts using induction technology. This module is composed by a movement system (robot) in charge of moving the coil that will be connected to a generator. For controlling this process this module uses a pyrometer that will measure the temperature in the composite panel. These physical assets will be connected to a central unit (industrial PC) in charge of publishing the process data in the DIMOFAC platform and getting information of the product to be manufactured. For taking the decision of the process parameters the physical simulation (software: ProCAST) will store in the *Product AAS* the parameters that can be check by the operator using the Digital Thread application.

4.4.2 Architecture

In this module all physical devices (robot, generator, pyrometer) will be connected to a Profibus master device, which will allow to add more slave to the network and communication between them. The DTs will communicate with the platform through HTTP protocols. In Figure 27 the communication protocols are specified.

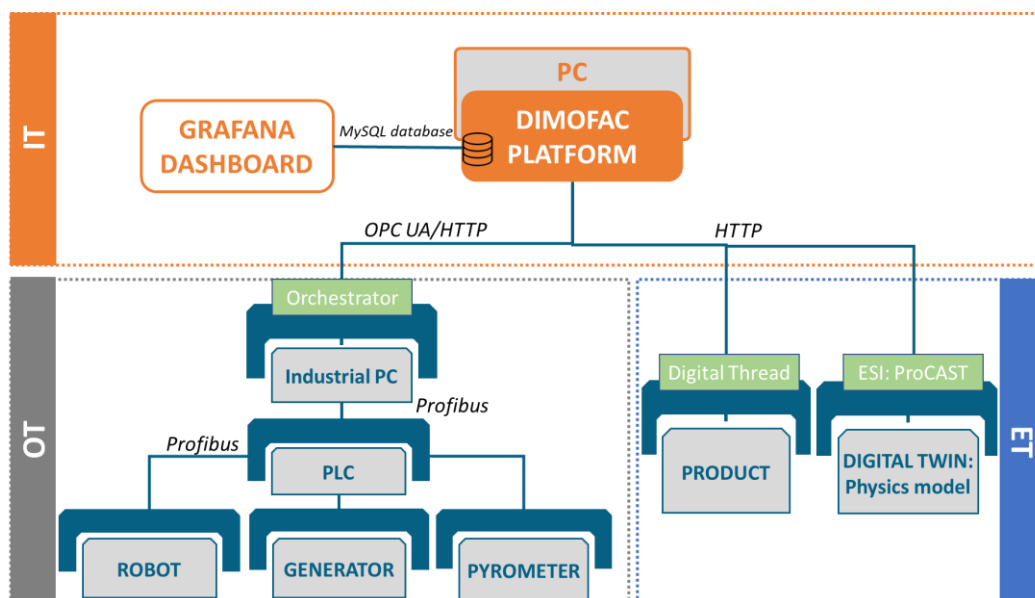


Figure 27: Induction joining architecture

4.4.3 Interfaces

4.4.3.1 Physical assets – central unit

- Profibus



DIMOFAC

The data of all the devices of this module is accessible through Profibus protocol. The PLC is the Profibus master that communicate with the generator, pyrometer and robot. The PLC group all the devices' data needed to the monitor and control process and sent they to the Jetson device, which is a slave in the Profibus network.

The implementation of the Profibus communication will be declared in the D3.4 deliverable.

- **Central unit – DIMOFAC Platform**

The first step to communicate with DIMOFAC platform is the register of the different assets. For doing this, it is necessary to export the AAS and the different Submodels (described in the next section) in JSON format. The JSON information will be registered using HTTP methods on the DIMOFAC Platform and will be necessary to describe what type of format is expected.

The DIMOFAC Platform not only allows us to communicate the module with the different assets using different communication protocols but also record the information using a MySQL Database. This Automated Tape Laying (ATL) module will represent in real time some key information in a Grafana dashboard.

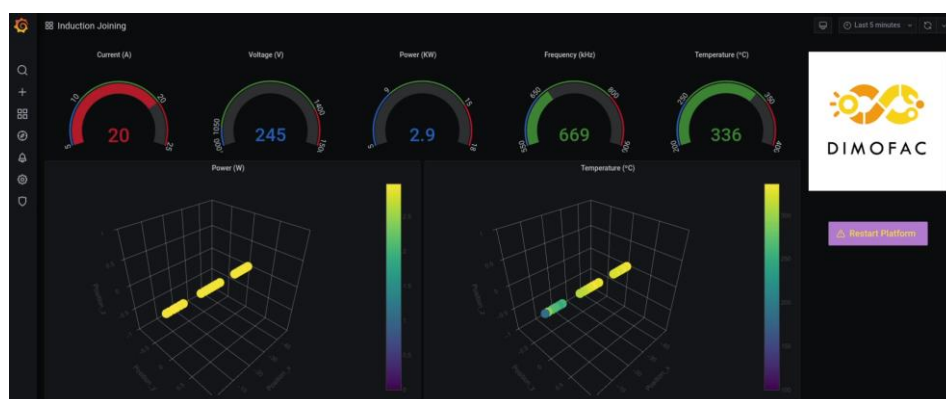


Figure 28: Grafana panel for the Induction Joining module

- **OPC UA**

The industrial PC device publishes the parameters' values of the Operational Data and Parameters submodules (described in the following section) through OPC UA server.

- **REST API**

The industrial PC device receive the information of the products that have been created through the digital thread application. This data is consulted to the DIMOFAC Platform by the industrial PC through a GET method in REST API.

When the process has finished, the industrial PC send to the platform the new information that have been created in the manufacturing process. This information is sent to the platform through a PUT method in REST API.



DIMOFAC



4.4.4 AAS modelling

The AAS modelling of the different assets follows the specifications described in T1.4 where the Common Information Model is defined. For this module, the assets needed for achieving PnP capabilities in Induction process are the following ones:

- **Robot**

This asset has the two main capabilities (*move* and *communicate*). In this asset the key information is the position that will be send to the central unit to be synchronize with the process parameters.

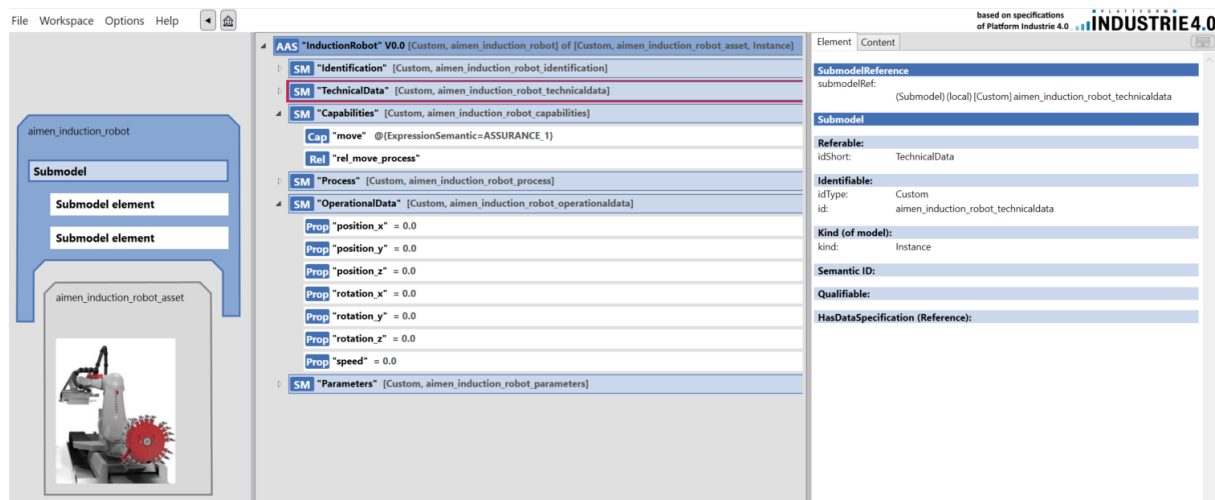


Figure 29: Robot AAS

- **Generator**

This asset will be connected to a master Profibus PLC and the data that can be read/written is the one specified in the submodels *OperationData* (read) and *Parameters* (write).



Figure 30: Generator AAS



DIMOFAC

- **Pyrometer**

This asset (Figure 31) will be also connected to the master Profibus PLC and it will be used as sensor for the development of the controller that will be described in D3.4. The parameter needed for the configuration of this device is the emissivity (*Parameters* submodel) and the output will be the temperate (*OperationalData* submodel).

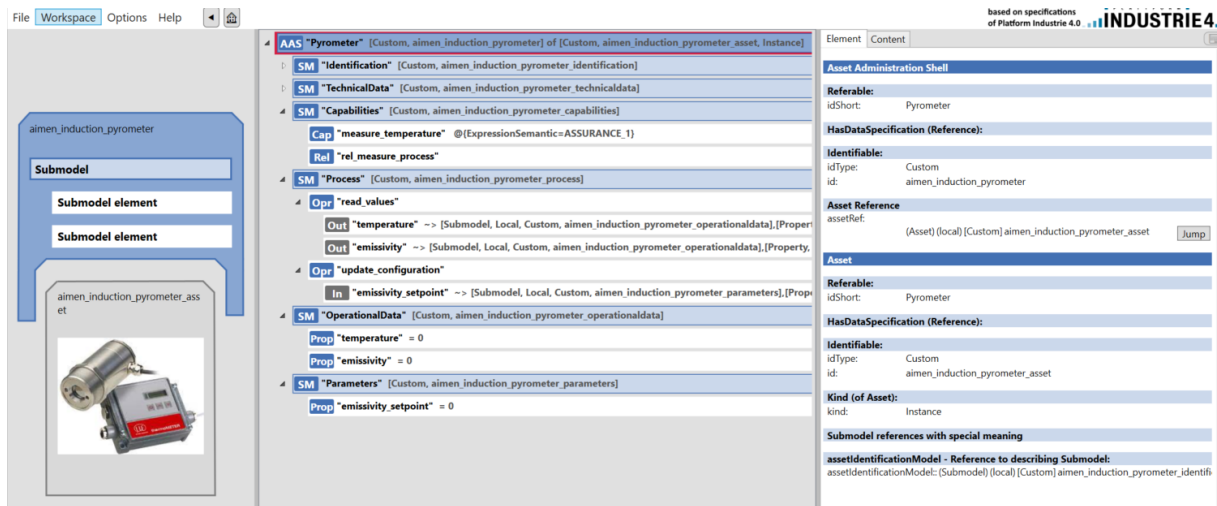


Figure 31: Pyrometer AAS

- **PLC**

This PLC is in charge of communicating the central unit/industrial PC with the different assets. The asset will be composed (*AasComposition* submodel) by the robot, generator, and pyrometer. The main capability of this asset is the Profibus communication, that will be specified in the submodel *Configuration_Profibus*, in which the different parameters will be defined for establishing the communication and allowing the reading/writing.

- **Module.**

This asset represents the whole induction joining module and the industrial PC capabilities. This asset is a Profibus slave and will communicate with the master PLC for publishing data through OPC UA server and getting product information through HTTP protocol.

- **Induction Process and Product**

These two assets are used for linking the manufacturing information with the specific product information. The Product AAS will record/trace all the information generated during the engineering and manufacturing stages as well as it will reference to the specific assets.



DIMOFAC



4.4.5 PnP capabilities

The PnP capabilities for this module are achieved by following a CIM. The central unit is able to decide what physical assets are available and start the communication (Hardware PnP capability), and it is also able to decide the assets/modules needed for manufacturing a specific product as well as run a specific simulation (Software PnP capability).

For hardware PnP capability, the software ('Orchestrator') will try to establish the communication with the module and physical assets that compound this module. So, for doing that, the software will take the configuration specified in the Submodel Communication and execute the corresponding method with the attributes specified in the Submodel (i.e., type Profibus, configuration: settings and variables). If the asset is not physically connected to the central unit, the software will not allow start the recording of the process.

For software PnP capabilities, the software ('Orchestrator') will take the decision of what module is needed for manufacturing a specific product. The AAS of the product will reference the processes that need to be performed. Then, the software will search the process AAS, which specifies the capabilities needed for executing the process, and the module that can perform these specific capabilities (e.g., 'induction welding', 'measure temperature', 'move'). Therefore, as the software is based on the Capabilities Submodel, the user does not need any programming task.

4.4.6 Software

The sequence and iteration that the user has to perform with the different software involved in this module is: the user will start with the creation of an asset of the product using the Digital Thread application. Then, for optimising the process parameters of the induction process, the user will run a simulation using the ESI Software (ProCAST). The result of the simulation (optimised process parameters) will be stored in the DIMOFAC Platform. For starting the manufacturing, in the Orchestrator software the user will select from a list the product to be manufactured, the module available for this manufacturing and the process parameters for running the process. During manufacturing, the user will be able to see the status of the process using the dashboard created in Grafana and/or the Orchestrator with the real time information of the process parameters. Once the manufacturing will have finished, the user could check that the information related to the manufacturing would have been automatically linked with the product previously selected.

ProCAST is an ESI tool dedicated to physics-based DT, and it is mainly dedicated to casting process. ProCAST can also be linked to an ESI electromagnetic solver. The simulation provides information about thermal behaviour, magnetic field, joule heating, etc. This software is used in EIRE PL to simulate the thermal behaviour of the



DIMOFAC



part during induction welding to optimise process parameters. Detailed information is provided in deliverable D3.3.

A ProCAST induction welding application has been created for this use case to develop a standard application for other similar projects. An AAS model has been also created, allowing an easy integration of the induction welding application into Industry 4.0. For this particular induction welding application, manufacturing process has been analysed to identify key aspects of the process. These aspects have been introduced in a simulation workflow matching the real process. This process is then modelled in an ESI workflow manager tool considering 4 main steps:

5. **Load case:** different induction welding could be managed at the same time in a factory. This step allows to load the correspondent model to the current use case.
6. **Update data:** identified parameters to be optimized are coil height, temperature threshold and coil path. This step updates into the model these data.
7. **Run simulation:** sets the parameters correspondent to the solver and runs the simulation.
8. **Retrieve data:** this step gets the desired result from the software.

The AAS model of the application has been created in accordance with the Common Information Model. It contains 4 capabilities that are identified with the 4 main steps of the application. The first capability, Update Data, is implemented by 3 skills: update coil height, update temperature threshold and update coil path. Input and output data are included in the correspondent submodel collections. Figure 32 shows some of the information included inside the AAS model.

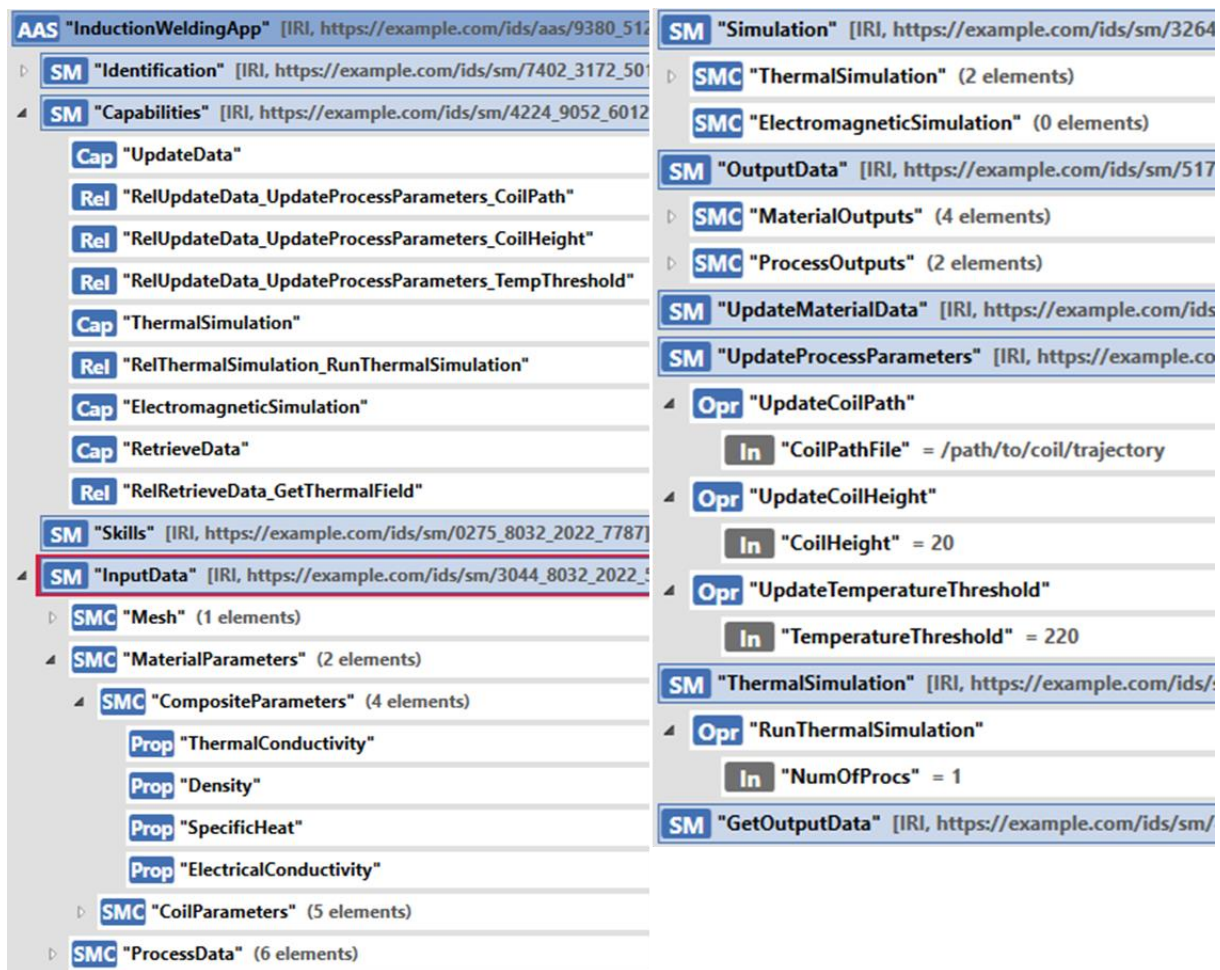


Figure 32: Induction Welding application AAS (ESI)

The capabilities are exposed to the rest of the module by REST API connection. The information about the connection is detailed in software section inside Module 1 (Figure 16).

The data to be communicated during the process are:

- **Input data:**

data to be update into the model. These are coil height, coil path and temperature threshold.

- **Output data:**

these data consist of geometrical information (coordinates of the nodes of the mesh) and the temperature value per node. It is planned to use HDF5³ standard to exchange this information with AIMEN, which will interpret the data to monitor the process with the most suitable way: 3D representation, curve plots, etc.

³ <https://www.hdfgroup.org/solutions/hdf5>



4.5 MODULE 5 – Processing module: laser joining

4.5.1 Description

The Laser Joining module is part of the EIRE pilot line. It is composed of:

- **Laser joining cell,**

It is preassembled at TECNALIA facilities, It is basically composed of the laser source, the laser head and the fixture or clamping device.

- **A Python Flask server,**

It receives the data from the laser joining cell (laser head and fixture) from HTTP POST calls, and it is installed in a TECNALIA virtual machine.

- **Digital data management platform based on Intrasoft AAS server implementation**

It receives the data from the Laser Joining Cell via the Python Flask server and stores it in a database. Inside the Intrasoft AAS server the data is stored according to AAS standard. This digital data management platform is installed also in a TECNALIA virtual machine.

At the time of this deliverable is being written a subpart in charge of performing analytics is being developed and it will be added in this module in the near future. The aim of this subpart will be to serve as input to the control part of the laser using Machine Learning algorithms.

4.5.2 Architecture

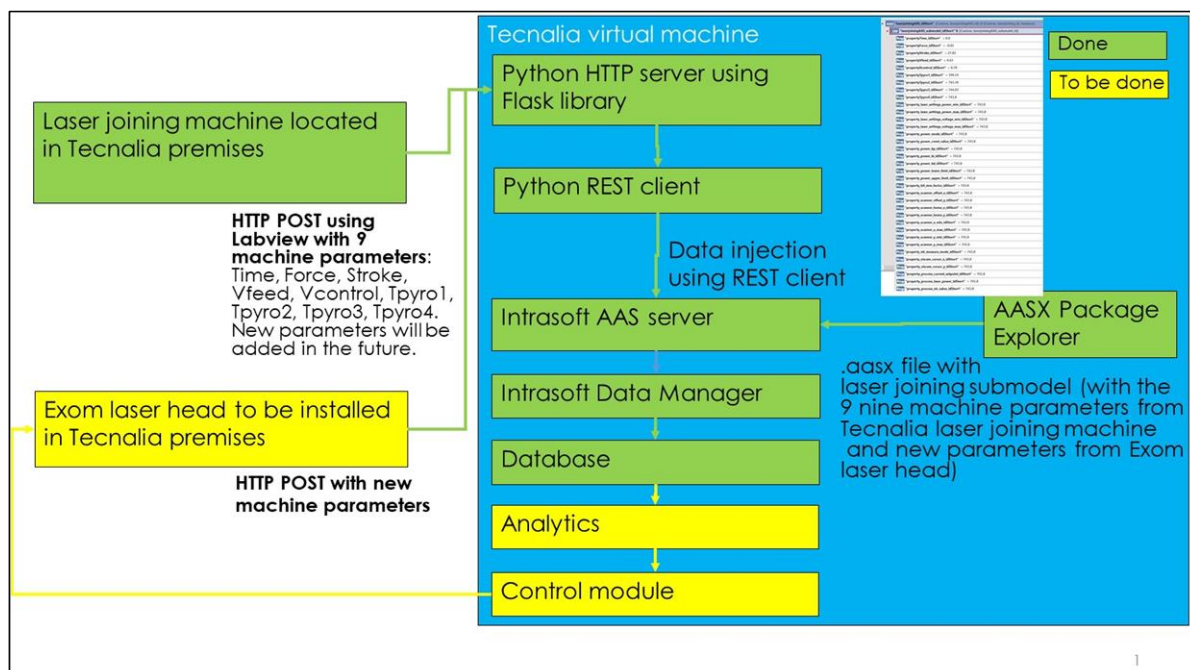


Figure 33: Architecture of the laser joining module

The EXOM laser head and TCN fixture device communicate with the Python Flask server using HTTP POST calls. The server sends the data to a Python REST client which is



DIMOFAC

in charge of injecting the data into Intrasoft AAS server. The Intrasoft AAS server automatically stores all the data into a database which will be used by the analytics part of the module.

The result obtained by performing analytics will feed a control module which will be communicated with the laser cell. The laser head and the fixture device will retrieve the control commands continually from the control part of the digital data management platform by means of HTTP GET calls.

The digital data management platform comprises the Python HTTP server, the Python REST client, the Intrasoft AAS server, the Intrasoft Data Manager, the Database, and the Analytics and Control part. This digital data management platform is installed in a TECNALIA virtual machine.

4.5.3 Interfaces

In our case the main interfaces are based on REST. The laser joining cell communicates with the TCN virtual machine using HTTP POST and HTTP GET calls. The Python HTTP server which be seen in the architecture figure injects also the data into the Intrasoft AAS server using a REST client.

At the moment, the AAS server of this module has not communication with other AAS servers.

4.5.4 AAS modelling

The variables from the laser fixture device from TCN which are stored into the AAS are the following:

- **Time (s)**
- **Force (kg)**
- **Stroke (mm)**
- **Vfeed (bar)**
- **Vcontrol (V)**
- **Tpyro1 (C)**
- **Tpyro2 (C)**
- **Tpyro3 (C)**
- **Tpyro4 (C)**

The variables from the EXOM laser head that will be installed in Tecnalia premises are also stored into the AAS and are the ones listed in Table 2.



Table 2: EXOM laser head parameters stored into the AAS server

bits	type	Variables	bits	type	Variables
32	uint32	Scanner head status	32	uint32	Scanner head status
Scanner/Designer Settings			Control (Process)		
32	uint32	Bit/mm factor	32	uint32	Control mode (constant or PID)
32	float	Offset X	32	float	Kp (PID variable)
32	float	Offset Y	32	float	Ki (PID variable)
32	float	Home X	32	float	Kd (PID variable)
32	float	Home Y	32	float	Set-Point
32	float	X min limit (working area)	32	float	Power upper limit (set)
32	float	Y min limit (working area)	32	float	Power lower limit (set)
32	float	X max limit (working area)	32	float	Laser power output
32	float	Y max limit (working area)	32	float	NIR Camera feedback (process variable)
Real Time Modifications for Wobbles			Laser settings		
32	float	Rotation angle	32	float	Min. Power [W]
32	float	Offset X	32	float	Max. Power [W]
32	float	Offset Y	32	float	Min. input voltage [V]
32	float	Scale X	32	float	Max. input voltage [V]
32	float	Scale Y	VIS-Camera		
NIR-Camera settings			32	uint32	FPS
32	uint32	Measurement mode	32	uint32	Exposure time [us]
32	uint32	X pixel (Mode 3)	32	float	Gain
32	uint32	Y pixel (Mode 3)	32	uint32	Auto exposure time
32	uint32	Width X pixel (Mode 3)	32	uint32	Auto gain
32	uint32	Height Y pixel (Mode 3)	Wobbles		
32	float	Gain	32	uint32	Local wobbles (in PC)
32	float	Bias	32	uint32	Remote wobbles (in scanner head)



Currently, there is an .aasx file with one submodel. The submodel contains all the properties from both the laser fixture device and laser head.

Figure 34 depicts all the variables in one single submodel.

AAS	"laserjoiningAAS_idShort"	[Custom, laserjoiningAAS_id] of [Custom, laserjoining_id, Instance]
SM	"laserjoiningAAS_submodel_idShort"	V. [Custom, laserjoiningAAS_submodel_id]
Prop	"propertyTime_idShort"	= 0.0
Prop	"propertyForce_idShort"	= -0.03
Prop	"propertyStroke_idShort"	= 27.83
Prop	"propertyVfeed_idShort"	= 0.63
Prop	"propertyVcontrol_idShort"	= 0.70
Prop	"propertyTpyro1_idShort"	= 199.14
Prop	"propertyTpyro2_idShort"	= 745.10
Prop	"propertyTpyro3_idShort"	= 744.93
Prop	"propertyTpyro4_idShort"	= 743.8
Prop	"property_laser_settings_power_min_idShort"	= 743.8
Prop	"property_laser_settings_power_max_idShort"	= 743.8
Prop	"property_laser_settings_voltage_min_idShort"	= 743.8
Prop	"property_laser_settings_voltage_max_idShort"	= 743.8
Prop	"property_power_mode_idShort"	= 743.8
Prop	"property_power_const_value_idShort"	= 743.8
Prop	"property_power_kp_idShort"	= 743.8
Prop	"property_power_ki_idShort"	= 743.8
Prop	"property_power_kd_idShort"	= 743.8
Prop	"property_power_lower_limit_idShort"	= 743.8
Prop	"property_power_upper_limit_idShort"	= 743.8
Prop	"property_bit_mm_factor_idShort"	= 743.8
Prop	"property_scanner_offset_x_idShort"	= 743.8
Prop	"property_scanner_offset_y_idShort"	= 743.8
Prop	"property_scanner_home_x_idShort"	= 743.8
Prop	"property_scanner_home_y_idShort"	= 743.8
Prop	"property_scanner_x_min_idShort"	= 743.8
Prop	"property_scanner_x_max_idShort"	= 743.8
Prop	"property_scanner_y_min_idShort"	= 743.8
Prop	"property_scanner_y_max_idShort"	= 743.8
Prop	"property_nit_measure_mode_idShort"	= 743.8
Prop	"property_viscam_cursor_x_idShort"	= 743.8
Prop	"property_viscam_cursor_y_idShort"	= 743.8
Prop	"property_process_current_setpoint_idShort"	= 743.8
Prop	"property_process_laser_power_idShort"	= 743.8
Prop	"property_process_nir_value_idShort"	= 743.8

Figure 34: Laser joining AAS submodel



DIMOFAC



The AAS submodel properties are initialized in one random value and when the platform is running the Python REST injects real-time machine data into the Intrasoftware AAS server.

4.5.5 Software

Regarding the software used, Python scripts are used for the Python HTTP server and Python REST client (see figure architecture). For the AAS model storage, the solution that has been developed in this project by Intrasoftware is being used. The Database is also provided by the Intrasoftware solution.

Regarding the analytics module that is being studied and developed we foresee the usage of Python.

4.6 MODULE 6 – Quality Control for fibre reinforced panels

4.6.1 Description

Within the scope of the EIRE pilot line (PL2), an entirely new module is being developed for the quality control (QC) of parts containing composite Carbon/Glass Fiber Reinforced Plastic panels (CFRP/GFRP), targeting mainly the aero-space industry. The quality control is done with non-destructing testing methods (NDT) and in this particular case with an Ultrasound Testing (UT) technique.

The main objective of this QC module is to inspect composite panels of relatively large dimensions (>30x30 cm²) in order to identify manufacturing defects. The focus here is to check the quality of bondings that are done with induction welding or similar methods. Future developments might include adaptations for inspecting metallic inserts in panels or multi-material panels.

4.6.2 Architecture

The module is composed of two assets, the UT device and a robotic arm. The first ensures the signal acquisition and the second ensures the movement of the transducer over a pre-defined path for the inspected sample. A central unit manages the inspection sequence and the connection to the DIMOFAC platform and to other modules. The global architecture is depicted in Figure 35 with the foreseen communication solutions.



DIMOFAC

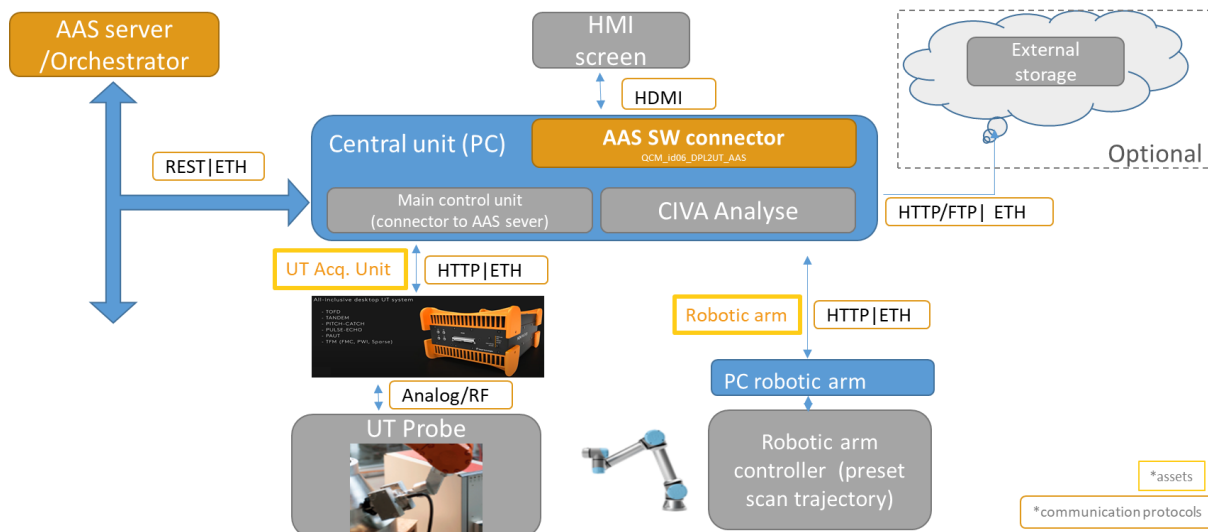


Figure 35: Architecture of the UT QC module (module 6)

4.6.3 Interfaces

The interfaces that will be used for communications between the different elements of the module are indicated in Figure 33. The central unit that will manage the main sequence will include a software connector to the AAS Server/Orchestrator, through a REST API mode. The UT acquisition unit and the controller of the robotic arm will both be connected through HTTP to the central unit. The UT acquisition unit has an analogic link the UT probe. An optional connection to external storage could be set also via HTTP.

4.6.4 AAS modelling

The AAS model of module 6 was defined using AASX Package Explorer. A set of submodels was defined, which is an extension of a previous work as an Excel table that was used by WP3 contributors.

A brief description of the submodels is provided in the following list:

1. Identification

This submodel contains identification information about the module, such as the assigned name, the manufacturer, the serial number and the manufacturing year, the model version and a generic module category.

2. Technical data

This submodel contains the information related to the technical specifications of the module, i.e. the dimensions, weight and acquisition mode

3. Operational data

The operational data includes the set of parameters used for the inspection process, such as the type of scan (e.g., TFM-PWI) and scan frequency (e.g., 10 MHz). The parameters for the robotic arm can be included too.



DIMOFAC



4. Input data

The input data contains information and parameters regarding the necessary inputs for the process in terms of batch or part identification and characteristics. This dataset is common for all usage modes.

5. Output data

The inspection process needs some processing steps and partial and intermediate data can be stored for as historical reference or for further use. Besides an identification of the part and batch, this submodel contains the information on the result type and a link to a folder where intermediate data is stored.

6. Output_QC_result

This submodel contains only the final result of the inspection process, as an acceptance criterion (*QC_result* : *OK*, *NOK* or *ToBeVerified*), a list with identified flaws, and depending on the scan type, a list with dimensional deviations and if created, a link to a file describing the deviation map.

The full definition as done with AASX Package Explorer is shown in Figure 34.

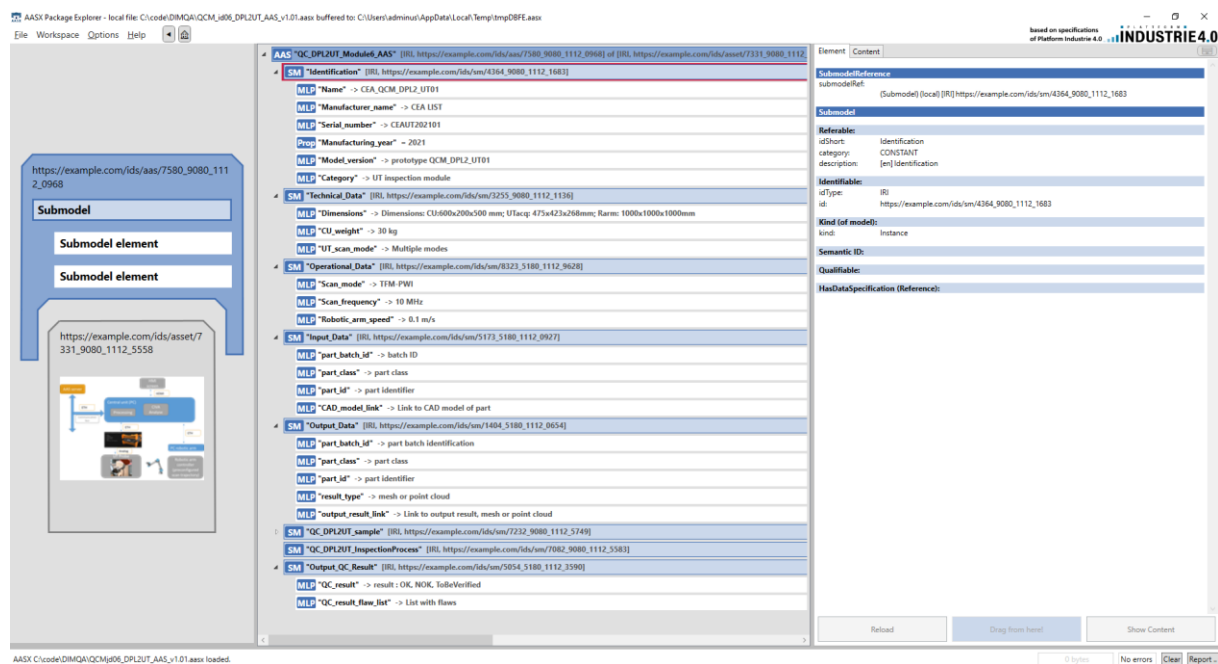


Figure 36: AAS model for UT QC module (module 6)

4.6.5 PnP capabilities

For the first version of the demonstrator module, the plug and produce capabilities will be reduced mainly to the use or not of the module. The acquisition parameters and the scan trajectory are preset for a given sample. A larger set or list with scan parameters and trajectories for several cases can be easily implemented. However, a



DIMOFAC



full and automatic reconfiguration for any type of sample would imply much more effort and it is out of scope of the planned work.

4.6.6 Software

A Python based graphical user interface (GUI) was implemented for the module management and usage. This application ensures the pre-configuration of the module in terms of AAS model and of the connection to the DIMOFAC platform. A second functionality is the start and stop of the acquisition and analysis processes, with the visualisation of the scan result in a separate panel. Depending on the usefulness, the result may also be presented under different forms. A screen copy is presented in Figure 37.

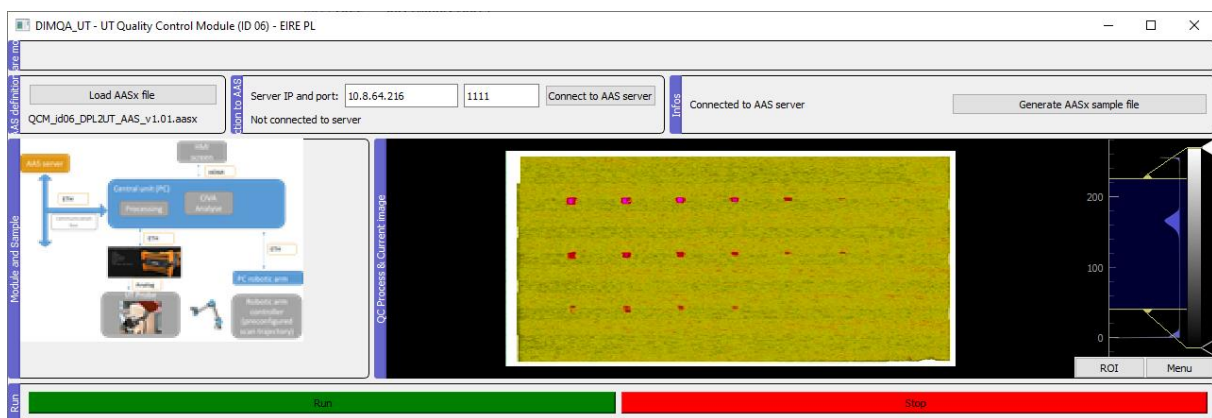


Figure 37: Main GUI of UT QC module (module 6)

Several pre-processing and post-processing steps are required for the signal analysis, and these will be done by a sequence of algorithms run in a batch mode or as external commands. The communication to the robotic arm controller will also be implemented as sequences of commands in batch scripts.

4.7 MODULE 7 – Small parts assembly

4.7.1 Description

The Small parts assembly specifics module ID 7 is being developed for Philips pilot line (PL3). It aims to design and implement a highly flexible and reconfigurable module able to handle the assembly of various small parts. The configuration of the module consists of 2 robotic manipulators, the needed fixtures (assembly table, jigs for each part to be assembled, etc.) and a transportation system (i.e., a conveyor belt) to move parts to and from the module. Setting and controlling of the assembly process are based on the exchange of data between the different Asset Administration Shells of the module's assets with the ROS environment and between the AASs themselves. In the module assets' list is also included a ROS-based DT (visualised in RVIZ) of the assembly cell and a multi-purpose vision system, used for multiple different part detection and quality assessment of the assembly outcome.



DIMOFAC



The Small parts assembly module aims to provide a resource agnostic, highly flexible module, both in hardware and in software layer, to enable rapid reconfiguration of the module, and eventually of the whole assembly line, to easily adapt to changes in production demand (fluctuations in production throughput demand, introduction of new products or new product variants, etc.).

4.7.2 Architecture

The hardware components of the module are:

1. The robotic manipulators (one UR10e and one Comau Racer7-14)
2. Custom designed robot tools that carry multiple grippers and are able to handle all the different parts
3. The 2D vision sensor (Basler camera)
4. The force-torque sensor (Onrobot HEX-H Force/Torque sensor)
5. The assembly table with specifically designed, easy to exchange, fixtures for each part to be assembled
6. The transportation system (Conveyor belt)
7. A Linux-PC that contains the DT of the module and it is used as the main control unit of the module.

Depending on the requirements of the part to be assembled, the robot tooling and the fixtures of the assembly table can be easily replaced to enable the module's fast reconfiguration. On that front, the fixtures, and tools for a second product (a monitor cover) have been designed and integrated to the module to demonstrate the module's fast reconfigurability and adaptability of the module.

The physical assets are integrated to the ROS framework to enable communication and data exchange between them. Figure 38 shows the architecture of this module.

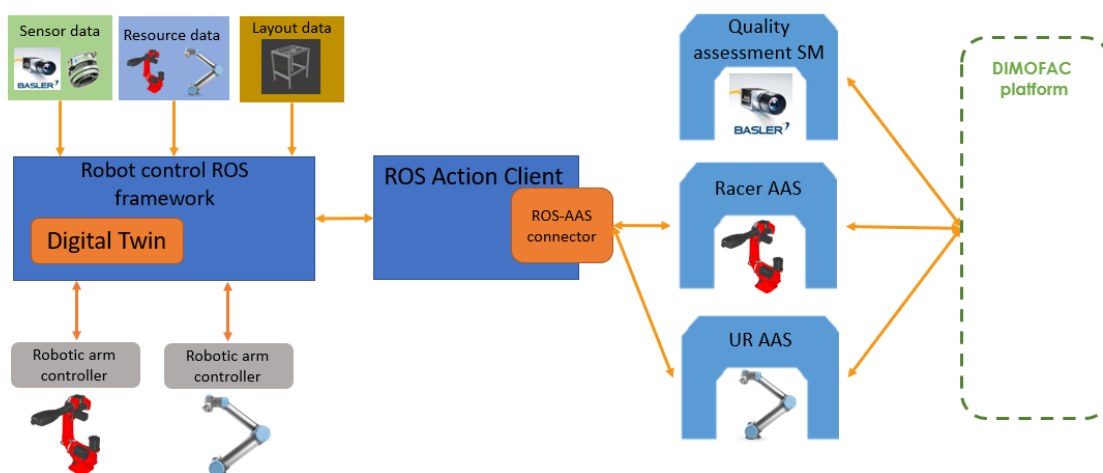


Figure 38: Architecture of Small parts assembly specifics Module



DIMOFAC



4.7.3 Interfaces

4.7.3.1 Physical assets – central unit

- **Ethernet TCP/IP**

Ethernet TCP/IP is used for the communication between the two robotic manipulators and the central unit. Specific ROS drivers are used to connect the physical robots with the ROS framework using their IP address and ports. Through these connections the robots expose their data to the ROS environment.

The same communication protocol is used to connect the camera and the force sensor control unit to the central unit. The data coming from the vision system for the parts detection and the force sensor are then also exposed to the ROS environment using ROS drivers.

- **ROS – AAS connection**

The main interfaces between physical assets and the DIMOFAC digital platform are custom ROS – AAS connectors. For the connectors between the AAS of the DIMOFAC platform, RESTFUL methods are used inside the ROS nodes that communicate with the assets that ROS is connected with. Specifically, the HTTP commands PUT and GET are used to update the AAS from ROS and to get information from the AAS to ROS respectively.

- **Connection of QA – AAS**

The implemented Quality Assessment (QA) system is connected to two different AASs. The first connection is with the UR robot AAS, using the modelled operation “quality check” (see Figure 38). This operation is used by the Process Orchestrator to trigger the quality assessment task execution. Also, a robot capability is connected to this operation to make the Task Planning aware that the UR robot is capable to perform a quality assessment task and therefore such a task can be assigned to this robot.

The second connection of the system with the AAS is regarding the reporting of the quality assessment results. The outcome of the algorithm is exposed to the ProducedPart AAS and more specifically is updating a field in the ProducedPart AAS which is related to the part quality control.

Both implemented QA – AAS interfaces are established with custom Python – AAS connectors by making use GET/PUT commands of the requests' library.

4.7.3.2 Central unit – DIMOFAC Platform

The communication of the information from the central unit of the module to the DIMOFAC platform is enabled thanks to RESTful webservices (REST API) commands and mainly through GET/PUT commands.



DIMOFAC



4.7.4 AAS modelling

At the current stage of development, the small parts assembly specifics module comprises 3 different AASs as listed below:

- Racer7_14 Robot AAS
- UR10e Cobot AAS
- Quality Assessment AAS

The building of the AASs and their respective submodels were based on the analysis of the use case and the identification of the data to be exchanged between all the assets. For the creation of the AAS models, the AASX Package Explorer was used. A brief description of the AAS models and their submodels are presented in detail below.

4.7.4.1 Robotic manipulators AASs

The AASX models for the two robotic manipulators have been built following the same approach. Static information regarding the robot model and the manufacturer are defined in the "Identification", "nameplate" and "technical_data" SMs. For each task to be executed by a robot, a cross check is performed between the capabilities SM of the robots and the required capabilities of a specific task to define whether the specific robot is capable to perform this task. A new configuration of the robot (e.g., the change of the tooling) is perceived as a change to the robot capabilities. Also, a property called "status" in the operational data SM is updated based on the real time status of the robot and is used to assign new tasks to the robot, if the robot is in idle state. Similarly, like a "malfunction" property is used to define whether the robot is in operation or malfunction state and thus able to perform tasks or not. Other capabilities of the robot, like its payload and type of movements that is capable to perform are also defined in specific SM in the AAS.

For each robot capability a specific SM has been created, where the related operations and the needed input parameters are modelled. These operations are used by the Process Orchestrator to trigger the execution of the robot task related to the specific capability by filling in the input parameters for the specific task.

4.7.4.2 Quality Assessment system AAS

As explained in 4.6, the Quality assessment system has no specific AAS model. It has been modelled as a capability of the UR robot inside the UR robot AAS and a specific operation is linked to this capability inside the same AAS model for the Process Orchestrator to trigger the execution of the quality assessment task from the UR robot. Hence, the quality assessment system has been modelled as an operation that the UR robot is capable to perform in a submodel embedded in the UR robot AAS. Also, the outcome of this task is exposed to the ProducedPart AAS using the quality control field existed in this AAS model.

4.7.5 PnP capabilities

The programming and control of the robotic manipulators are enabled through ROS (Robot Operating System). Due to its flexibility and open-source nature, ROS is widely supported by multiple robotic and industrial equipment manufacturers, including COMAU and Universal Robots. Thus, ROS, as well as the provided custom connection between ROS and the AAS, enables the capability to plug any ROS-compatible equipment to the module architecture without the need for major integration actions. This interoperability ROS brings will be validated and demonstrated as part of the PCL pilot in which Omron robots are used.

The PnP in software level is also enabled through the adoption of the CIM. A common AAS modelling approach is followed for all the robots. Thus, no major reconfiguration effort is needed to swift from one robot to another.

- **Physical interfaces**

The main physical interface of the robotic manipulators is the flange of the robot arms, through which the robots can mount different end-effectors. The flange represented in Figure 39 is a standard ISO 9409 - 1 - A 25.

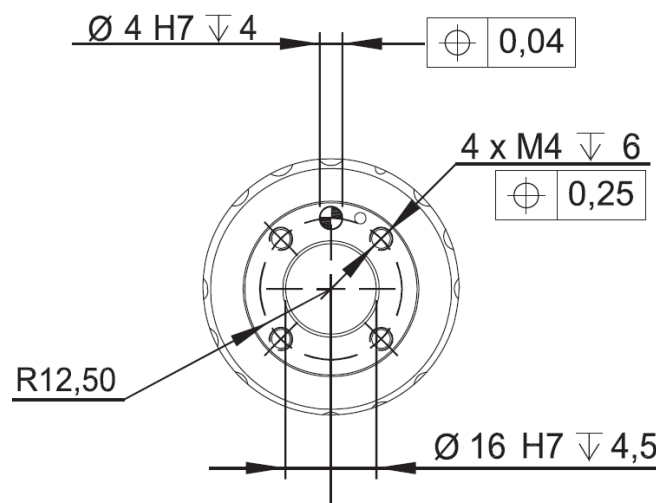


Figure 39: Robots' flange (ISO 9409 - 1 - A 25)

4.8 MODULE 8 – Automated Tape Laying

4.8.1 Description

The Automated Tape Laying (ATL) process is an automated technique used for composites manufacturing based on fiber placement processes. This module is part of AIMEN's Open Pilot Line focusing on manufacturing of multi-material components. With this process, wide tapes which may be composed of different materials based on carbon fiber are laid onto a part mold using a loaded roller system and a heating source to achieve the best compaction between layers. This module is composed by a movement system (robot) and heat system (ATL head), which can be composed by



DIMOFAC

IR system or laser source. The whole ATL module will be modelled, and it will physically be represented by the central unit (i.e., Jetson). The information gathered by the central unit (OT) will be available in the DIMOFAC platform (IT) as well as the product information related to the design and/or simulation (ET). In the central unit the software in charge of taking the decision and allowing PnP capabilities is named “Orchestrator”, and in the product side, the software in charge of register all the information related with a specific software “Digital Thread”. Figure 40 depicts the aforementioned considerations.

4.8.2 Architecture

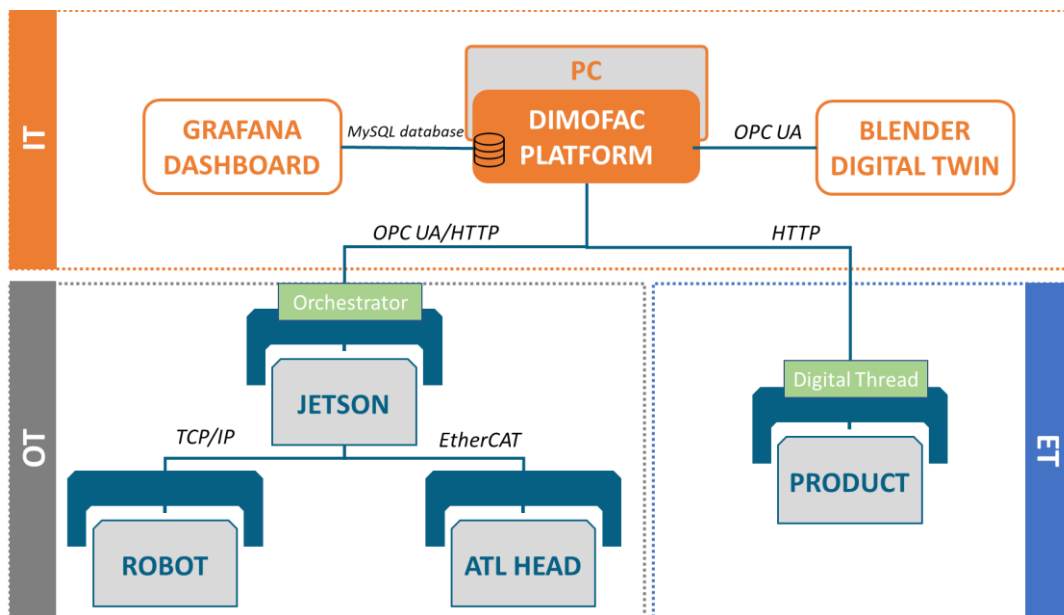


Figure 40: ATL module: architecture

4.8.3 Interfaces

4.8.3.1 Physical assets – central unit

- **Ethernet TCP/IP**

Ethernet TPC/IP is used to communicate the Fanuc robot with the Jetson device. The robot has a background task enabled which reads its variables and send them to the clients that are connected to its service, using web sockets.

The Jetson device will be connected to the robot using its IP address and port. The implementation of this communication will be specified in the D3.4 deliverable and described in the .aasx file.

- **EtherCAT**

The data of the ATL head is accessible through EtherCat protocol. The embedded PLC that contains all parameters of the ATL head is the Ethercat master of the network, where the Jetson device is defined as an EtherCat slave. Using this fieldbus, the Jeston



DIMOFAC



device can read and write all the ATL head parameters (described in the aasx file) needed to monitor and control the process.

The implementation of the EtherCat communication will be declared in the D3.4 deliverable and described in the .aasx file.

4.8.3.2 Central unit – DIMOFAC Platform

The first step to communicate with DIMOFAC platform is the register of the different assets. For doing this, it is necessary to export the AAS and the different Submodels (described in the next section) in JSON format. The JSON information will be registered using HTTP methods on the DIMOFAC Platform and will be necessary to describe what type of format is expected.

The DIMOFAC Platform not only allows us to communicate the module with the different assets using different communication protocols but also record the information using a MySQL Database. This ATL module will represent in real time some key information in a Grafana dashboard and the visualisation of the robot movements in Blender, as shown in Figure 41.

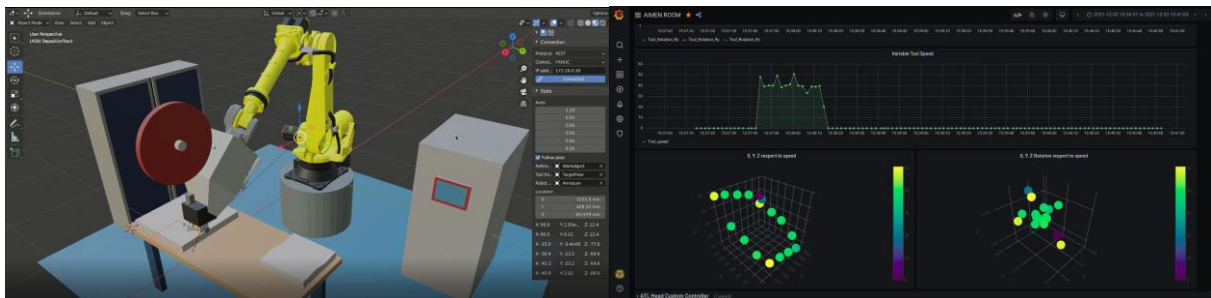


Figure 41: DT in Blender and Grafana dashboard

- **OPC UA**

The central unit/Jetson device publishes the parameters' values of the Operational Data and Parameters submodules of the physical assets (robot and ATL head) through OPC UA server.

- **REST API**

The central unit/Jetson device receives the information of the products that have been created through the digital thread application. This data is consulted to DIMOFAC platform by the Jetson device through a GET method in REST API.

When the process has finished, the Jetson device send to the platform the new information that have been created in the manufacturing process. This information is sent to the platform through a PUT method in REST API.

4.8.4 AAS modelling

The AAS modelling of the different assets follows the specifications described in T1.4 where the Common Information Model is defined. For this module, the assets needed for achieving PnP capabilities in ATL process are the following ones:



DIMOFAC

- Robot

This asset (Figure 42) has the two main capabilities (*move* and *communicate*). In this asset the key information is the position that will be send to the central unit to be synchronize with the process parameters.

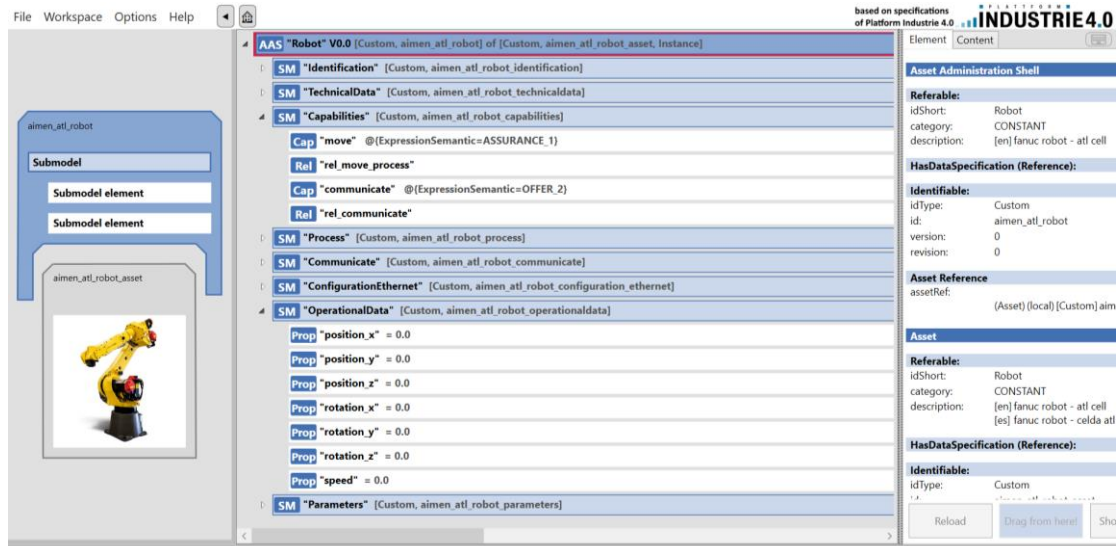


Figure 42: Robot AAS

- ATL head

This asset (Figure 43) is in charge of heating the material and it could be composed by (AasComposition submodel) two different assets: IR (AAS: ATLIR) or Laser source (AAS: ATLLaser). The submodel AasComposition will reference the asset used for the process (the next figure shows the example with IR). In the submodel Parameters only the common parameters are listed.

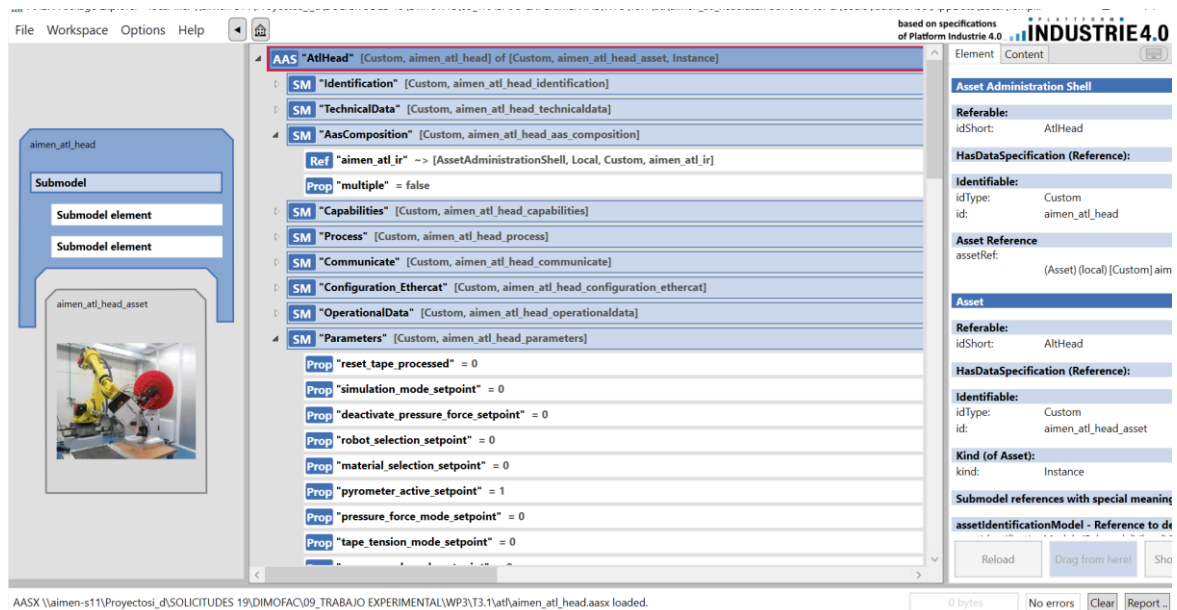


Figure 43: ATL Head AAS



DIMOFAC



- **Heat system**

In this module the heat system could be IR or Laser. The main capability of this asset is *heat* the material. The main difference between these two assets is the submodel *Parameters* and *OperationalData* shown in Figure 44.

AAS "AtIIR" [Custom, aimen_atl_ir] of [Custom, aimen_atl_ir_asset, Instance]	AAS "AtILaser" [Custom, aimen_atl_laser] of [Custom, aimen_atl_laser_asset, Instance]
SM "Identification" [Custom, aimen_atl_ir_identification]	SM "Identification" [Custom, aimen_atl_laser_identification]
SM "TechnicalData" [Custom, aimen_atl_ir_technicaldata]	SM "TechnicalData" [Custom, aimen_atl_laser_technicaldata]
SM "Capabilities" [Custom, aimen_atl_ir_capabilities]	SM "Capabilities" [Custom, aimen_atl_laser_capabilities]
SM "Process" [Custom, aimen_atl_ir_process]	SM "Process" [Custom, aimen_atl_laser_process]
Opr "heat"	Opr "heat"
SM "Communicate" [Custom, aimen_atl_ir_communicate]	SM "Communicate" [Custom, aimen_atl_laser_communicate]
SM "ConfigurationEthercat" [Custom, aimen_atl_ir_configuration_ethercat]	SM "ConfigurationEthercat" [Custom, aimen_atl_laser_configuration_ethercat]
SM "OperationalData" [Custom, aimen_atl_ir_operationaldata]	SM "OperationalData" [Custom, aimen_atl_laser_operationaldata]
Prop "angle" = 0.0	Prop "laser_heat_source" = 2
Prop "power" = 0.0	Prop "mode_control_setpoint" = 1
Prop "power_min" = 0.0	Prop "angle_setpoint" = 0.0
Prop "power_max" = 0.0	Prop "power_setpoint" = 0.0
SM "Parameters" [Custom, aimen_atl_ir_parameters]	Prop "min_power_setpoint" = 0.0
Prop "ir_heat_source" = 1	Prop "max_power_setpoint" = 0.0
Prop "angle_setpoint" = 0.0	Prop "threshold_setpoint" = 0.0
Prop "power_setpoint" = 0.0	Prop "fuzzy_out_min_setpoint" = 0.0
Prop "power_min_setpoint" = 0.0	Prop "fuzzy_out_max_setpoint" = 0.0
Prop "power_max_setpoint" = 0.0	Prop "fuzzy_dead_band_up_setpoint" = 0.0

Figure 44: Heat system (Laser and IR) AAS

- **Module**

This model represents the whole ATL module: what it is composed of *AasComposition* and the *Capabilities* submodels. In this case, the submodel *Capabilities* references to the process (*atl_manufacturing*) and communication capabilities (*monitor* data from the assets which compound the module and capability of *publish* data to be accessible in the Dimofac Platform).

- **ATL Process and Product**

These assets are used for linking the information gathered in the shopfloor with the data coming from other stages (i.e. design, simulation, quality). The main submodel of the *Product* is the *Lifecycle* which will reference the module used for manufacturing the part/product as well as the information coming from simulation.

4.8.5 PnP capabilities

The PnP capabilities for this module are achieved by following a CIM. The central unit is able to decide what physical assets are available and to start the communication (Hardware PnP capability), and it is also able to decide the assets/modules needed for manufacturing a specific product (Software PnP capability).

For hardware PnP capability, the software ('Orchestrator') will try to stablish the communication with the module and physical assets that compound this module. So, for doing that, the software will take the configuration specified in the Submodel Communication and execute the corresponding method with the attributes specified



DIMOFAC



in the Submodel. If the asset is not physically connected to the central unit, the software will not allow start the recording of the process.

For software PnP capabilities, the software ('Orchestrator') will take the decision of what module is needed for manufacturing a specific product. The AAS of the product will reference the processes that need to be performed. Then, the software will search the process AAS, which specifies the capabilities needed for executing the process, and the module that can perform these specific capabilities. Therefore, as the software is based on the Capabilities Submodel, the user does not need any programming task.

4.8.6 Software

As mentioned before, this module will be managed by using two main software: the software for creating and managing Product AAS information and the software for orchestrating the physical assets and monitoring the process.

The sequence and iteration that the user has to do with the different software shown is the following one: the user will start with the creation of an asset of the product using the Digital Thread application. Then, for starting the manufacturing, the user will select from a list the product to be manufactured and the module available for this manufacturing. During manufacturing, the user can see the status of the process using the DT in Blender, the dashboard created in Grafana and/or the Orchestrator with the real time information of the process parameters. Once the manufacturing would have been finished, the user could check that the information related to the manufacturing is automatically linked with the product previously selected.

4.9 MODULE 9 – AGV-Robot

4.9.1 Description

The Comau mobile platform (Figure 45) is a novel mobile manipulator that is being prototyped by COMAU. It is the result of the integration of two robotic resources made by COMAU itself, the Agile 1500 AGV and the Racer5 Cobot.

The main advantage of this manipulator comes from the possibility of handling and transporting a wide variety of objects extremely accurately and without any additional infrastructure required.



DIMOFAC

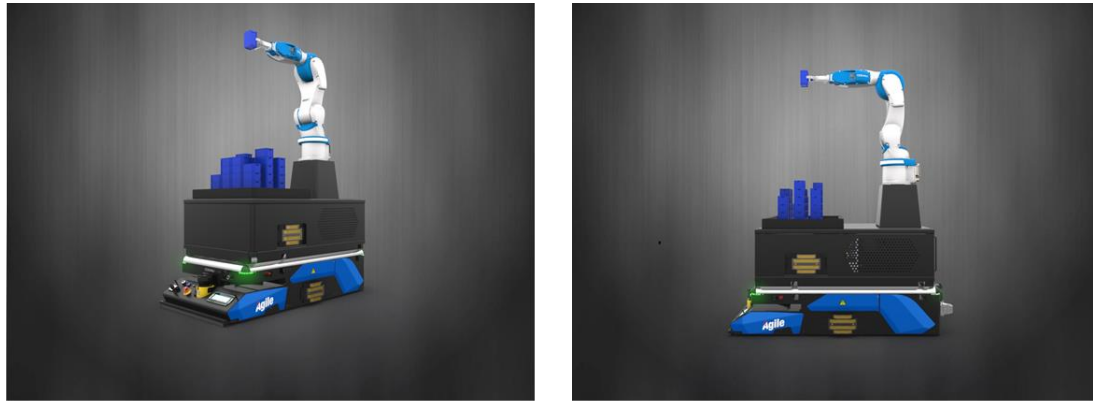


Figure 45 : Comau mobile platform

The mobile platform will be used in the VDL pilot line for robotized machine tending (Figure 45). The process can be broken down into the following three tasks:

1. Pick and place of raw materials
2. Load and Unload of milling machines
3. Mobile platform navigation

4.9.2 Architecture and interfaces

Comau robots are programmed in PDL2, a proprietary programming language. A viable option to interface the robotic resource with the external world is the Socket use via PDL2. The Ethernet port may be used to transfer data through a socket with TCP or UDP protocol using the instructions available in PDL2.

The choice between TCP and UDP protocol depends on the application requirements:

- **TCP:**

Data are transferred in a not predictable time, but it is guaranteed they will reach their destination. With this protocol, the communication takes place between two entities that have the respective roles of TCP Client (asking the server for a service) and TCP Server (providing the asked service to the client).

- **UDP:**

Data are transferred quickly, but it is not possible to guarantee they reach the destination. Also, in this protocol, the communication takes place between two entities that have the respective roles of UDP Client (asking the server for a service) and UDP Server (providing the asked service to the client).

Examples on the programming of this communication can be found in the manual "PDL2 Programming Language - System Software C5GPlus - R1C" which is provided by Comau.



A more organic way of controlling the robot is through ROS (Robot Operating System). Due to its flexibility and open-source nature, ROS is commonly used in innovative projects, and it is fully supported by Comau robots. In brief, ROS is a flexible framework for developing and integrating robot related applications. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

ROS communicates with the robot via TCP-IP communication protocol managed by the Comau-ROS driver. This package monitors both the robot status in terms of pose and the task status, in order to maintain an updated knowledge of the system state. Moreover, it waits for new motion commands coming from the Motion Handler and passes them to the robot.

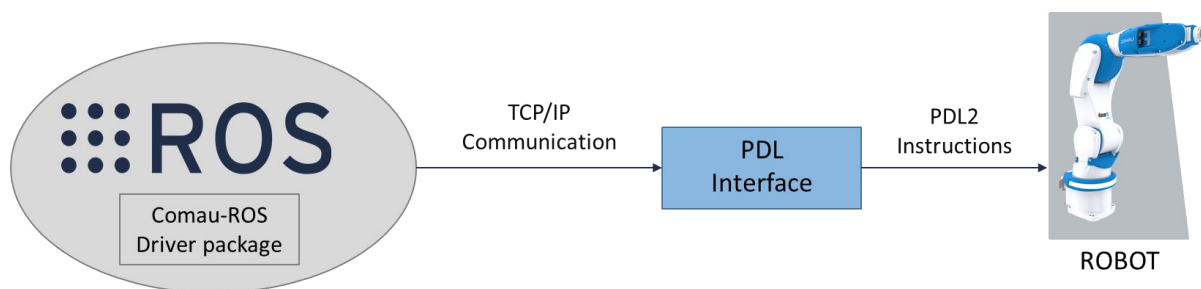


Figure 46. ROS communication scheme

The possibility of using ROS with the Agile1500 is present, although it is not yet fully exploited. In fact, an extensive programming effort is needed in order to create the ROS driver for the AGV.

4.9.3 AAS modelling and Software

This information will be described in section 4.17, within the real application of the COMAU Robot.

4.10 MODULE 10 – Milling

4.10.1 Description

The milling module developed in DIMOFAC (ID number: 10), will be part of the VDL pilot line, and the working group around it, is VDL, COMAU and LMS. The developments of this module focus on:

- Enabling the digitalization of the milling process and the milling machines.
- Enabling the monitoring and control both of milling process and the machine.
- Increasing process responsiveness and flexibility.

For the monitoring and control, the chatter is investigated throughout this module, while the workflow followed provides useful information for monitoring and controlling other milling phenomena. The architecture of the developments bases on being adaptable to different machines and different manufacturing scenarios.

4.10.1.1 Architecture

In Figure 47 the architecture of the milling module is presented:

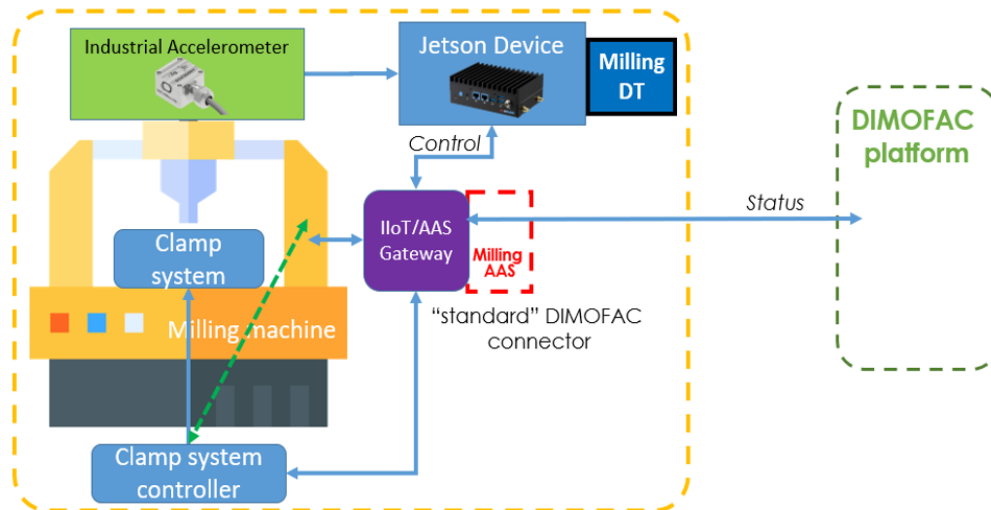


Figure 47: Milling Module (ID: 10) Architecture

The hardware components of the module are:

1. Milling Machine:

Two milling machines are investigated in this phase:

- I. DMU 80T 3-axis horizontal milling machine, equipped with the Heidenhain TNC 430 controller.
- II. ACCUWAY UM 110H 3-axis horizontal milling machine, equipped with the Heidenhain iTNC530 controller.

2. Clamping system and clamping system controller:

For automating the load and unload of parts that are to be milled or milled, on the machine an automated clamping system is developed, consisting of the actual clamping system, as also the external clamping system controller.

3. Accelerometer:

- o Kistler 8762A10 tri-axial accelerometer

Specification	Value
Acceleration range [g]	± 10
Acceleration limit [g]	± 16
Sensitivity, $\pm 5\%$ [mV/g]	500
Resonant frequency [kHz]	30
Frequency Response, $\pm 5\%$ [kHz]	0.5 ... 6000



Specification	Value
Amplitude non-linearity [%FSO]	± 1
Sensing element	Ceramic shear
Mounting (thread)	M6
Case Material	Durable hard anodized, ground isolated aluminum housing
Ingress Protection Type	IP66

Table 3: Specifications of accelerometer Kistler.

- Micromega Dynamics IAC-CM-U-03 3-axis accelerometer

Specification	Value
Acceleration range [g]	± 25
Acceleration limit [g]	± 1000
Sensitivity, $\pm 2 \dots 5\%$ [mV/g]	80
Resonant frequency [kHz]	Not Specified
Frequency Response [kHz]	0 ... 10000
Amplitude non-linearity [%FS]	$\pm 0.5 \dots 1.5$
Sensing element	MEMS
Mounting	3.2mm diameter holes (4x)
Case Material	Stainless Steel Casing and Cable Gland A4 / AISI316 Grade
Ingress protection grade	IP67

Table 4: Specifications of accelerometer Micromega.



4. DAQ system:

For the acquisition of the chatter related signals from the accelerometer, a Labjack T7-Pro is used, which specifications can be found below in Table 5, and detailed information in: <https://labjack.com/products/t7>

Specification	Value
Analog Input	14 Analog Inputs 24-bit low-speed ADC Software Configurable Resolution Settings Single-Ended Inputs (14) or Differential Inputs (7) Analog input ranges: $\pm 10V$, $\pm 1V$, $\pm 0.1V$ and $\pm 0.01V$ Sampling up to 100kS/s
Analog Output	2 analog outputs (12-bit, 0-5V)
Communication Options	USB Ethernet 802.11b/g WiFi
Other Highlights	Capable of stand-alone operation by running Lua Scripts Field Upgradable Firmware Industrial temperature range (-40 to +85C)

Table 5: Specifications of Labjack T7-Pro

5. Industrial Computer:

For the operation of the Milling DT and the Milling Chatter Monitoring and Control system, a computer that can be equipped with python and the respective libraries can be used. Following the needed specifications of the system, two different options are provided:

- I. Dell Latitude 5501 laptop, which specifications can be found below in Table 6:



Specification	Value
CPU	Intel Core i7-9850H with 2.6 GHz clock speed, 6 Cores and 12 Logical Processors
RAM	16 GB
Operating System	Windows 10 Pro
Hard Drive	Western Digital SN730 NVMe Solid-State Drive

Table 6: Specifications of Dell Latitude 5501 laptop

- II. AD-Link DLAP-401-Xavier, which specification can be found below in Table 7, and detailed information in:

https://www.adlinktech.com/Products/Deep_Learning_Accelerator_Platform_and_Server/Inference_Platform/DLAP-401-Xavier

Specification	Value
CPU	NVIDIA® Jetson AGX Xavier™
RAM	32 GB
Operating System	Ubuntu 16.04
Hard Drive	eMMC 32 GB

Table 7: Specifications of AD-Link DLAP-401-Xavier

6. IIoT/AAS Gateway:

A gateway will be used between the different systems for enabling the communication between the different assets. The model of it is Kunbus RevPi Connect+, which specifications can be found below in Table 8:

Specification	Value
CPU	Broadcom BCM2837B0 quad-core ARM Cortex A53 @1.2 GHz
RAM	1 GB
Operating System	Ubuntu 16.04
Hard Drive	eMMC 32 GB



DIMOFAC



Table 8: IIoT Gateway Specifications

The software components of the module are:

1. Milling DT:

A DT of the Milling Process enabling the digitalization of the process and machine related information.

2. Chatter Monitoring and Control System:

A software implemented in python, responsible for detecting the milling chatter, and suppressing it.

4.10.2 Interfaces

4.10.2.1 Interfaces Physical Assets-Central Unit

The physical assets are connected via ethernet, using the Ethernet protocol, except for the accelerometer where information is sent via voltage signals, converted on the DAQ to accelerometer signals, through an analogue to digital converter and communicated to the industrial computer through the Modbus protocol and USB2.0 hardware connection.

4.10.2.2 Central Unit-DIMOFAC platform

The communication of the information from the central unit of the module to the DIMOFAC platform is enabled thanks to RESTful webservices (REST API) commands and mainly through GET/PUT commands.

4.10.3 AAS modelling

The milling machine AAS model, has been developed with regards of the guidelines provided by DIMOFAC Task T1.4 Common Information model, with additions specified to the respective needs. More specifically, the implemented submodels are listed below:

1. Identification submodel:

On the identification submodel, information regarding the machine ID, such as the model, manufacture, identification factory number, etc. are stated.

2. Technical Data submodel:

On the technical data submodel, several Submodel collections are defined, grouped in general technical data regarding the machine, specific data per axis of machining and data for the tooling used.

3. Documentation submodel:

Here documentation related data, such as machine manuals etc., are modelled (e.g., a URI on user manual, opens a pdf file of it where you can download it, etc.)



DIMOFAC



4. Operational data submodel:

In this submodel, data regarding the operation of the milling are defined, such as feed rate and spindle RPM during machining, or accelerometer signals, the status of the clamping device (clamps on/off), etc.

5. Input data submodel:

In this submodel, the parameters on the machine regarding feed rate and spindle RPM override change requests from the chatter monitoring and control system, as also the gcode file (file with the trajectories for the milling process) to be loaded to the machine are modelled.

6. Capabilities submodel:

The capabilities of the Milling are modelled in this submodel. Due to modelling reasons, each capability is modelled separately as a submodel, and then imported in the submodel capabilities.

4.10.4 PnP capabilities

The developments around the module, have been designed in a way to enable the plug and produce adaptation, in the case that the communication of the controller is enabled, and the data needed are available. Other than that, the system, from a hardware and software perspective, as also in the case of algorithms, can run and perform regardless of the machine, controller type with minor modifications, such as changes on where to get/put data from the machine controller to the AAS, and sensor placement on the machine.

4.10.5 Software

The software developed for this module consists of two components, as also stated above:

1. The Milling DT:

that focuses on enabling the digitalization of process and machine, through extruding and providing operators and engineers with data and generated knowledge.

2. Chatter Monitoring and Control System:

algorithm that detects and suppresses chatter, regardless of the manufacturing scenario, adaptable to reconfiguration.

Both software components are implemented in a self-developed python program and visualized through self-developed GUIs.

4.11 MODULE 11 – Laser quality control / sheet-metal bending and additive manufacturing

4.11.1 Description

Laser Quality Control module (ID number: 11) is focused on the dimensional quality assessment of manufactured parts of a production line through 3D laser scanner techniques and advanced metrology software. In DIMOFAC the module is applied for 2 pilots' lines: VDL (for manufactured sheet metal parts) and SCL (for 3D printed parts). The main purpose is to verify that the final product is in concordance with its design specifications and tolerances and/or identify possible deviations through the capture, processing and analysis of point clouds that represent a DT of the part under assessment. In the case of SCL, for example, the plastic parts (polymers) are produced by batches of several samples. Individual parts of the batch will be selected for the QC analysis. For the project a specific part model is under analysis (it can be visualized in Figure 48).

4.11.2 Architecture

In Figure 48 is presented a high-level reference architecture of the laser QC module and its interaction with other components.

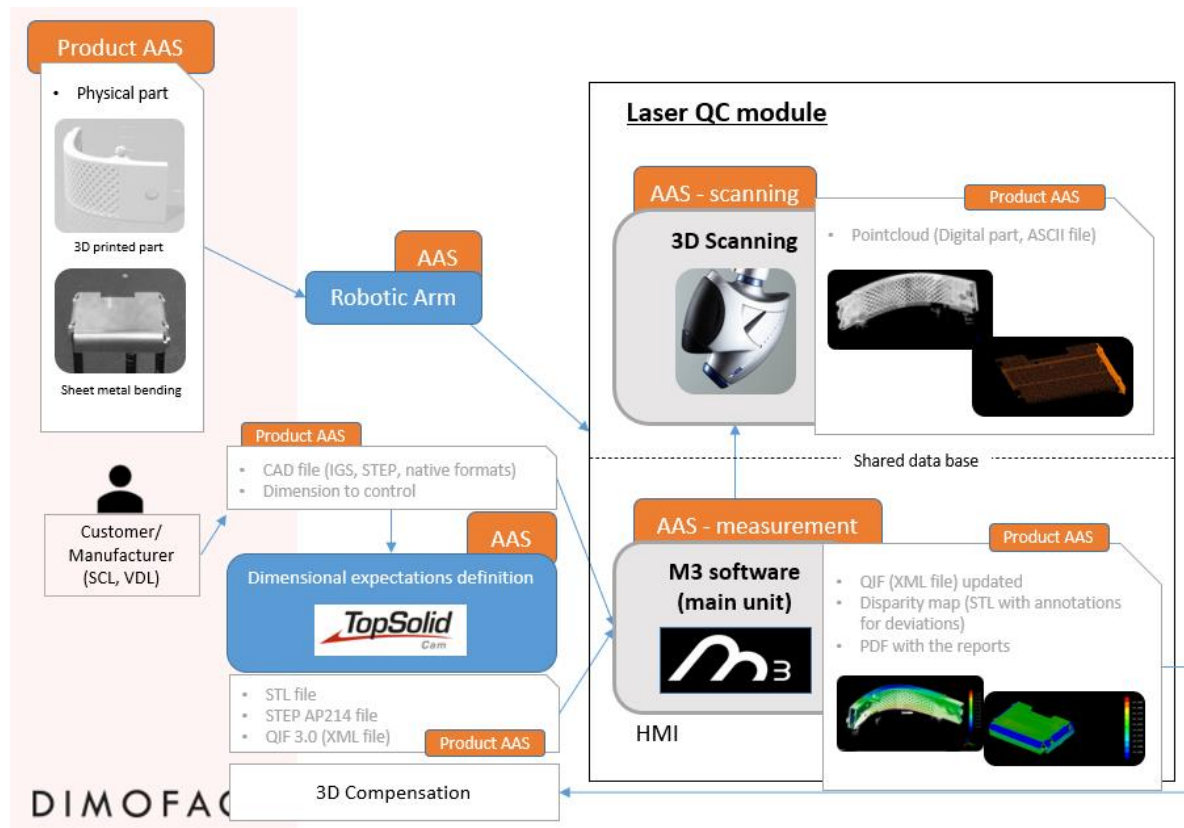


Figure 48: Reference Architecture of QC module 11 and its main interaction with other modules and components

As can be stated, the module is composed by two main components (3D scanner + metrology software).



DIMOFAC



The **scanner** is an optical sensor for high-speed scanning and capture of 3D point clouds formed by the laser, electronics and optical technology to allow synchronization with the mover (that can be either a CMM, CNC machine, robot or portable arm) and provide better accuracy and repeatability in the different measurements. With the 3D scanner is possible to inspect the quality of products by obtaining a digital copy that can serve for comparisons against CAD dimensional information for the product. Following PnP principle, any laser scanner can be used in this module if it complies with the properties of field of view, lateral resolution, accuracy, acquisition rate, etc., thanks to the high compatibility that the advanced **metrology software** facilitates. In this way, the physical parts (in the case from VDL and SCL pilots) once manufactured are correctly positioned in order to be scanned by the device.

The **metrology software**, in this case, M3 software, is a high-performance software for capturing and analysing point clouds that covers the entire spectrum of metrology, regardless of device, brand or model and works with the widest range of geometries and geometric dimensioning and tolerancing (GD&T). M3 receives the point clouds coming from the sensor and processes them to generate a digital copy of the product and to analyse the relevant dimensions and geometries comparing the information with the PMI of the part within the CAD model or QIF data that is also an input for the M3 Software. After this process, valuable information is generated that indicates if the part has passed or failed the quality control inspection. M3 has a simple, powerful and agile interface.

The important components within the quality inspection process will be linked with the AAS server, in this case are the manufactured product and the robotic arm (external to the module) along with the scanner and the M3 software processing and generating the quality outputs (central unit of the module).

4.11.3 Interfaces

The physical asset (scanner device) is connected to the central unit M3 by proprietary protocol based on TCP/IP. This interface provides fast and reliable connection to obtain the point cloud from our OptiScan scanner. On the other side, our main software M3 (Central unit of the module) will be connected to the DIMOFAC platform by REST API or MQTT protocol. The DIMOFAC platform will store the AAS and will launch the scan process via operation of the AAS. Figure 49 show the dataflow between the different assets.

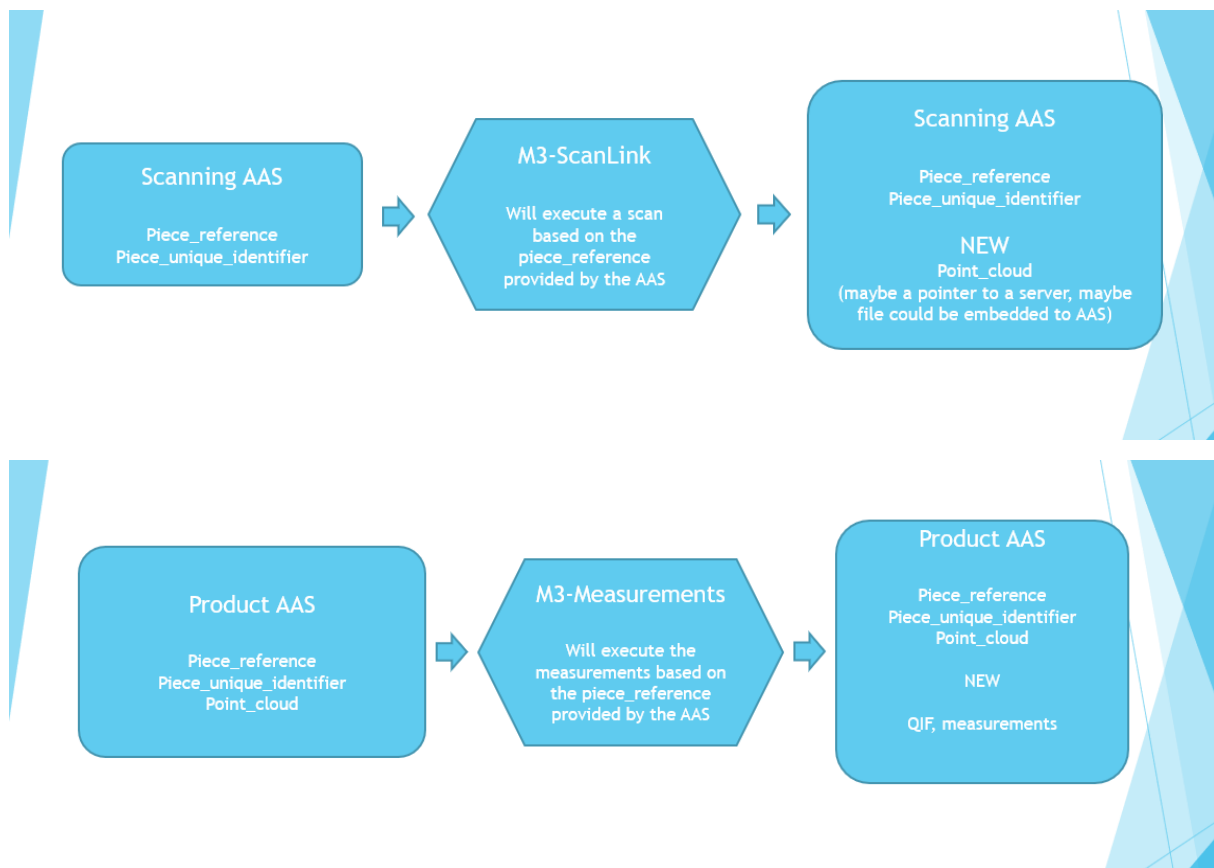


Figure 49: Reference flow diagram of Laser QC module.

4.11.4 AAS modelling

DPX has two different working lines. First of all, an object is measured by our scanning software, which will create a point cloud. Afterwards, the point cloud will be measured and a colour mapping will be created, indicating the quality of the piece.

For this purpose, two assets have been created, modelling our scanner and measurement software. This have been made with AASX Package Explorer.

The first asset, called "dpx_scanner_aas" (Figure 50) will manage the M3 scanning software and all parameters needed for a successful scan. In our case, we have information about the scanner and just some properties that will be updated in future. For scanning purposes, we have another section with operational data, called "scan_program" that will face with all parameters needed for scanning, with different modes, like slow/fast and so on. Another section is used to know which piece is going to be scanned and its serial number, so it can be tracked in every step. At the end, a point cloud will be created and embedded. There is another section "triggering" in which we could see two different types of triggering the scanner. It could be done by a flag or by an operation, will be discussed in T3.5 which one will be implemented.



DIMOFAC

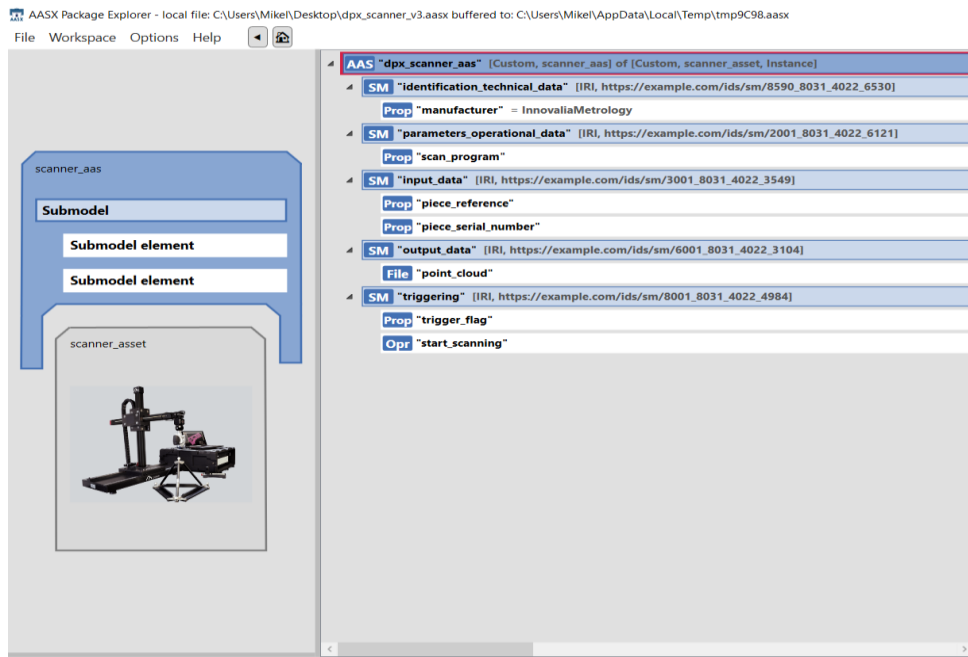


Figure 50: Scanner AAS.

The second asset (Figure 51) will measure the point cloud and as an output will create a QIF file and a colour mapping pattern. The scheme is similar to the previous one. Information about technical data and parameters, in this example “measurements program”, which will be made for each piece. The information this asset will receive is all about “input data”, that contains reference of the piece, serial number and the point cloud record by the previous operation. All this information will generate output data, as previously said will be a QIF file and a pattern.

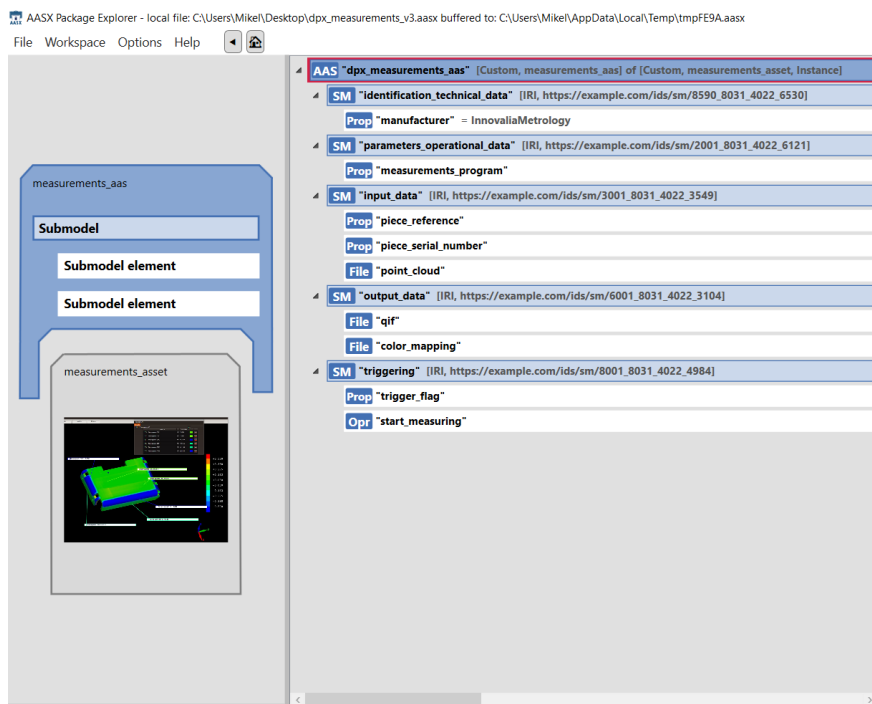


Figure 51: Measurement AAS.



DIMOFAC



4.11.5 PnP capabilities

DPX 3D scanner and metrology software have been based on the principle of plug-and-produce since its conception. The complete system in the module can perform its work regardless of the hardware used for the obtention of the point clouds of the component. In fact, for the implementation of the module in the SCL pilot is being considered to execute the M3 software analysis to a point cloud mesh delivered by module 18 that is an X-ray QC module. On the other hand, for the VDL pilot at this point it is defined to use a 3D scanner solution by DPX. Based on this it can be showed the PnP capabilities of the module that allows fast adaptation of the different scanning tools along with multiple configurations and adaptations to the Internet of Things, different types of manufactured parts and different types of production lines. The communication is based on the well-known ethernet and TCP/IP protocol.

4.11.6 Software

The central unit of the module is the M3 software. In this sense, the communication flow goes as follow:

Inputs for M3:

- Raw point clouds ASCII file XYZ coming from the sensor (3D scanner or Module 18)
- PMI (Product manufacturing information) of the product within CAD file and/or QIF

Outputs from M3:

- Processed point cloud mesh in txt format (This can come from an external component or module as well, as have been described before)
- Colour deviation map in STL format + annotations
- QIF XML report with inspection results

As commented in previous sections a Scanning AAS have been set and it will contain, in a first phase, a part reference and unique identifier. The scanner will start the scanning process by a triggering coming from the AAS and will fulfil the Scanning AAS with the point cloud of the scanned part (it is under assessment whether this will be in the form of a pointer to a server or if it could be embedded to AAS). Then, all available data about the part is received by the Measurement AAS that is referred to M3 software where the dimensional analysis is done and a QIF with the results is generated.

In Figure 52 is shown how it is visualized in our software the deviation colour map from one of the SCL sample parts scanned.

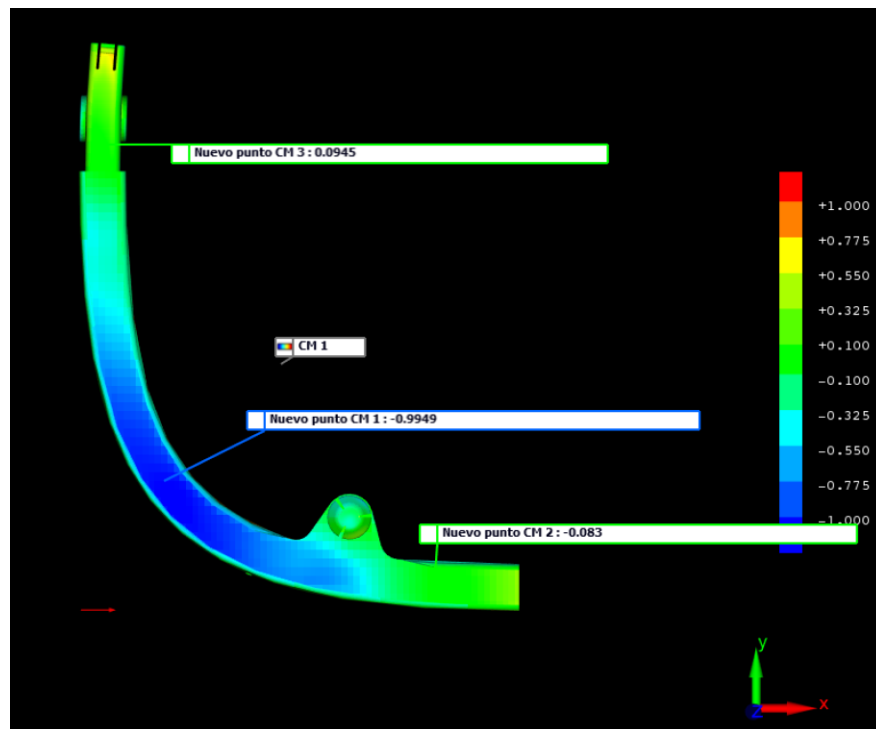


Figure 52: Colour deviation map SCL sample part.

4.12 MODULE 12 – AR assembly instructions

4.12.1 Description

The AR assembly instructions module which will be applied to VDL's use case, will provide support to the human operators in the shopfloor. It is a multimodal application providing assembly, inspection, reconfiguration and recovery instructions and resource status visualization (Figure 53). Also, the developed AR-based operator support system is integrated to the overall system via the AAS which will enable the seamless communication between different components increasing the reconfigurability of the production system.

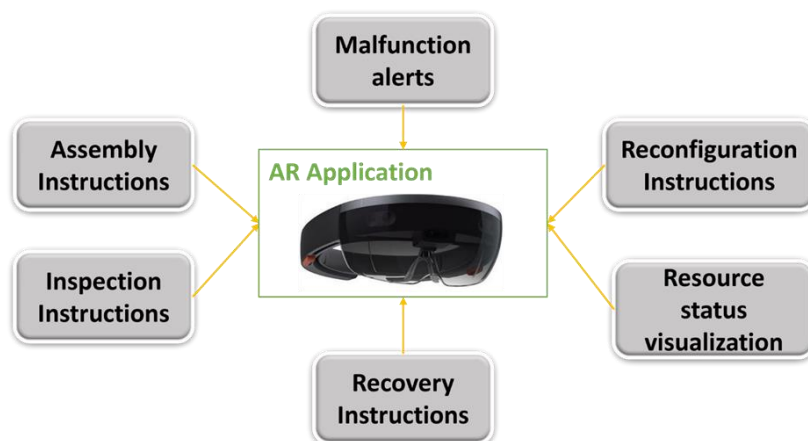


Figure 53: AR application functionalities

4.12.2 Architecture

In Figure 54, the architecture of the system is shown, presenting the connections between the various components for the AR application in the shopfloor.

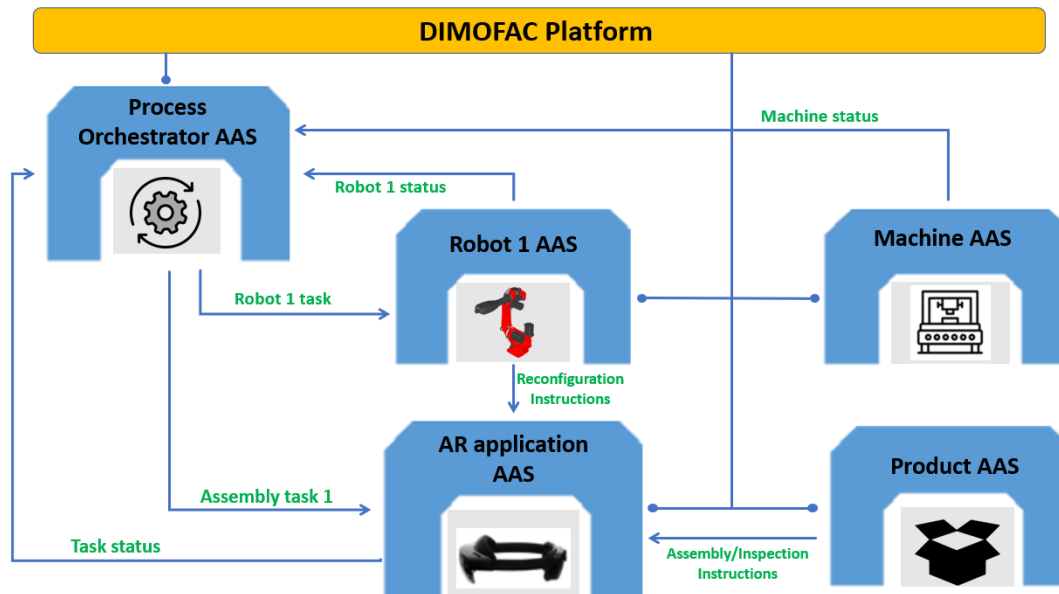


Figure 54: Overall system architecture

The AR application in the HoloLens2 glasses is connected to the AASs of the entities from which data input is needed. The assembly instructions are received from the AAS of the product to be assembled, while the recovery instructions and the reconfiguration instructions that have to do with the robot, as well as the status of the robot for visualization in AR, are all retrieved from the robot's AAS. In general, all status visualization inputs of resources are received from the respective AAS of each resource. The AR application itself sends to its AAS the status of the task that is currently executed by the human operator (pending, ongoing, completed, rejected, etc.). The AR application is built in Unity3D which utilizes a C# scripting API, so the exchange of data between the application and the various AASs is achieved with RESTful services, HTTP methods such as PUT, GET, etc.

4.12.3 Interfaces

4.12.3.1 Physical asset – Central unit

The HoloLens2 device where the AR application is deployed is not connected to the central unit during the execution of the application. However, before the execution, a connection between the HoloLens and the central unit needs to be established for the developed application to be deployed to the AR Glasses. This connection can be performed either with a USB-C cable or via Wi-Fi.



4.12.3.2 Physical asset – DIMOFAC platform

For the communication with the AAS of the DIMOFAC platform, RESTful methods are used. Specifically, the HTTP methods PUT and GET are used. PUT is used to update the fields of AASs of other assets, for example, update the AAS of the product with the result of the inspection instructions, or the state of the human in the AAS of the AR application. GET is used to receive elements from other AASs, such as the assembly instructions that are received from the product AAS.

4.12.4 AAS modelling

For the AR Application, a specific AAS model has been created to represent all the relevant information and the functionalities of the application into specific submodels. These submodels of the AAS of the AR application can be seen in Figure 55.

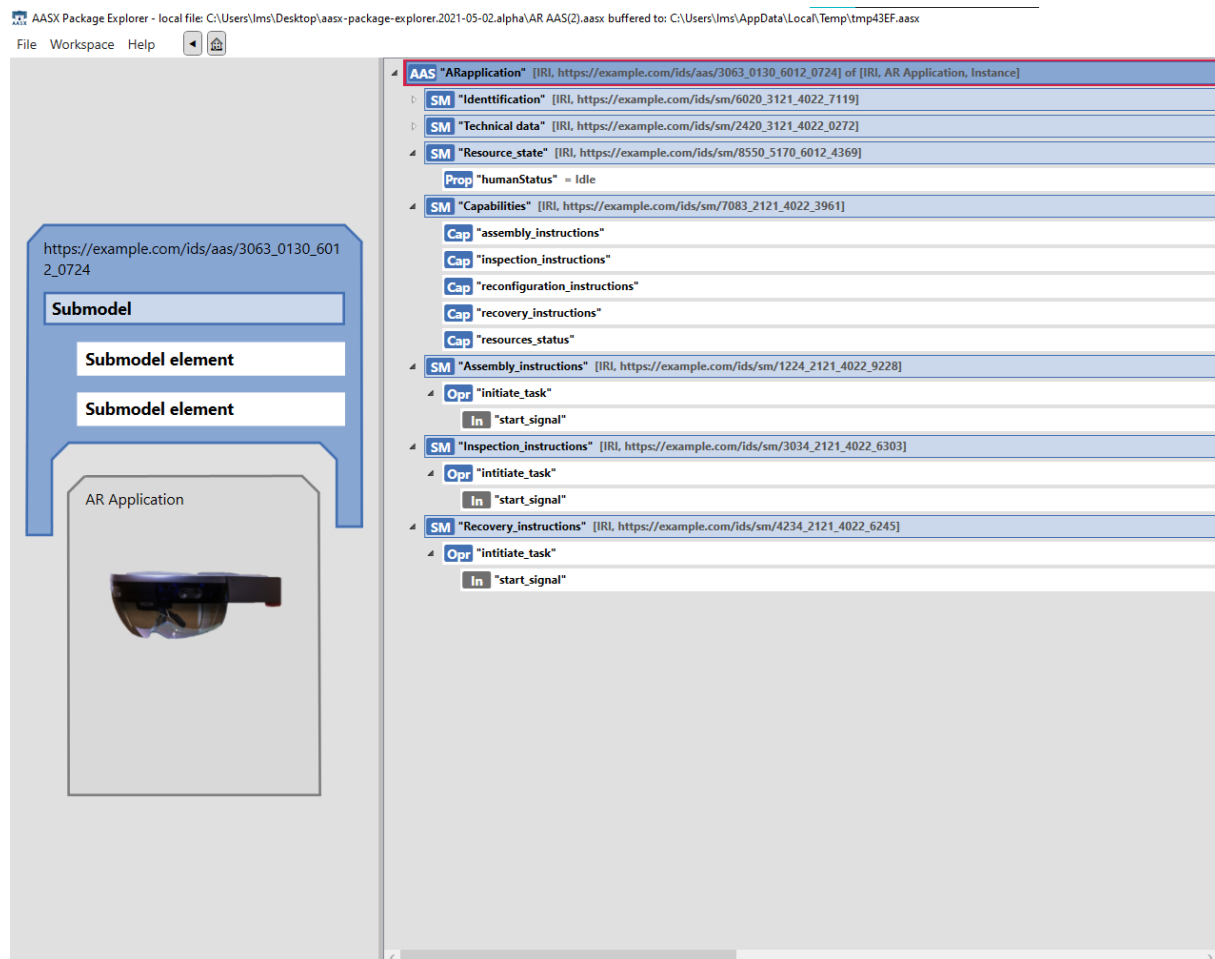


Figure 55: AR application AAS

The submodels are as follows:

- Identification contains the identification information for the application.
- Technical data contains the technical data of the Hololens2 headset.
- Resource_state contains the resource state of the human, updated from the AR applications



DIMOFAC

- Capabilities contains the capabilities of the application, namely the assembly, inspection, reconfiguration, and recovery instructions, as well as the resource status visualization
- The Assembly_instructions, Inspection_instructions and Recovery_instructions are submodels that contain the operations that will be used by the orchestrator to initiate the respective capabilities.

4.12.5 PnP capabilities

The AR application works with instructions in JavaScript Object Notation (JSON) format, and thus able to work with any instructions derived from AAS submodels, that contain text, images, and 3D models with minimum configuration. The resource status visualization receives the resource status from the AAS of the resources, and as such, can work with any kind of machine and robot that has a connection to the central platform AAS. With these advantages, the AR application can be attached to multiple pilots with minimum configuration.

4.12.6 Software

The software that is used for the development of the AR application is Unity3D, the graphical interface of which is shown in Figure 56.

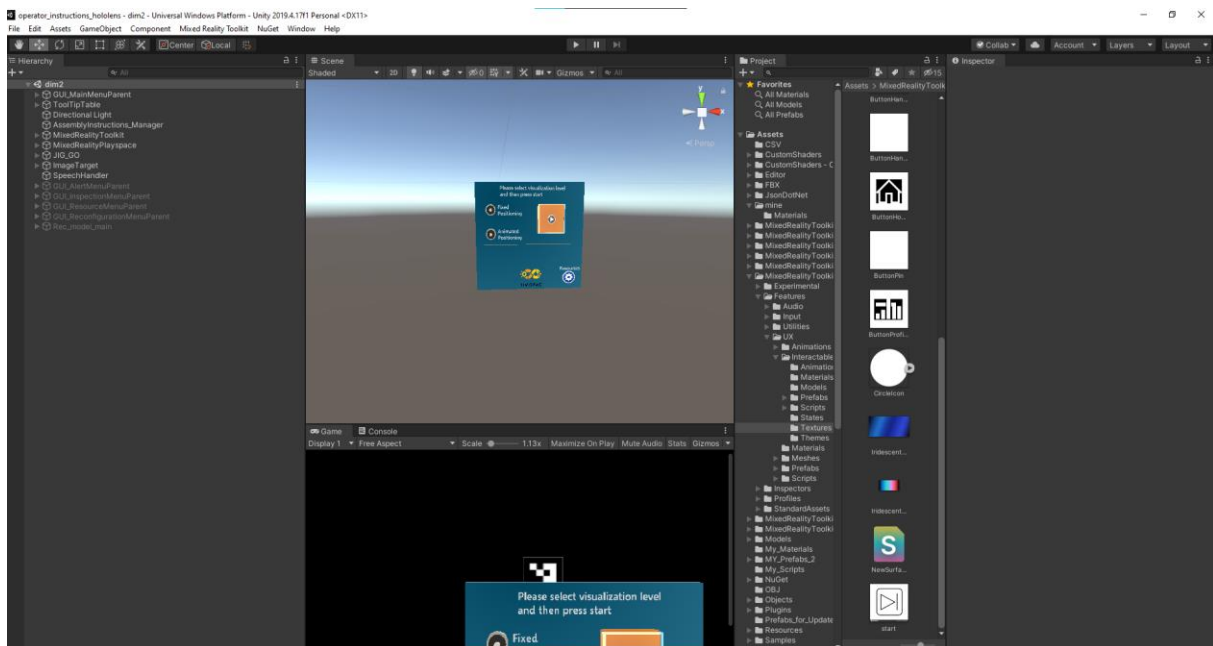


Figure 56: Unity3D graphical interface

The framework works with graphical objects as well as scripts written in C# language. Unity3D is a cross-platform engine, is supported on several operating systems, such as Android and Windows, and is widely used in AR and VR applications

4.13 MODULE 13 – Laser welding

4.13.1 Description

The laser welding module developed in DIMOFAC, will be demonstrated through the VDL pilot line. The developments on this module focus on the implementation of a vision-based quality control system that will be responsible to identify any gaps existing between assembled parts before the initialisation of the welding process. Also, simulation tools have been implemented for the simulation of the welding procedure, to assist during a reconfiguration scenario.

4.13.2 Architecture

In Figure 57 the architecture of the laser welding module is presented.

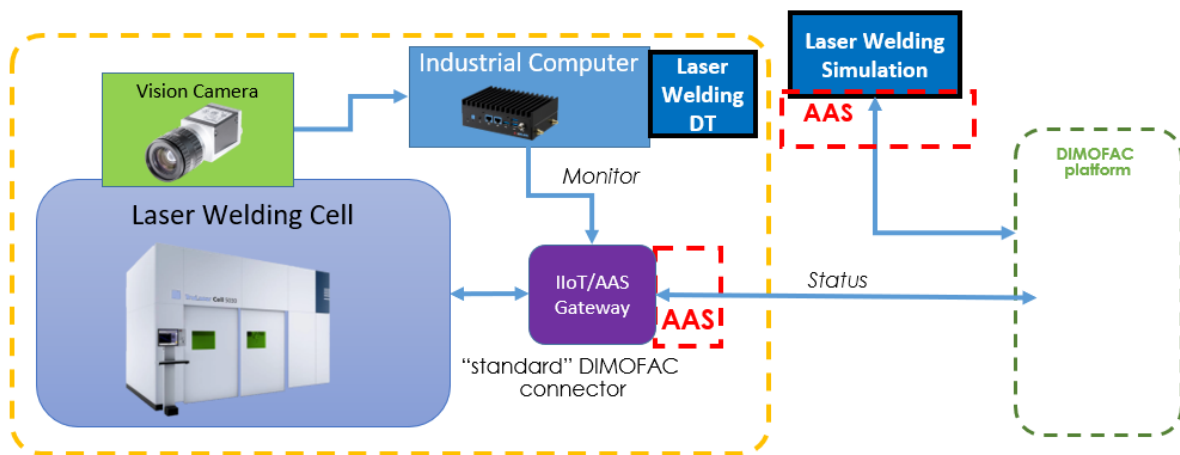


Figure 57: Laser Welding Module Architecture

The hardware components of the module are:

1. Laser Welding Cell:

The laser welding cell that is investigated is the TruLaser Robot 5020 constructed by Trumpf.

2. Vision Camera: A 2D vision sensor (Basler camera)
3. Industrial Computer:

A Linux-PC that runs the gap detection algorithm

The software components of the module are:

1. Laser Welding Gap Detection:

A software tool that is capable of detecting gaps between parts to be welded

2. Laser Welding Simulations:

A software that calculates process related data for assisting the reconfiguration phase.



DIMOFAC



4.13.3 Interfaces

4.13.3.1 Interfaces Physical Assets-Central Unit

Ethernet TPC/IP is used to connect the camera used for the vision-based quality control system with the industrial computer of the laser welding station.

The implemented quality control system is connected to the laser welding AAS model. Inside the AAS model of the laser welding, a capability named "gap detection" has been modelled in order for the Task Planning system to be aware that the specific laser welding station is capable to perform a quality check task of the assembled part before the welding task and therefore such a task can be assigned to this station. The result of this check is exposed through the same AAS to the human operator to act accordingly (proceed with the welding if no gaps are detected or correct the assembly in case of detected gaps). The implemented QC – AAS interface is established with custom Python – AAS connectors by making use of GET/PUT commands of the requests' library.

4.13.3.2 Central Unit-DIMOFAC platform

The communication of the information from the central unit of the module to the DIMOFAC platform is enabled thanks to RESTful webservices (REST API) commands and mainly through GET/PUT methods.

4.13.4 AAS modelling

The AAS model of the Laser Welding cell has also been developed with regards of the guidelines provided by DIMOFAC task T1.4 Common Information model, wherever applicable. The defined submodels are listed below:

1. Identification submodel:

Here, information regarding the machine ID, such as the model, identification factory number, etc. are stated.

2. Technical Data submodel:

On this submodel, technical information regarding the laser welding module is presented, as also a reference to the robot AAS of the laser welding cell.

3. Simulation submodel:

On this submodel, the inputs and outputs of the simulation tools are modelled.

4. Gap detection submodel:

The gap detection related data, are modelled here, such as the camera distance, or the detection point ID created, etc.

5. Capabilities submodel:

The capabilities of the Laser welding module are modelled here, following the same modelling method as in 4.10.3.



DIMOFAC



4.13.5 Software

As stated above, the software components are two for this module and these are:

1. Gap Detection System:

Tool that is capable of detecting gaps occurred during the assembly of sub-parts on a jig before the final welding. In order for the welding process to be executed correctly, the gap between the various sub-parts should not exceed a specific threshold. Thus, the designed tool is used before the welding, and more specifically during the robot teaching phase to measure the existing gaps during the robot teaching and notify the operator in case of a detected gap that exceeds the acceptable threshold. Therefore, the operator is allowed to correct the assembly of the part before the welding minimizing the possibility of a faulty welding.

2. Laser Welding Simulation tools:

Tools that can simulate and predict process related information, useful for process adaptation during the reconfiguration phase. This tool has been implemented on MATLAB (Figure 58), using as input an .xls file, while exporting the outputs of the simulation on a .csv file.

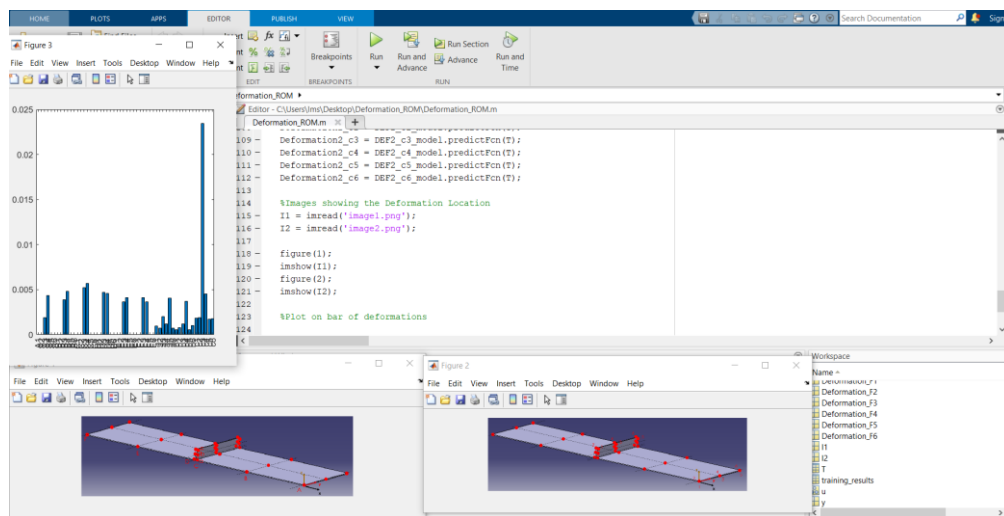


Figure 58: Laser Welding Simulation Tool Screenshot

4.14 MODULE 14 – Flexible module (assembly, cabling quality control)

4.14.1 Description

The SchaltAG pilot line 5 will be equipped with at least three flexible modules, the tasks performed on this module are the following, assembly, cabling, and quality control. The tasks will be performed by workers and a collaborative robot place on the robotic module described in the next chapter. The flexible modules are flexible in way they are able to be removed and interchanged between them thanks to a standardized electro-mechanical interface showed in Figure 59.



DIMOFAC

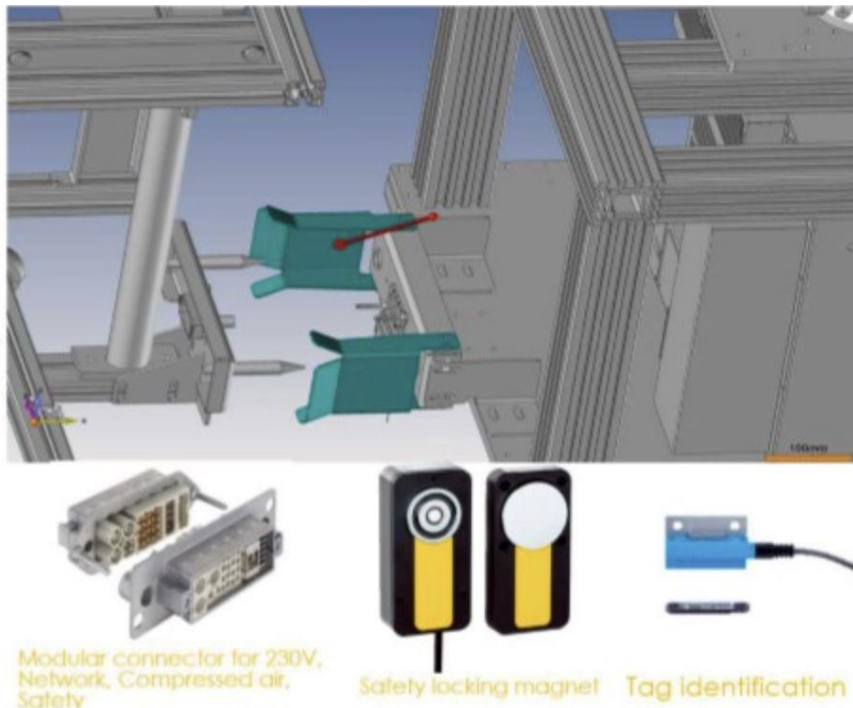


Figure 59: Electro-mechanical interfaces between the flexible module and robotic module

4.14.2 Architecture

One flexible module is composed of the two IO-link Sick (encoder and distance) sensors shown in Figure 60 to measure the conditions of the module: those devices are connected over IO-Link to a Siemens IO-Link master, which is connected to a S7-1200 Siemens PLC. On the PLC an OPC-UA server is programmed to first provide the DIMOFAC platform with the condition data, and second to provide the interface to control remotely the flexible module from the TopSolid DT. Figure 61 shows the infrastructure of the flexible module.



Figure 60: Sensors of the modular module



DIMOFAC

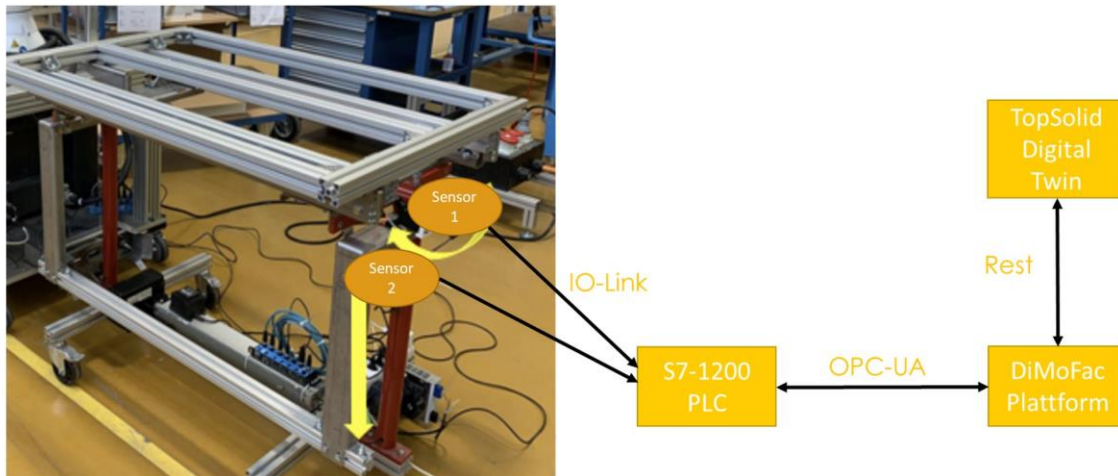


Figure 61: Pilot line 5 flexible module infrastructure

4.14.3 AAS modelling

In the AAS Modelling shown in Figure 62 of the flexible module, you find the sensor values and the trigger signal to control the modules remotely. Z_Height means the current height of the workplace, and Tilt_Angle is the angle of the workplace compared to its zeroed position, which is horizontal.



Figure 62: Pilot line 5 flexible module AAS ModelPnP capabilities

Thanks to the common data model used for all the flexible modules in the pilot line 5, all of them are connectable through the DIMOFAC platform.

4.14.4 Software

The software is directly programmed with siemens TIA Portal 16 and upload on the Siemens PLC

4.15 MODULE 15 – Robotic handling

4.15.1 Description

The robotic handling module is the centre of the pilot line 5 for SchaltAG, where a collaborative robot is used to take over repetitive tasks, and the preparation of the work done by hands. In the SchaltAG pilot line a Fanuc CRX10iAL has been chosen for its reach and payload capabilities and its simple and intuitive programming, which will be needed to reengineer the pilot line for every new product produced by SchaltAG. Figure 63 shows the robot and its accessories, a Schunk EGP-C 64 collaborative gripper.



Figure 63: Fanuc CRX10iA with Schunk EGP-C-64 cobotic gripper

4.15.2 Architecture

Figure 64 shows the architecture of the robotic module of the pilot line 5. The robot will communicate to the Dimofac Plattform over TCP-IP or OPC-UA, defined after the final interface tests. The target is to extract the different angle values of the six joints from the robot controller. After receiving the joint angle values, those will be used, for ex., by the DT of TopSolid.



DIMOFAC

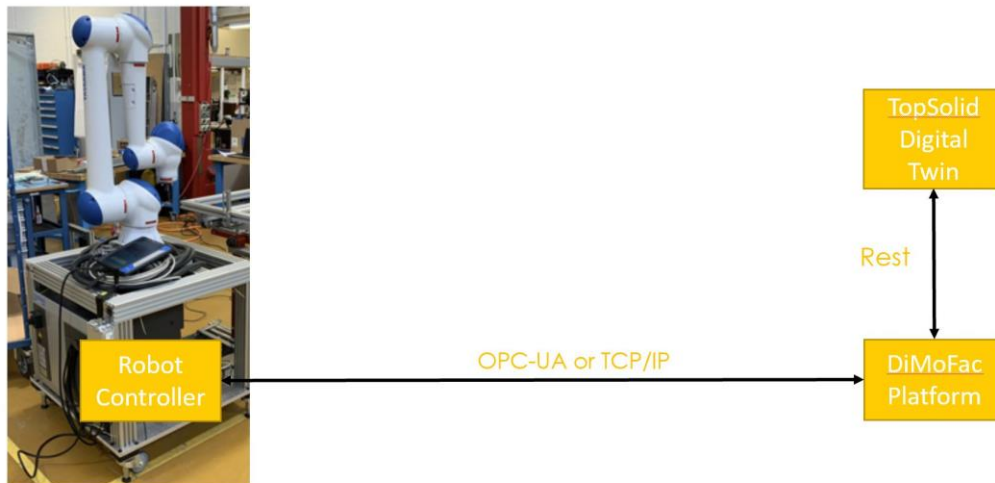


Figure 64: Pilot line 5 robotic module infrastructure

4.15.3 Interfaces

The interface between the robot and the gripper are inputs and output to control the movement of the gripper. The Safety Magnet are as well connected to the robot in order to release or lock them in place.

The main software interface of the robot is an OPC-UA server where all the needed data about the robot are stored, this server is provided directly by Fanuc. The OPC-UA server is then connected to the DIMOFAC platform.

4.15.4 AAS modelling

In the AAS Model of the robotic module, the conditions of the robot axis are described, in this case each joint in degrees, Figure 65 shows the robotic module AAS model.

AAS "SLT_Robot" [IRI, https://example.com/ids/aas/3271_9030_2022_2208] of [IRI, https://example.com/ids/asset/7380_9030_2022_0658 , Instance]	
SM "Fanuc_Robot" [IRI, https://example.com/ids/sm/0181_9030_2022_9446]	
Prop "Pivot6"	= 0
Prop "Pivot5"	= 0
Prop "Pivot4"	= 0
Prop "Pivot3"	= 0
Prop "Pivot2"	= 0
Prop "Pivot1"	= 0
Prop "RobotId"	= Fanuc CRX 10 AL

Figure 65: Pilot line 5 robotic module AAS Model

4.15.5 PnP capabilities

The OPC-UA server of the robot allows PnP capabilities because of its standardized Data Model given by the manufacturer. In its data model, the robot's condition is described and permits as well to choose programs remotely.

A second PnP capability is given by the direct teaching function of the collaborative robot. It will raise the possibilities of reengineering the tasks realized by the cobot for



DIMOFAC



SchaltAG, every few days, the job will change, and within a short time, a SchaltAG worker will be able to re-teach the robot for its new task.

4.15.6 Software

The robot is programmed directly with Fanuc built-in programming interface.

4.16 MODULE 16 – Parts storage

4.16.1 Description

The SmartLocker has sensors that keep track of what is happening to the assets. Every bin has a weight sensor, the locker door senses when opened and all of the information transmits to Bossard's Online Plattform Arims. Every time user opens the door. All the bins are weighted. Afterward the system can provide the user with an exact sum of the parts in the locker. The SmartLocker and the users can ensure the disponibility of the pieces in stock

4.16.2 Architecture

Figure 66 shows the architecture of the parts storage module in pilot line 5—the Smartlocker connection to the Dimofac Platform over the provided RestApi.



Figure 66: Parts storage infrastructure

4.16.3 Interfaces

The interface used here with the Parts storage module is the REST API provided by Bossard. In the Rest Api the type and quantity of the parts are listed.

4.16.4 AAS modelling

In the AAS Model of the parts storage module, the amount of parts in each bin are listed. Figure 67 shows the parts storage module AAS model.



AAS	"Parts storage"	[Custom, AssetAdministrationShell---25F955D7]
SM	"Bossard-Smart-Locker-1"	[IRI, https://example.com/ids/sm/7204_7072_4022_0803]
Prop	"Bin1"	
Prop	"Bin2"	
Prop	"Bin3"	
Prop	"Bin4"	
Prop	"Bin5"	
Prop	"Bin6"	
Prop	"Bin7"	
Prop	"Bin8"	
Prop	"Bin9"	
Prop	"Bin10"	
Prop	"Bin11"	
Prop	"Bin12"	
Prop	"Bin13"	
Prop	"Bin14"	
Prop	"Bin15"	
Prop	"Bin16"	
Prop	"Bin17"	
Prop	"Bin18"	
Prop	"Bin19"	
Prop	"Bin20"	
Prop	"Bin21"	
Prop	"Bin22"	
Prop	"Bin23"	
Prop	"Bin24"	
Prop	"Bin25"	
Prop	"Bin26"	
Prop	"Bin27"	
Prop	"Bin28"	
Prop	"Bin29"	
Prop	"Bin30"	
Prop	"Bin31"	

Figure 67: Parts storage AAS Model

4.16.5 PnP capabilities

The REST API definition gives the PnP capability.

4.16.6 Software

The ARIMS software developed by Bossard manages the system. It has functionalities similar to an ERP. It controls the amount, the type, and the reordering of the different parts in the 34 boxes.

4.17 MODULE 17 – Bin picking handling

4.17.1 Description

The module with ID 18 is a bin picking task being developed for the Sculpteo pilot line (PL6). The main purpose of this module is to pick specified parts from a mixed container using a small payload collaborative manipulator.

The part recognition and pose detection of the object is done using a colour-depth camera and the ROS framework to run the part recognition algorithm.

4.17.2 Architecture

The architecture consists mainly of the COMAU Racer 5 Cobot, the Zivid One+ Medium 3D camera and the gripper (see Figure 63). The two resources are managed by ROS framework, in particular with an algorithm of image recognition through which the object to grasp is selected. The objects lay in a box with mixed parts, and the aim is to recognize the desired item and its pose, in order to give the motion command to the cobot (see Figure 69).

The gripper that will be mounted on the robot wrist is still to be decided, however the most likely to be used is the OnRobot 2FG7. This gripper is described thoroughly in section 4.15.

The algorithm for part recognition will be developed as a ROS module and will be adapted to the specific parts to be picked. Based on a CAD model of the object to localize and a 3D point cloud of the scene in which the object to grasp lay, the module will register the CAD to the scene (estimate the 3D pose of the object). With this pose, the robot will be able to grasp the object (see Figure 68).

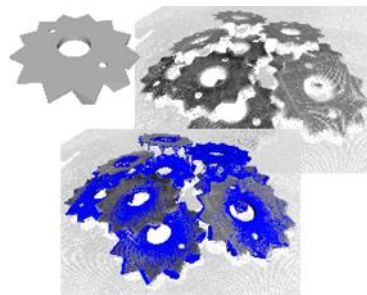


Figure 68: CAD model (top left), 3D scene provided by a 3D sensor, with different instances of the object (top right) and multiple registration of the CAD in blue over the scene (middle below)

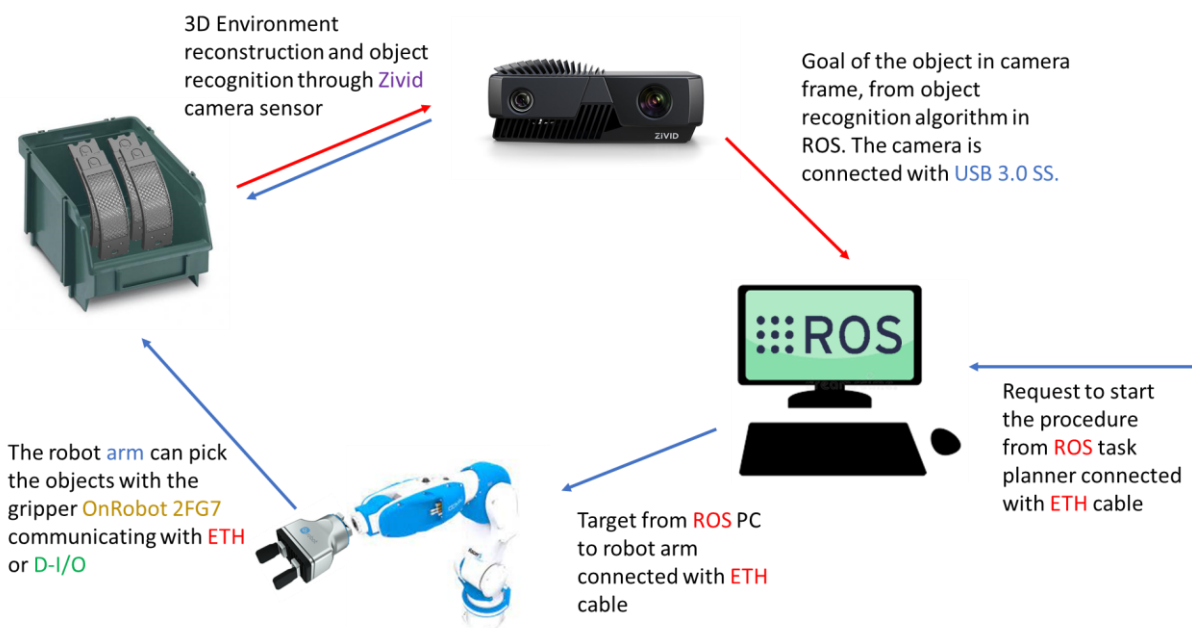


Figure 69: Bin picking module architecture

4.17.3 Interfaces

The interface between the Racer 5 Cobot and the computer hosting the ROS framework consists of an Ethernet cable communicating in TCP/IP protocol. This enables ROS to give motion commands to the robot.

The camera is connected to the ROS computer as well, using USB 3.0 superspeed and powered by its 24V, 5A power supply using an M12 x 5 pins male connector.

The gripper is connected to the robot wrist through an adapter flange and it communicates with the robot controller either via a set of 8 digital inputs and 8 outputs, or via Ethernet.

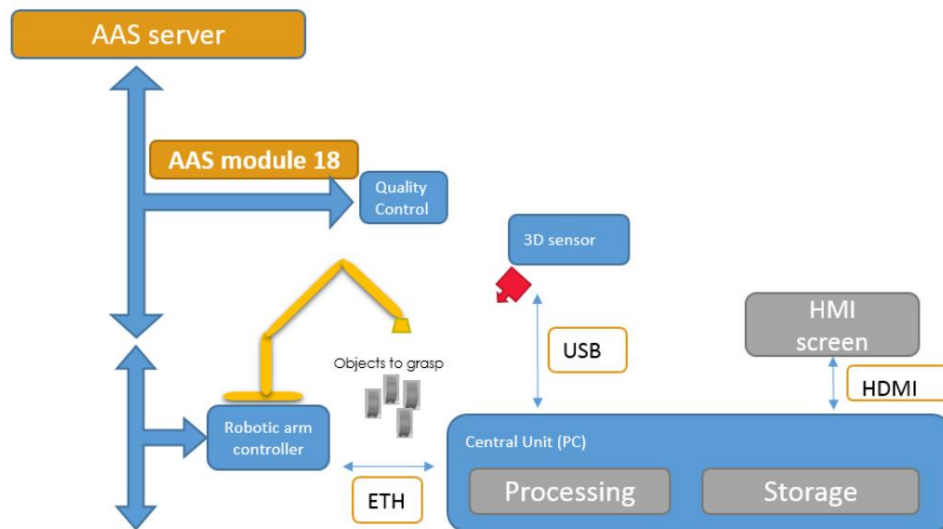


Figure 70: Global Architecture of the Bin Picking (module ID 17)

4.17.4 AAS modelling

The AAS model was defined using AASX Package Explorer (version 2021-08-18), following the identification assigned within work package 3, namely id 17.

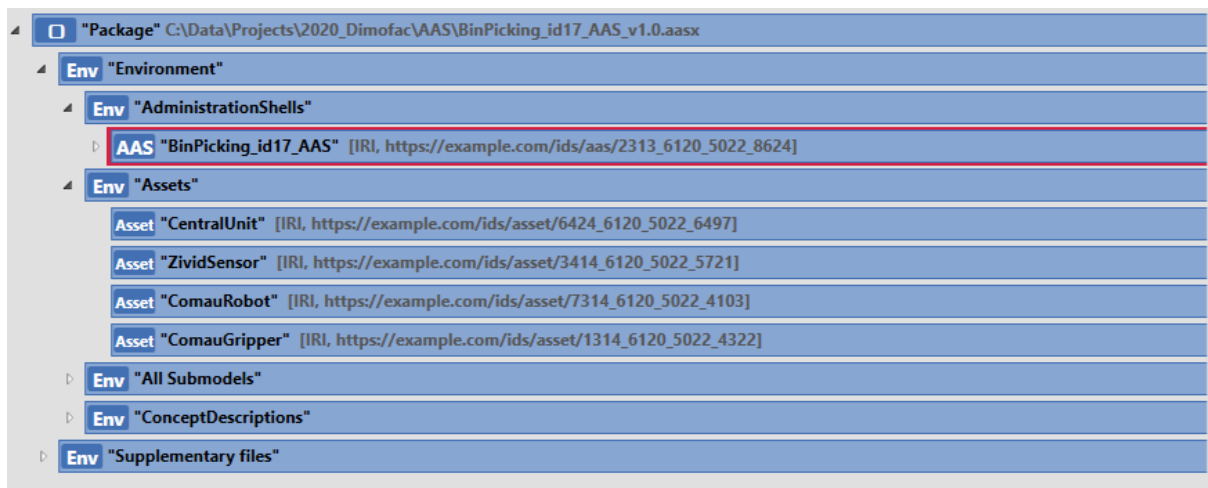


Figure 71: Bin Picking AAS model

This module orchestrates the different following elements:

- The Central Unit hosts the ROS software elements.
- The Zivid Sensor provides the 3D data to be processed by a ROS node.
- This processing results in a localization in space to which the robot has to be sent.
- The Comau Robot and Gripper are then used to grasp the localized part.

4.17.5 PnP capabilities

The Plug and Produce capabilities of this module consist in the bin picking task itself. Thus, it is possible to create a complete module with the robot arm, the robot program, the camera and the ROS modules, and this can be adapted to different types of objects to be grasped.

This module can be then applied to different environments, as well as different sizes of components and bins.

4.17.6 Software

The software modules to be developed involve mainly the algorithm for part recognition, which is a work in progress led by CEA and written in the ROS framework. The other ROS modules and drivers have already been developed and only need to be interconnected.

The list of the software sub-modules is the following:

- COMAU ROS driver
- Camera drivers
- Camera ROS module
- Image recognition algorithm
- Gripper interface program (digital I/O or Ethernet connection). This program is either written in the gripper's control box

4.18 MODULE 18 – Quality control – X-ray for additive manufacturing

4.18.1 Description

The module with ID 18 is a quality control device being developed for the Sculpteo pilot line (PL6). The main purpose of this device is to verify the quality of printed parts in the production chain of Sculpteo. These are plastic parts (polymers) mainly produced with a selective laser sintering process (SLS), on dedicated machines which work with batches of several samples. The module ID17 will pick individual parts from a finished batch and will transfer them to this QC module.

The device employs an adaptative method based on X-ray imaging. Depending on the input objective, it will be able to either provide a single radiographic image, a set of radiographies or a 3D result obtained from a full computed tomography (CT) scan.

4.18.2 Architecture

The module is composed of several devices and will be implemented as a shielded cabin having the following dimensions: 100 cm in length, 50 cm in width and 60 cm in height. The cabin will include an automatized sliding door on the top side. An overview of the CAD design to date is presented in Figure 72.

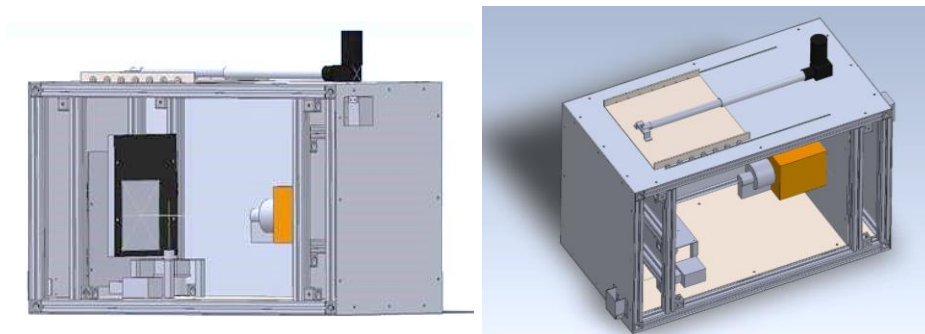


Figure 72 - CAD design of the inspection cabin of ID18 module

The module architecture is presented in Figure 73 and besides the inspection cabin it includes a central unit, a Human Machine Interface (HMI) as a screen and optionally external storage.



DIMOFAC

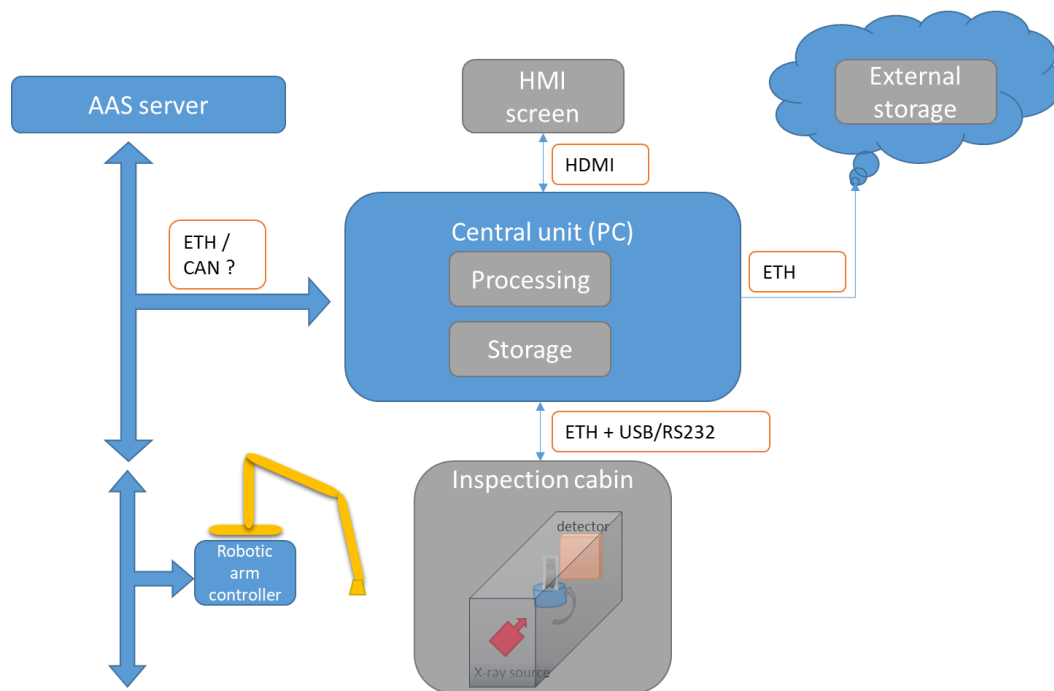


Figure 73 – Global architecture of the ID18 module

The hardware elements of the module are:

1. Central unit

A PC running Windows will be used as the main control unit and the connection to the DDP will be through Ethernet.

2. HMI

The module HMI will be provided through a standard computer screen as well as a typical keyboard and mouse.

3. Cabin with automatic door

The shielded cabin will have a sliding door on the top-side, in order to allow the access to the scan area. The door will be operated through an actuator or motor and will be controlled by the central unit.

4. X-ray source

The X-ray source is an electrical generator produced by Hamamatsu Photonics, model L9181. It is a compact and fully integrated microfocus X-ray source, working at up to 130 kV and 39 W.

5. X-ray detector

The X-ray detector is a Digital Detector Array (or Digital Flat Panel Detector) and more precisely an amorphous silicone active TFT-diode 2D array with a direct deposition of CsI:TI crystals as scintillator elements. The detector has a matrix of 1024x1024 pixels of



DIMOFAC



200 μm^2 . It can record up to 25 fps. The image acquisition system is a PC software running on Windows OS and the interface is ensured by a GigabitEthernet connection.

6. Turntable

A turntable is necessary for CT scans, for which a high number of projections have to be acquired (e.g., 720 projections). The hardware is a stepper motor including a gear reducer for a better angular precision. The turntable includes a stepper controller receiving TTL pulses from the Central unit in order to perform rotation steps. The connection is direct (twisted pair cable) from an extension card on the PC.

The software elements are the following:

1. Module control of the inspection sequence

A Python script will ensure the global sequencing of the QC process and external connections. In the framework of the DIMOFAC Digital Platform, this program will be a connector to the main server and will also run the QC sequence.

2. X-ray source control

The X-ray source is controlled by a specific software, provided by the manufacturer. An operator needs to start the program and to ensure that the startup sequence was correctly executed.

3. X-ray detector acquisition system

The acquisition system is based on proprietary code of the manufacturer. A command line code has been developed at CEA that enables an acquisition started from an external process. This executable code will be called by the main control program.

4.18.3 Interfaces

The main interface to the DIMOFAC digital platform will be a connector based on the MQTT protocol, implemented on a mini-server built for the module. The physical connection will be a standard ethernet cable.

Internal to the QC module, several interfaces are used. The central unit function as a Master, and all other elements are receiving commands from it. The interface to the XR source uses the RS232 protocol and its implementation is proprietary (Hamamatsu Photonics). An HMI is provided by the manufacturer for controlling the X-ray source. The interface to the XR detector is also proprietary to the manufacturer (Perkin Elmer) and physically it is implemented through a Gbit Ethernet connection. The turntable is controlled through an USB extension card which sends direct commands to the motor controller and also to the actuator controlling the cabin door.



DIMOFAC



4.18.4 AAS modelling

The AAS model was defined using AASX Package Explorer (version 2021-05-02), following the identification assigned within work package 3, namely ID18. For an easier identification, the acronym DPL6XR (**D**IMOFAC **P**ilot **L**ine **6** **X**-Ray) was added. A screenshot of the model is displayed in Figure 74.

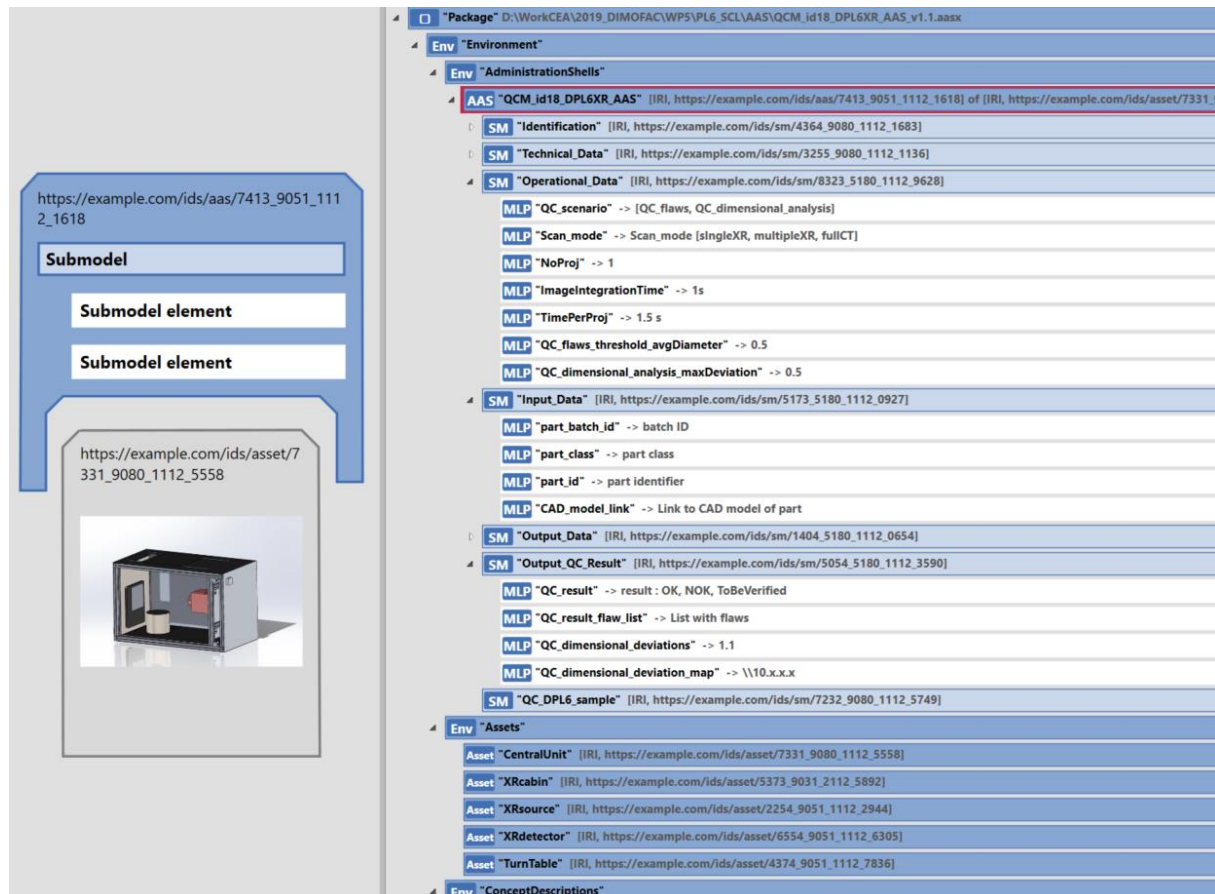


Figure 74 - AAS model of ID18 module

4.18.4.1 Submodels

A set of submodels was defined, which is an extension of a previous work as an Excel table that was used by WP3 contributors.

A brief description of the submodels is provided in the following section.

1. Identification

This submodel contains identification information about the module, such as the assigned name, the manufacturer, the serial number and the manufacturing year, the model version and a generic module category.

2. Technical data



DIMOFAC



This submodel contains the information related to the technical specifications of the module, i.e. the dimensions, weight and acquisition mode

3. Operational data

The operational data includes the set of parameters used for the inspection process and they represent the part of parameters depending on the scenario and selected use case. This list may be updated or extended, we mention here the most important properties: scan mode (which is a list of three possible modes), number of acquired projections, acquisition (integration) time per image, and total time per acquired image.

4. Input data

The input data contains information and parameters regarding the necessary inputs for the process in terms of batch or part identification and characteristics. This dataset is common for all usage modes.

5. Output data

The inspection process needs some processing steps and partial and intermediate data can be stored for as historical reference or for further use. Besides an identification of the part and batch, this submodel contains the information on the result type and a link to a folder where intermediate data is stored.

6. Output QC result

This submodel contains only the final result of the inspection process, as an acceptance criterion (*QC_result* : *OK*, *NOK* or *ToBeVerified*), a list with identified flaws, and depending on the scan type, a list with dimensional deviations and if created, a link to a file describing the deviation map.

For the Sculpteo pilot line, several use scenarios are possible which involve all or only a subset of the modules. We will be focusing here on a scenario which employs the QC module id18 for two purposes:

- a) quality control as identification of potential flaws, or
- b) dimensional inspection or analysis

For these two usages, there is a set of common parameters and a set of specific ones. The common parameters are listed in the *Input_data* submodel.

The specific parameters are listed in the *Operational_Data* submodel

The outputs are separated into two categories. The *Output_Data* submodel includes the identification of the inspected part, the result type and a link to the result file. The *Output_QC_Result* submodule includes the final results of the QC process, starting with the general criterion from the possible outcomes [*OK*, *NOK*, *ToBeVerified*] and depending on the usage, either a list with flaws as a table, or a table with deviations along X/Y/Z axis and a link to a file containing a deviation map in 3D.



DIMOFAC



4.18.5 PnP capabilities

The PnP capabilities of the module are predefined and are managed through the communication to the DIMOFAC platform and with the PL orchestrator.

The main feature is the possibility to adapt the inspection process to a sequence set by the orchestrator. Two distinct cases can be considered:

1. Include or not the use of the inspection module (id 18)
2. When used, the inspection mode can be:
 - a. Flaw identification.
 - b. Dimensional verification.

Depending on the accuracy, which is set for a given type of sample, the inspection can be either done by verifying a single radiograph (or a small set), or the result of a CT scan. These are respectively verifications on 2D and 3D images.

4.18.6 Software

A specific software is being implemented for the inspection module. It is a Python based implementation and it ensures connection and analysis functionalities. Within a first panel, the AAS model of the module is loaded from disk. The model was defined with the help of AASX Package Explorer software. The second panel concerns the connection to the DIMOFAC platform and the orchestrator. When all inputs are set and the connection validated, the module is ready and the Run button will launch the inspection sequence which includes the analysis process. The sequence can be stopped with a button on the interface. Figure 75 displays a screen copy of the Graphical User Interface (GUI) for this module.

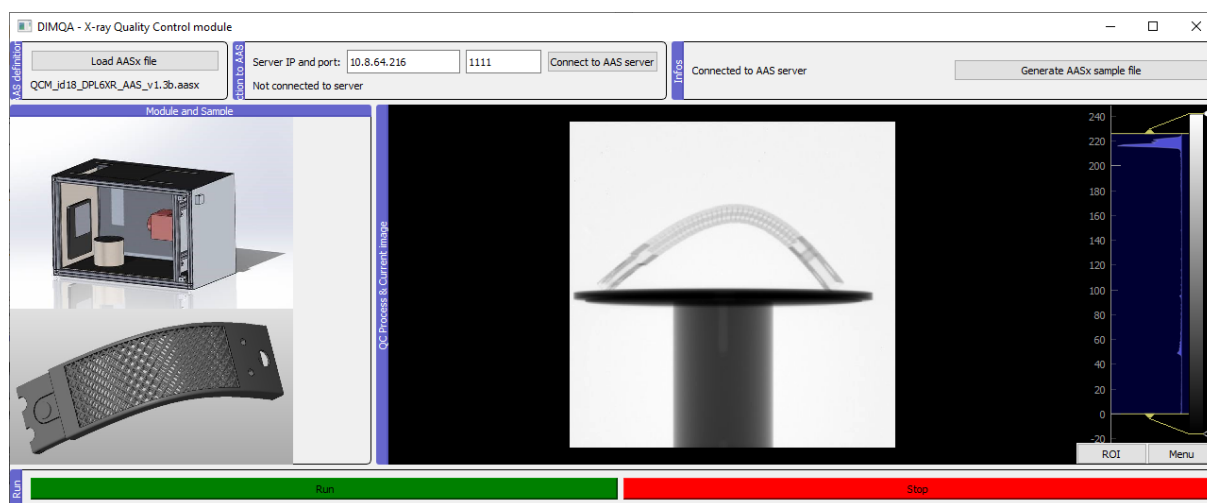


Figure 75: GUI of software for the ID18 module



5 Conclusions

The work presented in this deliverable shows the first implementation and testing of the Asset Administration Shell in the different DIMOFAC modules. This work consists of hardware and software interfaces development and implementation, and it is specific for each module.

This deliverable shows a variety of physical and non-physical assets connected through a central unit using different communication protocols (HTTP, OPC UA, MQTT,...), which will communicate with DIMOFAC platform using the connectors available in the platform and developed under WP2.

The data sent to the platform is based on Asset Administration Shell. This data is divided in common information (i.e. Identification submodel) and specific parameters that need to be shared/output parameters (i.e., Operation Data submodel) or the input parameters (i.e. Parameters submodel). This variety of AAS models, that will be used in DIMOFAC Pilot Lines, will be useful for T1.4, in which a Common Information Model has to be defined for achieving plug and produce capabilities in terms of module.

This task has a strong relation with other tasks: *T1.4 Common Information Model and data interoperability* where the common AAS submodels are defined, *T2.1 Horizontal Integration of production workflows* where the connectors of DIMOFAC platform are developed, *T3.2 On-line process control and quality assurance* where the monitoring systems is being developed, *T3.4 Intelligent digital twins for reconfigurable production modules* where the data exchange between virtual and physical worlds are defined, *T3.5 Deployment and testing of modules at pre-industrial scale* where the AAS models detailed in this deliverable will be tested and improved, *WP4 Reconfigurable Production Lines* in which the module has to be integrated at Pilot Line level and *WP5 Pilot Lines Integration & Demonstration* where the final implementation will be shown in final pilot line deployment.

As future work, the final hardware and software testing will be under T3.5 and some modifications in the AAS modelling will be probably necessary. This modelling should be also adapted taking in consideration the output of T1.4 and the needs from WP4 for integrating these individual modules in a complete pilot line.



6 Appendices

6.1 Glossary of Technical Terms

This section contains the glossary of all the technical terms and acronyms used in this document

Term/acronym	Explanation
IM	Injection Moulding
PL	Pilot Line
IMM	Injection Moulding Machine
TCD	Thermal Controller Device
ROM	Reduced Order Models
GUI	Graphical User Interface
RPM	Rotations per Minute
PnP	Plug-and-Produce
AAS	Asset Administration Shell
ROS	Robot Operating System
DT	Digital Twin/Digital Thread
QC	Quality Control
NDT	Non-destructive testing
UT	Ultrasound Testing
CFRP/GFRP	Carbon/Glass Fiber Reinforced Plastic
IR	Infrared
NFC	Near-field communication
ATL	Automated Tape Laying
QA	Quality Assessment
JSON	JavaScript Object Notation
HMI	Human Machine Interface
PnP	Plug and Produce



6.2 Result's disclosure plan

Every contributor of this deliverable is invited to assess the communication potential of the results reported in this deliverable. (Communication can take different forms: conference poster or oral presentation, articles, publication,). If that might be the case, thus, we should assess beforehand the exploitation potential of such results via an IP-survey in order to prevent any unexpected disclosure that would kill the possibility of patenting it later.

Do you plan to disclose some content of this Deliverable?

Each contributor of the deliverable should answer the above question on the table below:

#	Organisation	Sensitive information you do not want to disclose in this document? (Y, N, maybe)	If the answer is yes or maybe, please give details
1	CEA	N	
2	SIEMENS	-	-
3	PHILIPS	N	
4	ALBEA		
5	VDL		
6	SCHALTAG	N	
7	ESI Group	N	
8	INTRA		
9	COMAU	N	
10	SCULPTEO		
11	FILL		
12	EIRE	N	
13	NIT		
14	TTS	-	-



DIMOFAC



#	Organisation	Sensitive information you do not want to disclose in this document? (Y, N, maybe)	If the answer is yes or maybe, please give details
15	EXOM		
16	FBR	-	-
17	TopSolid		
18	INNOVALIA	-	-
19	BRAINPORT	-	-
20	SFKL	-	-
21	CECIMO	-	-
22	SSF	N	
23	LMS	Y	AAS modelling. Publication/Conference about AAS modelling
24	IPC	N	
25	TECNALIA	N	
26	TNO	-	-
27	DFKI	N	
28	AIMEN	Y	AAS modelling and part of the software architecture. Publication/Conference about AAS modelling and dataflow
29	POLIMI	-	-
30	DATAPIXEL	N	