Production harmonizEd Reconfiguration of Flexible Robots and Machinery

Horizon 2020 – Factories of the Future, Project ID: 680435

Deliverable 6.2

Self-Adaptive Highly Modular and Flexible Assembly Demonstrator Design and Set-Up

Lead Author: *SmartFactory*[KL]

Version: 0.3.3 (2. Draft)
Date: 29 Mar. 2017
Status: Public

Version history

Version	Date	Content
0.1	13 Sep. 16	First ToC and effort allocation (SF)
0.2	31 Jan. 17	First Draft (SF); Included contribution from partners (IPB, UNINOVA, Siemens, Polimi)
0.2.1	03 Feb. 17	Included GKN Scenario description (MTC)
0.2.2	07 Feb 17	Changes based to the adapted Solution Matrix from WP4 (SF)
0.2.3	10 Feb 17	Included description for the Simulation Environment (Siemens)
0.2.4	13 Feb 17	Included description for the adapter (Loccioni)
0.2.5	28 Feb 17	Included WHR and eDistrict description (Polimi); Description of the reconfiguration tool and test environment (IPB) Compiled tool descriptions with information from the templates (SF)
0.2.6	01 Mar 17	Included Middleware description (HSEL)
0.3	03 Mar 17	Second Draft (SF)
0.3.1	22 Mar 17	Included tool description from Paro and Xetics
0.3.3	29 Mar 17	Included tool description from Lboro
1.0	29 Mar 17	Final

Author List (Alphabetical order):

André Hennecke (SmartFactory)
Arnaldo Pereira (IPB)
Benjamin Lee (Siemens)
Ellina Marseu (SmartFactory)
Evangelia Dimanidou (MTC)
Filippo Boschi (Polimi)
Giacomo Angione (Loccioni)
Jeffrey Wermann (HSEL)
José Barbosa (IPB)
José Dias (IPB)
Manfred Hucke (Xetics)
Martin Frauenfelder (PARO)
Nils Weinert (Siemens)
Paulo Leitão (IPB)
Ricardo Peres (UNINOVA)
Ying Liu (Lboro)

Abstract

PERFoRM WP6 "*Validation and demonstration of reconfigurable and self-adaptive systems*" is responsible for testing and de-risking the technologies developed within the project, before these are applied to the production environment. In particular, Task 6.2 "*Self-Adaptive and Reconfigurable Highly Modular and Flexible Assembly Systems*" is demonstrating the PERFoRM-Technologies and Architecture in a modular and reconfigurable environment with a focus on the IT level.

This deliverable gives an overview of the test scenarios for the industrial use cases and identifies the technologies to be demonstrated. In additional test approaches are presented in different environments. The second deliverable of this task (D6.5: *"Self-Adaptive Highly Modular and Flexible Assembly Demonstrator Documentation and Results"*) will report the results of the tests and the demonstrations.

Index

List of Figures

List of Tables

1. Introduction

1.1. Objectives and Goals

WP6 the "*Validation and demonstration of reconfigurable and self-adaptive systems*" has the goal to validate the use case requirements and to de-risk the technologies developed in this project before the deployment in the industrial use cases. The procedure and the effort of this work is structured into three tasks as depicted in the PERFoRM framework (Figure 1) proposed by D1.1 [D1.1]. The PERFoRM framework shows the classification of the testing and demonstration activities: Task 6.1 "*Self-Adaptive and Reconfigurable Machines and Robots*" considers the assets view and is therefore de-risking and demonstrating the technologies on the assets floor level. Task 6.2 "*Self-Adaptive and Reconfigurable Highly Modular and Flexible Assembly Systems*" validates and demonstrates the IT-Level of the PERFoRM architecture and is therefore mainly responsible for the integration and demonstration of the tools in a modular environment and Task 6.3 "*Self-Adaptive and Reconfigurable Large Scale Systems*" considers the process view to demonstrate the use case processes within the shop floor and IT-Level.

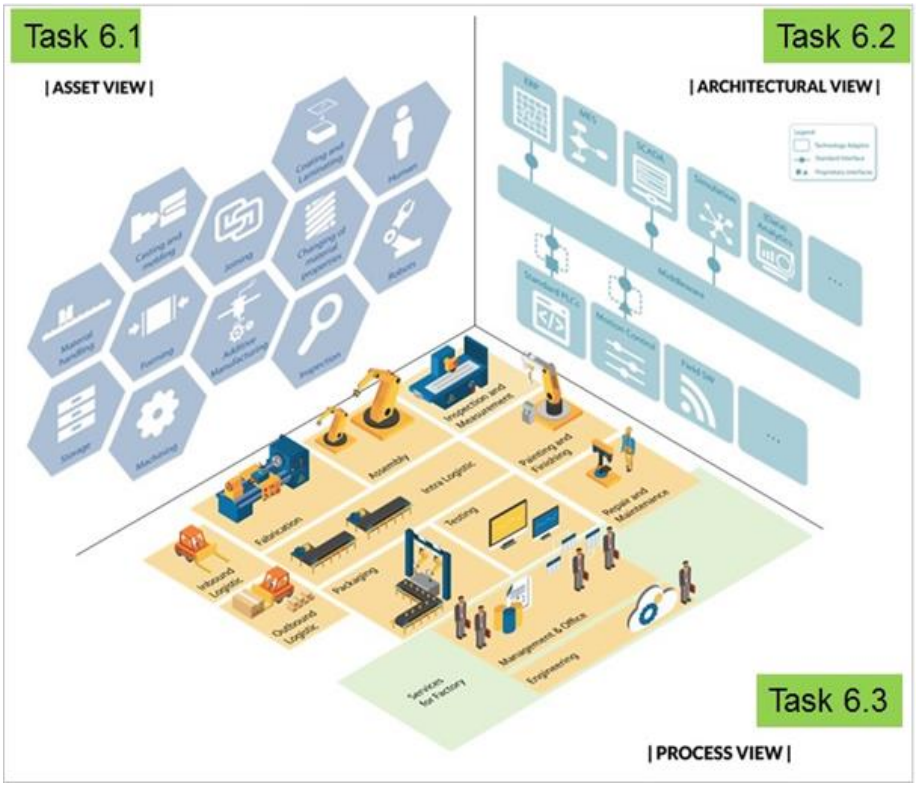


Figure 1: Classification of the testing and demonstration tasks

Task 6.2 is in particular responsible for the "Software Integration Tests" and the demonstration of the suitability of the PERFoRM software architecture in a modular and reconfigurable environment. Therefore this task focuses the achievement of the following goals:

1. Verify that the IT-Level of the PERFoRM architecture and its tools and concepts can be integrated in a proper way.
2. Validate that the use case requirements concerning the IT-Level are fulfilled.
3. Validate that the PERFoRM architecture works in modular environment.

To show that the PERFoRM architecture is suitable in a modular environment (Goal 3) and to validate and verify the integration of the PERFoRM tools and the fulfillment of the use-case requirements (Goal 1 and 2), a demonstration and test approach is considered (see section 8) where the PERFoRM

architecture and the tools will be connected to a real modular and reconfigurable demonstrator (see section 8.1).

This deliverable is the first of two in Task 6.2 and has the following objectives:

1. Define the scope of the demonstration and identify the IT-Components and their interaction.
2. Derive the critical test scenarios - scenarios with a high risk level.
3. Derive the environment requirements to setup the demonstrator
4. Define the test and demonstration approach to evaluate the criteria related to Task 6.2.

The second deliverable D6.5 “Self-Adaptive Highly Modular and Flexible Assembly Demonstrator Documentation and Results” will refine the critical test scenarios to test cases, documents the setup of the tests and the demonstration environment, reports the test results and shows the implementation of the demonstration scenarios.

1.2. Outline of the document

The outline of this report is as follows.

Chapter 2 presents the structured methodology used in this task to define the scope, collecting the test scenarios and to define the test and demonstrator environment requirements. Chapter 3 gives an overview of the PERFoRM architecture and its components.

The description of the test scenarios for the use cases Siemens “*Compressors*”, Whirlpool “*Home Appliances*”, GKN “*Aerospace*” and IFeVS “*Micro Electric Vehicles*” can be found in **chapters 4, 5, 6, and 7**. Each chapter describes the scope of the use case, the critical test scenarios and the features of adapter, middleware, planning logic, simulation, intelligent decision support and visualization developed in the work packages 2, 3 and 4.

Chapter 8 shows the test approach for the validation of the use cases and the demonstration approach to show the suitability of the PERFoRM architecture and concepts in a modular and reconfigurable environment.

Chapter 9 gives an overview of the test management, which includes the collection of environment requirements, a collection of risks and a test/demonstration coverage matrix.

2. Methodology

WP 6 is based on an overall testing methodology, with little changes depending on the specific task. The methodology for T6.2 is shown in Figure 2. This methodology has its core concept in describing test-scenarios reflecting the system and the use cases requirements, which will be refined to test cases which form the basis for performing and describing the necessary tests. Testing, in particular system and software testing based on test scenarios and test cases is state of the art and is a core element in the software test standards IEEE 829 [IEEE829] or its successor IEEE 29119 [IEEE29119].

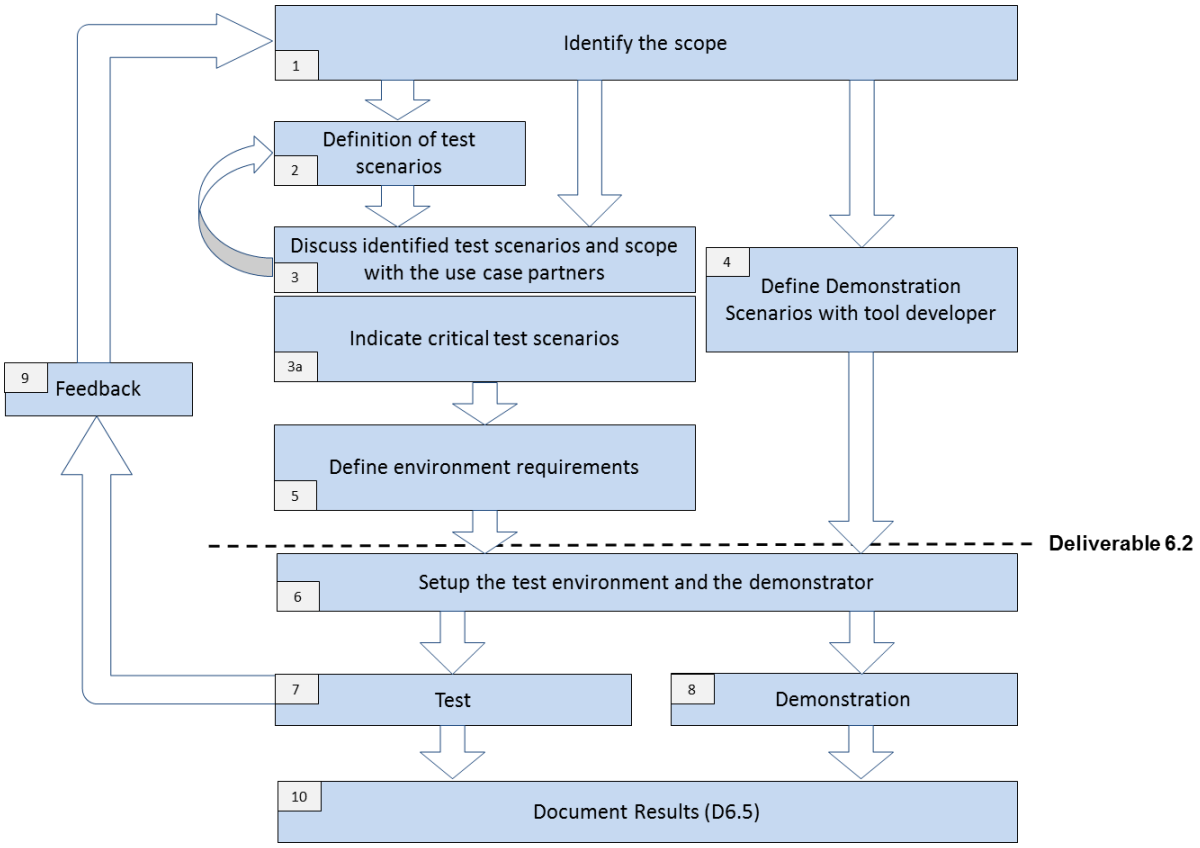


Figure 2: Structured methodology used in T6.2

In the first Step **(Step 1**) the technologies used in the PERFoRM-project or the technologies developed in the PERFoRM-project together with the concepts are identified. This step is important as the features provided by these technologies are directly related to a general PERFoRM-requirements or requirements from individual use cases.

Based on the identified scope together with the use case requirements, which were collected in detail in D1.2, the test scenarios for the use cases and the technology provider are defined in **Step 2.** Those scenarios and the technologies are discussed with the use cases in the next step (**Step 3**), to identify the critical test scenarios or scenarios with a high importance for a use case provider, but also to have a good coverage about what is necessary to test and to validate the requirements from the use cases. Critical test scenarios (**Step 3a**) will mainly cover features which have to be de-risked in a test-environment they can be implemented in a real industrial environment.

Because the requirements, concepts or technologies can change or envelope during such a complex engineering process or if the use case partners where not satisfied with the identified test scenarios, iterations back to the identification of scopes and the definition of test scenarios are considered.

Beside performing the integration tests and defining the necessary test scenarios, this task will also demonstrate the suitability of the PERFoRM tools and the architecture in a reconfigurable

environment. Therefore **Step 4** is defining demonstration scenarios which will be performed in one of the test labs.

In **Step 5** the environment requirements for testing the use cases scenarios are collected and matched with the available solutions in the different test labs, like MTC and SmartFactory.

Step 6 to **10** are not in the scope of this deliverable, therefore those steps will be defined more detailed in future deliverables of this task. These steps will mainly cover the setup of the test and demonstration environment, the detailed definition of the test cases and the actual implementation of the test. But also the documentation of the results and demonstration activities

3. Overview of the PERFoRM Architecture

The PERFoRM system architecture for the seamless production system reconfiguration, as described in D2.2 [D2.2], is based on a network of distributed HW devices and SW applications. This architecture addresses different ISA-95 levels, exposing their functionalities as services, and are interconnected in a transparent manner by using an industrial middleware, as illustrated in Figure 3.

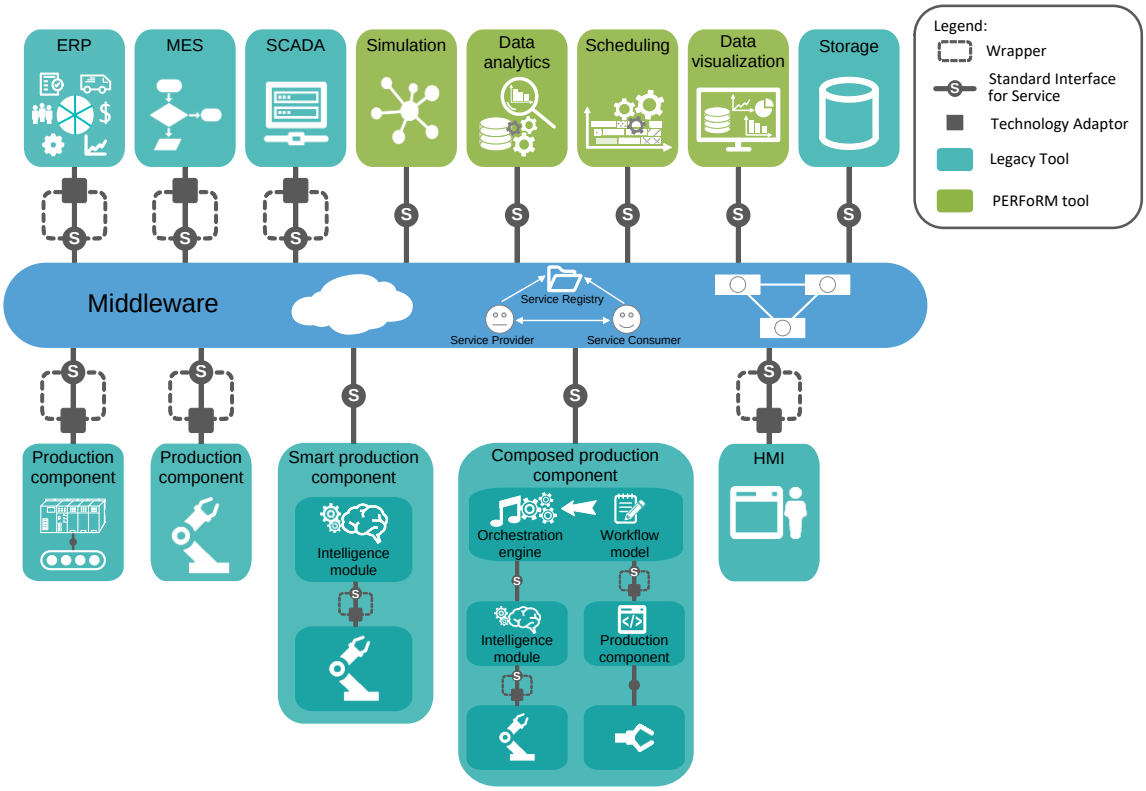


Figure 3: Overall PERFoRM system architecture.

The industrial middleware is a distributed service-based integration layer that aims to ensure a transparent, secure and reliable interconnection of the diverse heterogeneous hardware devices and software applications presented at the PERFoRM ecosystem.

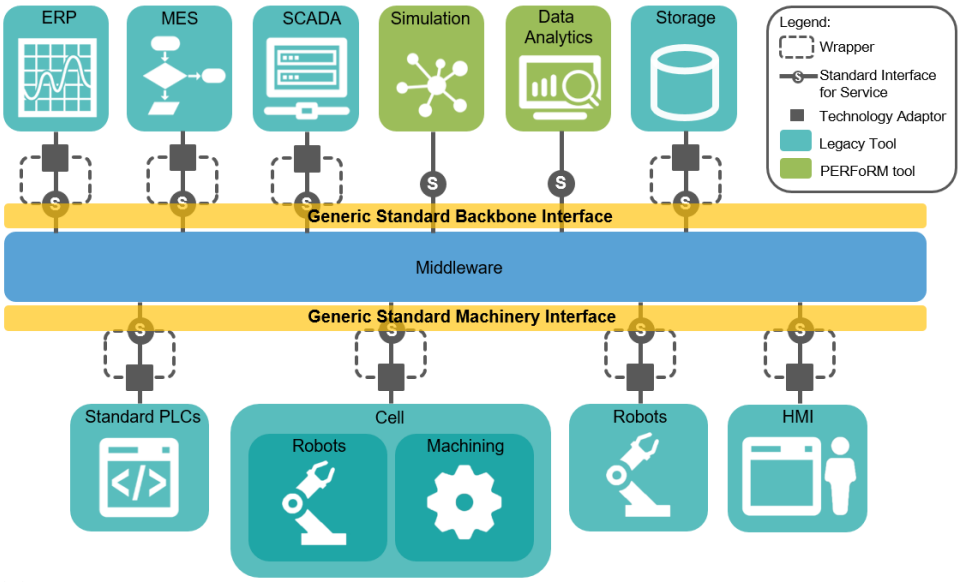


Figure 4: Generic Standard Backbone Interface in the PERFoRM-Architecture

Legacy hardware and software production systems are integrated by the use of customized adapters, which translate the device/module internal data model into the standard interfaces defined in PERFoRM (see Figure 9). As far as these interfaces and data models are used, the middleware will act as a platform to connect to whenever a certain service needs to be used.

Aligned with the general vision for Industry 4.0, one of the key challenges that the PERFoRM architecture aims to tackle is the interoperability in real industrial environments, coping with the representation and seamless exchange of data originating from a wide array of entities, often from different, albeit related, actions levels. Therefore, the interconnection of heterogeneous legacy hardware devices, e.g. robots and the respective controllers, and software applications, e.g. databases, SCADA applications and other management, analytics and logistics tools, is one of the main goals currently being pursued in this vision.

In this way, the PERFoRM architecture exploits the usage of standard interfaces throughout the whole system as the main drivers for plug ability and interoperability, aiming at enabling the connection between such devices and applications in a seamless and transparent manner. These interfaces support the devices, tools and applications with the means to fully expose and describe their services in a unique, standardized and transparent way to enhance the seamless interoperability and plug-ability. Therefore, a full specification of the semantics and data flow involved in terms of inputs and outputs required to interact with these elements was necessary. Therefore, from the system point of view, the standard interface specification and development abstracts the underlying function operation making transparent the way how the several architectural modules interact and operate.

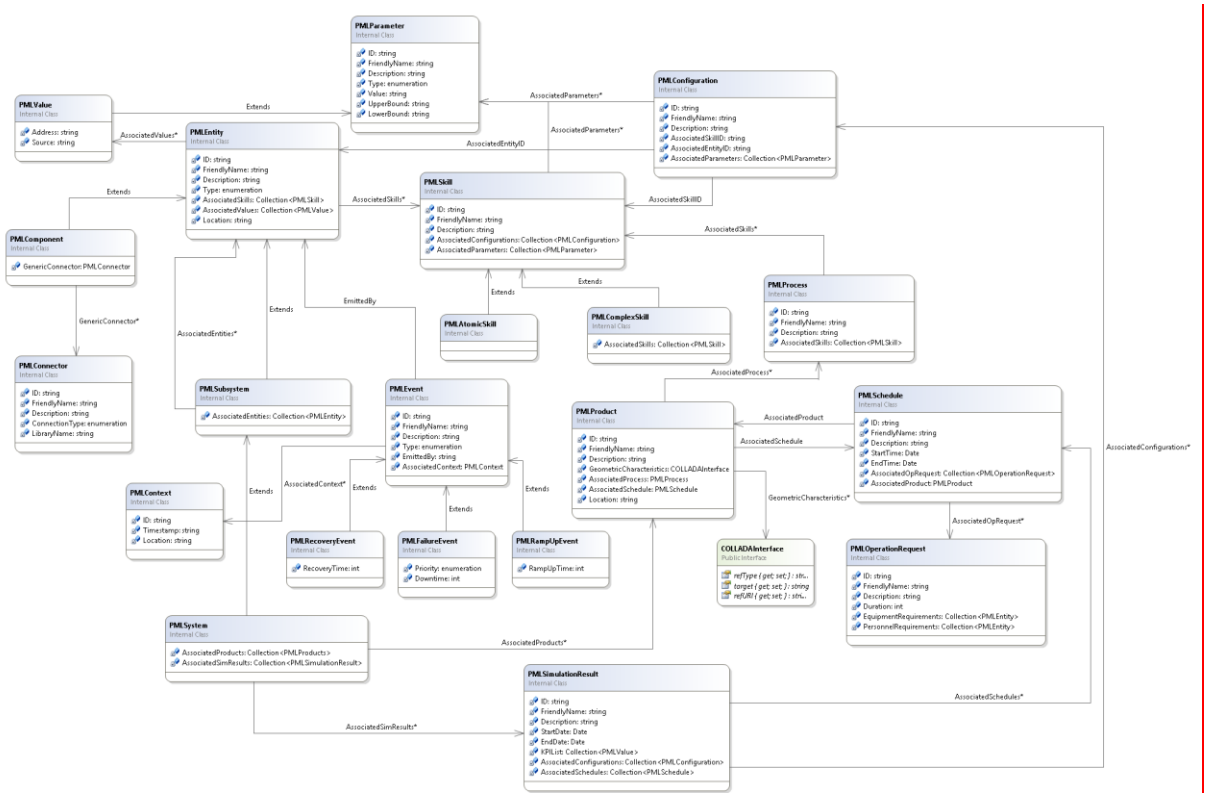


Figure 5: PERFoRM-Data Model

Additionally, and as a support to the standard interfaces, a common data model (depicted in Figure 10) was defined in Task 2.3, which is described in detail in D2.3, serving as the data exchange format between the PERFoRM-compliant architectural elements. This data model covers the semantic needs associated to each entity, namely the requirements related to each ISA-95 [ISA-95] layer. In this context, two particular data abstraction levels are taken into account, more specifically the machinery level, covering mainly layers L1 (automation control) and L2 (supervisory control), and the data

backbone level, which covers layers L3 (manufacturing operations management) and L4 (business planning and logistics).

As previously mentioned, manufacturing companies are usually characterized by the use of legacy and heterogeneous systems for the management and the execution of their production process. The innovative architecture proposed in the PERFoRM project can be industrially accepted and really adopted only if it is possible to integrate legacy systems. For this reason, technology adapters are key elements to connect legacy systems to the PERFoRM middleware and to transform the legacy data model into the standard interface data model (i.e. masking the legacy systems' data/functionalities according to the PERFoRM standard interfaces). The three adapter concepts developed in this project are depicted Figure 11 and encompasses not only the "Standard Technology Adapter", but also "Real-Time Technology Adapter" to capture data in real time and "HMI Technology Adapter" for the interaction with HMI. In this way, the technological adapters are only necessary when there is the need to connect a legacy component (e.g., an existing database or robot) to the PERFoRM system.

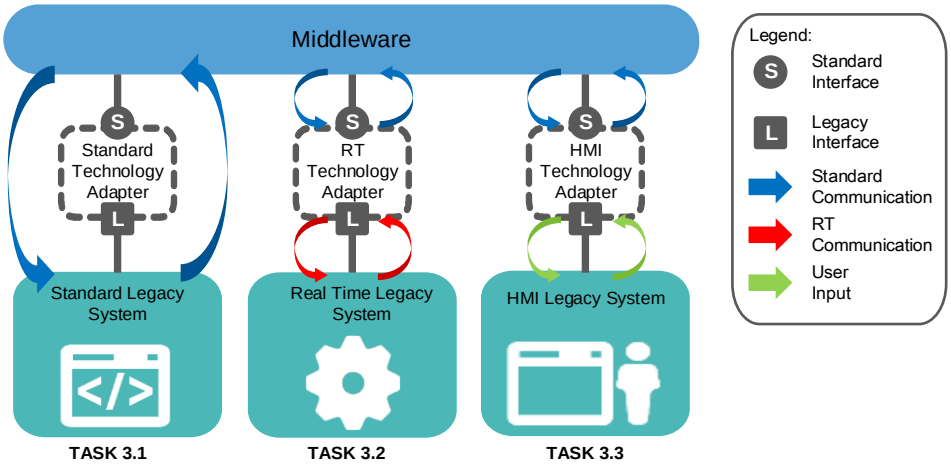


Figure 6: PERFoRM Adapter Concepts

The integration of the human in the loop is seen as a key factor to improve flexibility, as stated in D2.1 [D2.1]. Some recommendations were derived with implications for the planning and designing the human-machine and human-human interfaces, addressing two different levels: at strategic level, e.g., supporting decision-makers to take strategic decisions, as also at operational level, e.g., supporting operators or maintenance engineers to perform their tasks.

At a lower level, machines need, and are indeed becoming, to be smarter. To achieve this, robots and automation machinery need to be empowered with intelligence and higher processing capabilities to run more complex algorithms allowing them to process higher amount of data, producing a valuable analysis to be used when needed (i.e. in a real-time basis) and also supporting the seamless reconfiguration of the system and the achievement of self-* properties, e.g., self-adaptation, self-diagnosis.

In fact, the number of data being generated in shop floor plants is increasing at a very high rate. The proper analysis, locally (at the edge) and globally (at the cloud), of the collected data assumes a crucial aspect, generating new knowledge that can be used to detect trends, deviations and possible problematic situations beforehand and in a timely manner. Alongside with this, machines are being equipped with an increasingly higher number of sensors, generating enormous streams of raw data. Different data sources are commonly collected, each one having its own particularities, such as current, voltage or power, temperature and vibration, tool wearing and others. This naturally disjunctive data need to be analyzed and correlation patterns detected, identifying problems beforehand. Sending these streams of raw data over the network has the problem of overloading the network with *non-meaningful* data. Therefore, by analyzing this data locally and closer to the machine (where it is actually needed) has numerous potentialities, particularly by reducing the communication

latency times, enabling faster reaction to events or trend deviations and by reducing greatly the network overload.

In the PERFoRM architecture, the real time info processing module is located, if applicable, between the production components and the PERFoRM middleware, forming the smart production component. This intelligent module will react in a real-time basis due to operation/functional deviations and interact with the rest of the system as needed.

At a higher level, it is crucial to point out the need for tools particularly designed with advanced algorithms and technologies to support the production planning, scheduling and simulation may improve the system performance and reconfigurability. These tools should be PERFoRM compliant, i.e. following the service orientation and using the PERFoRM native interfaces. In PERFoRM architecture, these tools are inter-connected to the system by means of the middleware and are able to connect to the data available in the system, either for gathering data for the simulation or by generating new data to be used elsewhere by other tools. Naturally, these tools will use the PERFoRM native language. Nevertheless, legacy modelling and simulation tools are pluggable by means of the use of proper adapter and standard interface. WP4 will determine the landscape of existing modelling and simulation tools within the four use cases, and determine the need for these tools to be directly supported by the PERFoRM framework, or whether they will continue to be operated independently. Figure 7 depicts the developed software solutions in the solution matrix from WP4. The solutions are mapped to a use case where the provided functionalities are required. A green mark indicates that solution will be used in a use case and a yellow mark indicates that the use of this solution by a use case partner is still in discussion. This solution matrix set up the basis for the following use case chapters and the selection and description of the tools, the key features and the required tests.

	SOLUTION/MAIN GOAL	SIEMENS	EDISTRICT	WHIRLPOOL	GKN
SIMULATION: T(4.1)	*Agent Based Simulation*				Fixed
	Simulation of maintenance activities	Discussion			
	Simulation Environment	Fixed			
	Finite State Automata - KPI Excel-based Simulation	Discussion	Discussion	Fixed	Discussion
PLANNING LOGIC: L (T4.2 & T4.4)	*Multiobjective scheduling of production orders & maintenance task*	Fixed			
	Multi-objective plan. & scheduling for dyn., flex. manuf. sys.				Fixed
	Reconfiguration management in flex. manuf. sys.				Fixed
	Agent Based Reconfiguration Tool				Fixed
	Energy based planning with rescheduling		Fixed		
DESCISION SUPPORT: (T4.3)	*Value Stream Model in Excel*			Fixed	
	KPI monitoring with what-if-game functionality			Fixed	Discussion
	Min-Max Data Mining Toolbox	Fixed			
	Data Mining	Fixed			
	Automatic Monitoring and visualization of KPIs		Fixed	Fixed	
	Bayesian Diagnostics & Prognostics for Manuf. Equipm.	Fixed			

	Universal web based KPI visualization	Fixed			

Figure 7: PERFoRM Solution Allocation Matrix

All the aforementioned PERFoRM system architecture elements wouldn't be fully exploited if the architecture is not enriched with appropriate mechanisms for the seamless system reconfiguration as also the introduction of plug-and-produce concepts for a proper "modularity" approach. In PERFoRM, the seamless system reconfiguration is achieved by using the features commonly used in the development of distributed systems, namely those under the technological umbrella of Multi-Agent Systems (MAS) and Service-Oriented Architecture (SOA), particularly service- registry, discovering and composition, which also enhances the plugability. The plug-in of new services in the system is easily discovered by the other entities through the use of a service discovering mechanism, potentiating the cooperation/collaboration between different system components leading to a seamless system reconfiguration.

4. Siemens – Test Scenarios Description

The overall aim of the Siemens use case at the Duisburg Compressor Factory is found in an improved flexibility of manufacturing, focusing on the mechanical manufacturing of stator and housing parts [compare D7.1]. It is intended that by an improved prediction of the current machine health and its degradation within the near future maintenance tasks can be defined prior to actual machine breakdowns, moving over to a more predictive and condition based maintenance compared to the reactive or time based approach applied today. Following this approach, less production failures and quality issues shall arise and to better shift of production tasks between different machines shall become possible.

Logically, the necessary building blocks for fulfilling the use case are separated into two major groups (see Figure 8). At first, the definition or generation of maintenance tasks has to be carried out in order to define necessary works to keep manufacturing equipment available, or to maintain a certain level of breakdown risk respectively. Within the PERFoRM project, the condition prediction and task definition is focusing on three comparable machine tools (three Carnaghi turning machines) in the considered manufacturing area.

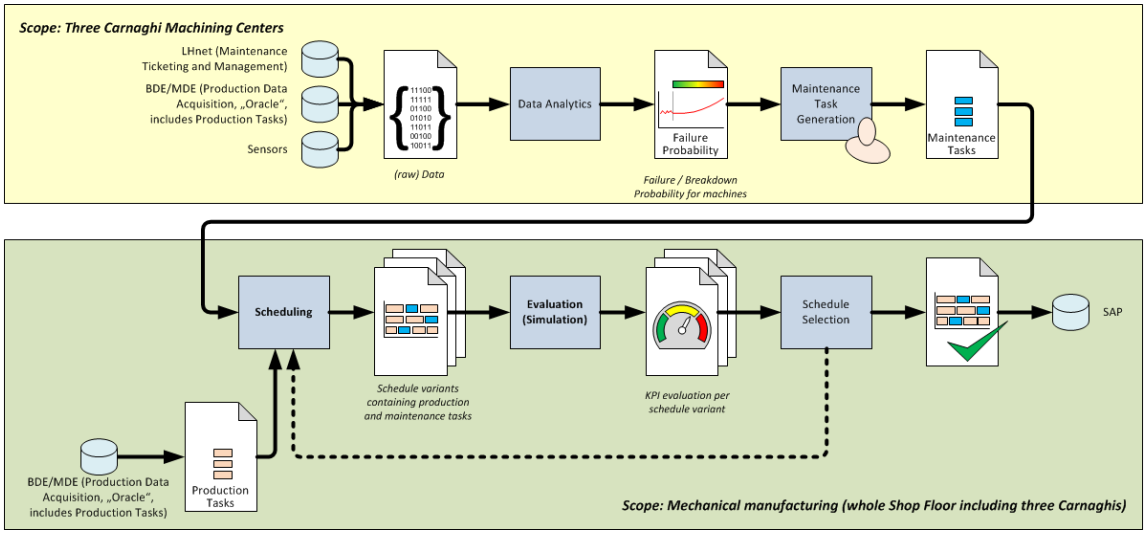


Figure 8: Siemens Use Case – Logic Workflow Overview

Since most maintenance works will still block equipment for the execution of normal production tasks for a certain time, as a second logical step an integrated scheduling of production and maintenance tasks shall be executed prior to finally deciding on the actual maintenance execution time. Scheduling here will be considering the actual situation which is found in the completion state of the several products within manufacturing at the particular time the scheduling is executed. Respectively, the resulting schedule is optimized regarding the given situation at scheduling time.

Maintenance times have to be fixed early compared to their actual execution time, e.g. for booking external suppliers. While schedule deviations are not to be completely avoided in a manufacturing environment, at the time the planned slot for a maintenance measure is reached, from an overall perspective that time will be probably no longer the ideal one for executing a maintenance task. Therefore, it is intended to generate several, slightly different schedules and test them in an evaluation/simulation according to their robustness in terms of limiting the (production-wise downstream) effects. Effects in this understanding are for example late completion times causing blocks in assembly processes due to lacking parts. Thus, the simulation based evaluation of multiple

schedule variants will consider the whole production area instead of being limited to the three machines mentioned above.

Assessment criteria:

1. Machine states are monitored correctly.
2. Prediction of machine failures.
3. Generation or suggestion of maintenance tasked based on the failures.
4. Generation of new schedules including the maintenance and the production task.
5. Factory schedules can be simulated and evaluated with the simulation result.
6. "Best" factory schedule are returned to the SAP-System manually.

4.1. Test scenario description

The following test scenarios will only include features which are in the scope of Task 6.2, which means the IT-Architecture.

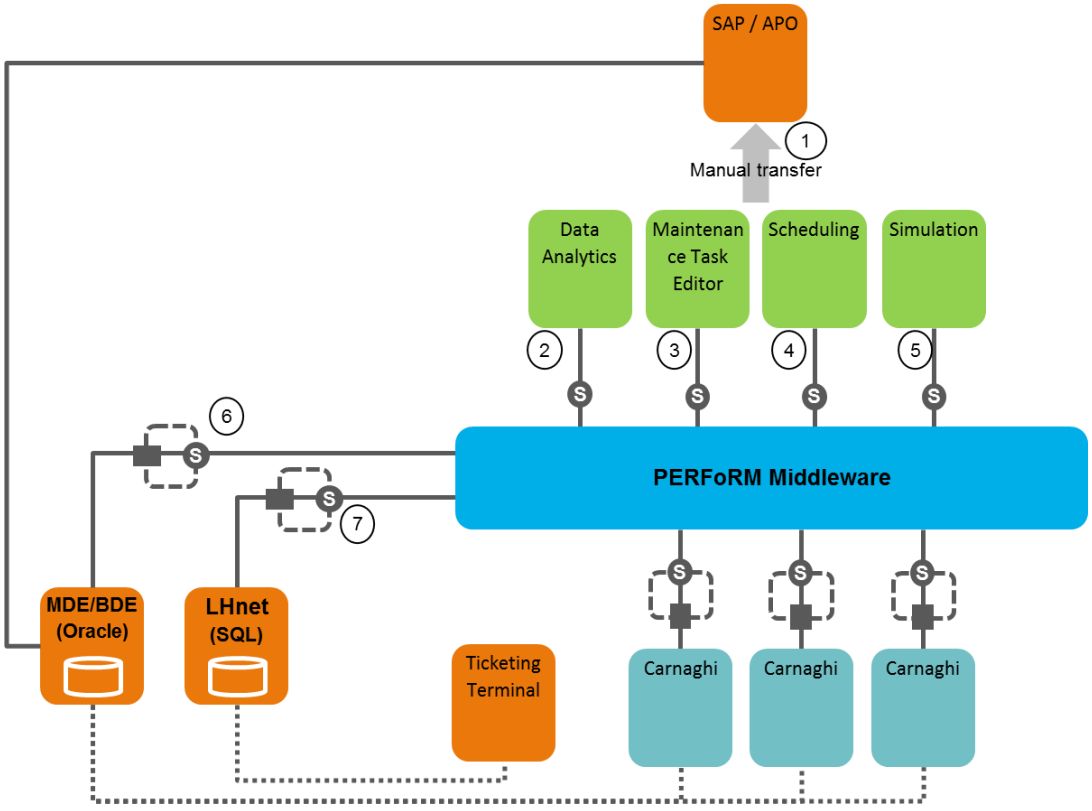


Figure 9: Components and interfaces in the compressor scenario

Table 1: Template for the test scenario description

ID	Test Scenario description
S-F-01	***Interface Middleware – Adapter - Database (MDE/BDE) (6):*** *Test that the MDE/BDE database (Oracle) can be accessed by the middleware via the adapter and that the adapter translate the data to the PERFoRM data model (PML)*

S-F-02	***Interface Middleware – Adapter – Database (LHnet) (7):*** *Test that the LHnet database (SQL) can be accessed by the middleware via the adapter and that the adapter translate the data to the PERFoRM data model (PML)*
S-F-03	***Interface Data Analytic to middleware (2):*** *Test that the Data Analytic tool can interact with the middleware via the standard interface including the valid PML exchange format.*
S-F-04	***Interface Maintenance Task Editor to middleware (3):*** *Test that the Middleware can interact with the Maintenance Task Editor via the PERFoRM standard interface including the valid PML exchange format.*
S-F-05	***Interface Scheduling to Middleware (4):*** *Test that the Scheduling tool can interact with the Middleware via the PERFoRM standard interface including the valid PML exchange format.*
S-F-06	***Interface Simulation to Middleware (5)****: Test that the Middleware can send a command to the Simulation Environment via the standard interface that includes valid PML arguments, and that the Simulation Environment can send a response to the Middleware via the standard interface that includes valid PML arguments.*
S-F-07	***Data Analytics access to database information (MDE/BDE) via the middleware (2+6):*** *Test that the data analytics tool gets the information from the MDE/BDE database via the middleware. Information are exchanged with the use of PML*
S-F-08	***Data Analytics access to database information (LHnet) via the middleware (2+7):*** *Test that the data analytics tool can access the LHnet database via the middleware. Information are exchanged with the use of PML*
S-F-09	***Maintenance tasks can be accessed by the scheduling system via the middleware (3+4):*** *Test that maintenance tasks can be read by the scheduling system via the middleware*
S-F-10	***Collaboration between Scheduling System and Simulation System via the middleware (4+5):*** *Test that the valid output of the Scheduling System provides a valid input to the Simulation Environment*
S-F-11	***Manual transfer of the factory schedule to the SAP-System (1):*** *Show that new factory schedule can be transferred back to the factory SAP-System*

4.2. Technology Scope and Test Items

With the previous defined test scenarios the following technologies are in the scope to be tested or demonstrated, highlighted in Figure 9:

- The **Adapter** to the databases which provides an interface to the middleware for accessing the data.
- The MDE/BDE (SQL) and LHnet (Oracle) **Database** with offline data.
- The capability of the **Middleware** to route the information to the involved components.
- The **standard interface** between the Middleware and the developed tools
- The **Information Model** which provides the exchange format between the Middleware and the Tools and the asset and tool description.
- **The Tool-Solutions**
  - Data Analytics
  - Maintenance Task Editor
  - Scheduling System

- Simulation Environment

4.3. Description of the components and features to be tested

The following subchapters describes the used technologies, which includes the standard interface and the middleware from WP2, the developed and used adapter technologies from WP3 and the higher level tools from WP4. The technologies and their use in this use case are shortly described, before the key features get summarized.

4.3.1. Architecture Technologies

Middleware

In the Siemens Use Case, the Middleware is used as a platform to integrate the various components, which have to be connected. Specifically, this includes the connection of two databases (Oracle and SQL) and the data, which they provide, and the PERFoRM tools, which are using the data for example to simulate or perform data analytics. The databases will have adapters, which provide the data in a PERFoRM compliant format. This data needs to be routed to the correct tool, using the interface it provides. This might also include a translation of the protocols used for the data transfer, depending on the choice taken within each specific application.

Table 2: Siemens Middleware - Key Features

ID	Feature Description
F-S-M-1	Middleware enables the data transfer from the Oracle DB to the Data Analytics tool
F-S-M-2	Middleware enables the data transfer from the LHnet DB to the Data Analytics tool
F-S-M-3	Middleware enables the data transfer from the Data Analytics tool to the Maintenance task editor
F-S-M-4	Middleware enables the data transfer from the Maintenance task editor to the Scheduling tool
F-S-M-5	Middleware enables the data transfer from the Scheduling tool to the Simulation tool

Standard Interface

For the Siemens scenario, regarding the standard interface between the tools seen in Figure 10 and the middleware, due to the bigger emphasis on data analysis and both scheduling and simulation for maintenance, the methods highlighted in Figure 10 should to be tested.

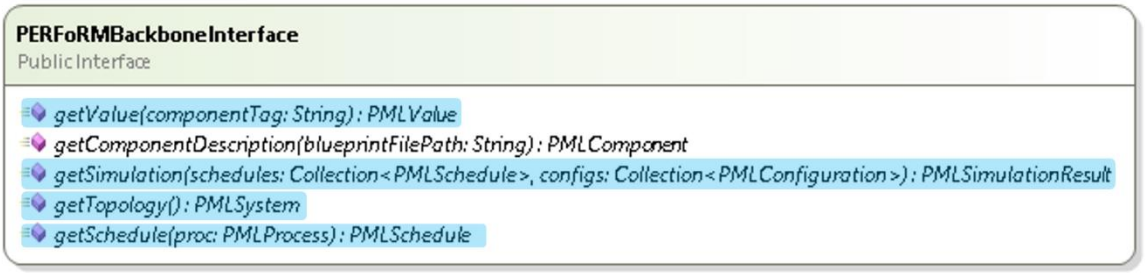


Figure 10: Backbone Methods to be tested for the Siemens Scenario

The *getValue* method is required by each of the tools to acquire specific data, in order for them to perform their respective tasks. On the simulation side, *getTopology* serves as the means to obtain the current system state, thus enabling the simulation tasks to be executed. The *getSimulation* is used to

request the start of a new simulation run, thus acting as the main trigger for PlantSimulation, and finally *getSchedule* serves a similar role, interfacing instead with the dynamic scheduling tool.

In terms of the lower-level machinery interface, the following methods represented in Figure 11 are of relevance:

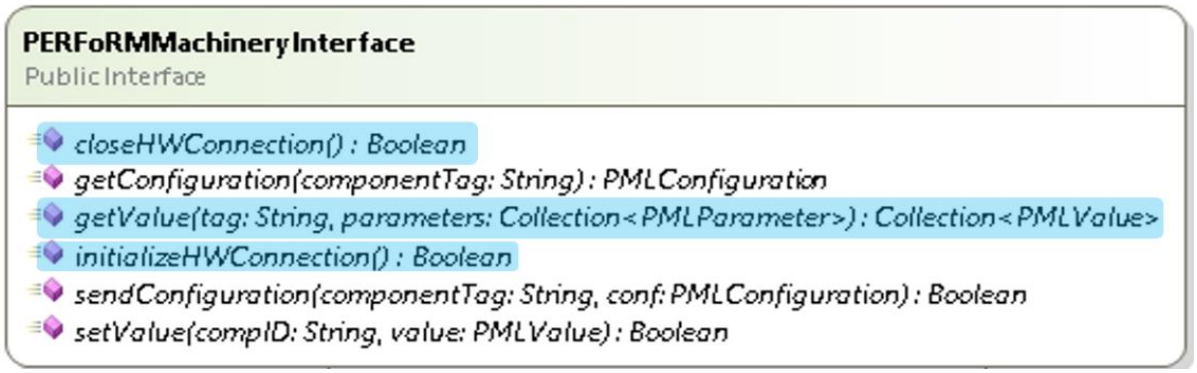


Figure 11: Machinery Methods to be tested for the Siemens Scenario

InitializeHWConnection and *closeHWConnection* should be used to open and close the connection to the machinery level, respectibely. The *getValue* method should be used to complement its counterpart in the backbone layer, thus enabling the acquisition of data from the machinery level.

Adapters

Legacy systems considered in the Siemens scenario for the implementation of the adapters are summarized in Table 3 with a focus on this task, the IT-Level. ERP is the standard enterprise resource planning software developed by SAP, BDE is an Oracle database containing production data (machines status and occupancy times), LHnet is a MS SQL Server database containing reports about machine failures and breakdowns.

Table 3: Siemens Legacy IT-Systems

Use Case	Objective	Legacy Systems
Siemens Compressors	Integration of a predictive maintenance system	➢ ERP System (SAP APO) ➢ MDE/BDE Data Logging System (Oracle DB) ➢ LHnet Ticketing System (MS SQL Server DB)

For the two databases (MDE/BDE – Oracle; LHnet Ticketing System- MS SQL) different adapters must be implemented because of the difference between the languages used by the two RDBMS. Although both systems use a version of SQL, MS SQL Server uses Transact-SQL (T-SQL), which is an extension of SQL used by Microsoft; Oracle, meanwhile, uses Procedural Language/SQL (PL/SQL).

The DB adapter permits to connect this database to the PERFoRM middleware. In particular, the activities of the DB Adapter consist of the following steps:

1. Implement the PERFoRM's standard interface and methods
2. Connect to the database using the proper driver
3. Send queries and update statements to the database
4. Retrieve the results from the database

5. Present the results according to the PERFoRMML data model

More details about the implementation of the adapter can be found in the Deliverable 3.1. Further modifications/customizations of the DB adapter for the Siemens use case will be tackled into the corresponding work package (WP7). In particular, Loccioni will implement an Oracle DB adapter for connecting the BDE Data Logging System, while PARO implements a MS SQL Server DB adapter for connecting the LHnet Ticketing System.

Table 4: Siemens Adapter - Key Features

ID	Feature Description
F-S-A-1	Adapter enables the middleware to query and update the MDE/BDE database (Oracle)
F-S-A-2	Adapter enables the middleware to query and update the LHnet databse (MS SQL Server)
F-S-A-3	Adapter translates the database information to the PERFoRMML data model

4.3.2. Simulation

Simulation Environment

The Simulation Environment is a tool that can be used to make predictions about the performance of the overall factory system. Given the current state of the factory (machinery, product orders, available workers,…) the Simulation Environment automatically generates a simulation model or parameterizes an existing one and then evaluates the production system configuration based several system-level KPIs for the production system – examples include: # products completed, average time to complete a product, # of unproductive hours, % of products delivered past deadline, average delay in product delivery…

For the Siemens scenario, the Simulation Environment's interfaces provides the following functionalities which must be tested in the test scenarios described above:

Table 5: Siemens - Simulation Environment Feature Description

ID	Feature Description
F-S-SE-1	Simulation Environment can accept a command via standard interface
F-S-SE-2	Simulation Environment can create/execute simulation of reconfigurable system
F-S-SE-3	Simulation Environment can send command via standard interface
F-S-SE-4	Simulation Environment can interpret command from Scheduling System
F-S-SE-5	Simulation Environment can send comparison of schedules to Schedule Selection tool

F-S-SE-1 Simulation Environment can accept command via standard interface

As defined in D2.3 [D2.3], the PERFoRMBackboneInterface includes the *getSimulation* method that "Requests a simulation of the system to be performed using a given set of schedules and configurations. Receives as an input two collections, one PMLSchedule elements and the other of PMLConfiguration elements. Returns the simulation results as a PMLSimulationResult object".

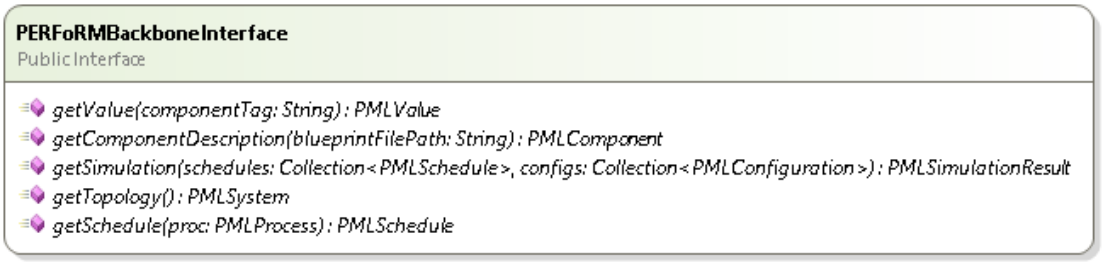


Figure 12: Backbone Interface

In this Test Item, this *getSimulation* method will be tested with respect to the ability to begin the execution of a simulation within the Simulation Environment. An example set of required inputs (the *PMLSchedule*, and the *PMLConfiguration*) will be provided as a direct input where the *getSimulation* method will be manually triggered.

For this Test Item, a successful test will require:

- The Middleware to be connected to the Simulation Environment
- The test inputs to be manually defined, and made available to the Middleware
- The Simulation Environment to accept the call from the standard interface
- The Simulation Environment to correctly accept the inputs included in the standard interface, ie. No data loss during transmission.

F-S-SE-2 Simulation Environment can create/execute simulation of reconfigurable system

Once feature F-S-SE-1 has been successfully tested, this Test Item will then focus on the internal capabilities of the Simulation Environment. In addition to testing performed within WP4.1 during the development of the Simulation Environment, this Test Item will provide the first functional testing of the simulation functionality.

For this Test Item, a successful test will require:

- A successful completion of Test Item F-S-SE-1
- Plant Simulation to be correctly installed and configured within the Simulation Environment
- The Simulation Environment to correctly convert the PML-conformant inputs into the simulation neutral format developed within WP4.1.
- The Simulation Environment's expandable model library to have been populated with base classes defining the machinery, products, workers, and other aspects present at the SmartFactory test track.
- The Simulation Environment to correctly open the base library model as defined in the last point within Plant Simulation.
- The Simulation Environment to update the model status given the simulation inputs stored within the simulation neutral format.
- The Simulation Environment to correctly execute the developed simulation.
- The Simulation Environment to correctly identify termination of the simulation, and to compile simulation results into a human-readable format. During this Test Item, the results of the simulation will not be transferred to the Middleware for further use.

F-S-SE-3 Simulation Environment can send command via standard interface

Once Test Item S-E-1.1 has been successfully executed, this Test Item will then focus on the capability of the Simulation Environment to pass the results of a simulation back to the Middleware. As defined in D2.3 {cite D2.3}, the *getSimulation* method returns a *PMLResult*. In this Test Item, a manually defined *PMLResult* will be developed and provided the Simulation Environment to be exchanged via the Standard Interface.

For this Test Item, a successful test will require:

- A successful completion of Test Item F-S-SE-2
- The test outputs to be manually defined, and made available to the Simulation Environment
- The Simulation Environment to call its internal method *returnSimulationResults* which will pass the manually defined outputs to the Middleware
- The Middleware to correctly accept the test outputs sent via the standard interface, ie. No Data loss

F-S-SE-4 Simulation Environment can interpret a command from Scheduling System

The key output of the Scheduling System is a (set of) schedule defining the order and timing of the maintenance and production tasks to be completed. This test will focus on showing that the linked tools (Scheduling System and Simulation Environment) both utilize a common implementation of the PERFoRMML specified *PMLSchedule* and *PMLConfiguration*, as defined in Deliverable D2.3 {cite D2.3} and shown below. As was described in F-S-SE-1, the standard interface method *getSimulation*() will be used here, with the additional level of detail that schedule sent will be that as generated by the Scheduling System.

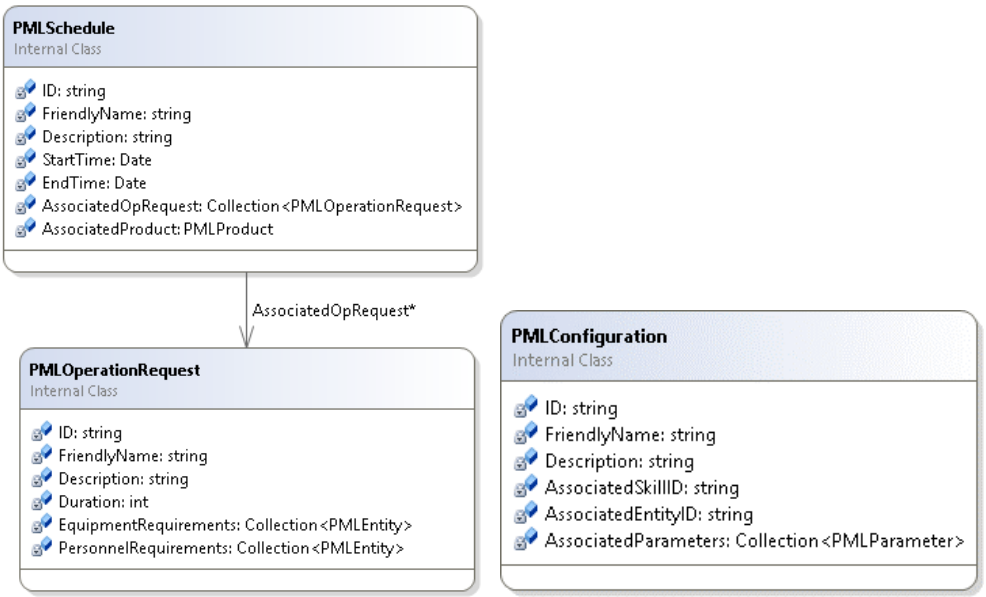


Figure 13: PMLSchedule / PMLOperationRequest and PML Configuration

For this Test Item, a successful test will require:

- A successful completion of Test Items F-S-SE-1, F-S-SE-2, F-S-SE-3
- The Scheduling System to provide an output *PMLSchedule* and *PMLConfiguration* (either defined manually within the Scheduling System, or the result of an actual execution)
- The Simulation Environment to be automatically triggered with these inputs.

- The Simulation Environment to accept the call from the standard interface
- The Simulation Environment to correctly accept the inputs included in the standard interface, ie. No data loss during transmission.
- The Simulation Environment to correctly convert the PML-conformant inputs into the simulation neutral format developed within WP4.1.
- The Simulation Environment's expandable model library to have been populated with base classes defining the machinery, products, workers, and other aspects present at the SmartFactory test track.
- The Simulation Environment to correctly open the base library model as defined in the last point within Plant Simulation.
- The Simulation Environment to update the model status given the simulation inputs stored within the simulation neutral format.
- The Simulation Environment to correctly execute the developed simulation.
- The Simulation Environment to correctly identify termination of the simulation, and to compile simulation results into a human-readable format. During this Test Item, the results of the simulation will not be transferred to the Middleware for further use.

F-S-SE-5 Simulation Environment can send a comparison of schedules to Schedule Selection tool

Once the prior Test Item has completed, the final Test Item to be considered regards the capability of the Simulation Environment to correctly send *PMLResult* responses to the Middleware that conform to the same standard as the Schedule Selection Tool.

For this Test Item, a successful test will require:

- A successful completion of Test Item F-S-SE-4
- The Simulation Environment to provide an output *PMLResult* (either defined manually or the result of an actual execution)
- The Simulation Environment to call its internal method *returnSimulationResults* which will pass the *PMLResult* to the Middleware
- The Middleware to correctly accept the test outputs sent via the standard interface, ie. No Data loss
- The Middleware to correctly pass the *PMLResult* to the Schedule Selection tool.
- The Schedule Selection to correctly accept the *PMLResult* sent via the standard interface, ie. No Data loss

Simulation of Maintenance Activities

Based on specific input data, the simulation considers the effects of foreseeable machine breakdowns to the remaining production schedule. The simulation calculates final KPI´s for a list of possible manufacturing schedules. Based on the final KPI´s it is possible to make a ranking of the manufacturing schedules. Because of a generic structure, the simulation is applicable for every use case if all required input data in the right data model are available for the simulation. With the aid of Monte Carlo experiments, the simulation varies in a given range the KPI limits for machine breakdowns and calculates e.g. the probability, if the considered production schedule meets all delivery dates from the customer orders.

Table 6: Siemen Simulation of Maintenance Activities – Feature Description

ID	Feature Description
S-SM-01	*Get the schedule and machine status that should be tested from the middleware (the arguments in getSimulation() and* getStatus()*)*
S-SM-02	*Automatically generate (and run) generic simulation models for the requested schedule/configuration for different factory configurations*

4.3.3. Planning Logic

Multiobjective scheduling of production orders and maintenance tasks

The tool "Multiobjective scheduling of production orders and maintenance tasks" is responsible to define a new scheduling, combining the production tasks with the maintenance tasks that need to be performed.

This task is being developed for the Siemens use case, to allocate the production and maintenance tasks for the three Carnaghi machines. Nowadays the maintenance tasks are managed separately from the production management and this constitutes a problem because most of the time the maintenance tasks allocation do not take in count the optimization of the production. Hence, the tool must combine two needs (Production and Maintenance) in order to daily deliver a schedule for the three machines and the maintenance teams.

The tool is automatically triggered by itself. Every day, before starting production, the tool will merge the production objectives and the maintenance tasks that need to be performed and generate a new schedule.

Table 7: Siemens Scheduling Tool – Feature Description

ID	Feature Description
S-PL-1	Receives Description of the production system and operations that need to be performed via the standard interface from the middleware.
S-PL-2	Can send three different possible schedules based on different objectives to the middleware via the standard interface which are consumed by the simulation tool. 1. Solve the maintenance issues as soon as possible 2. Solve the maintenance problems as late as possible 3. An optimized solution combining the production and the maintenance tasks
S-PL-3	Can interact with the middleware through the standard interface to receive relevant information from the data analytics tool. E.g. Time to be performed, Possible problem in the future.

4.3.4. Decision Support

Min-Max Data Mining Toolbox

This tool is adding capability to the machine for the measurement of machines (vertical turning process) current, voltage, phase shift and harmonics of the whole machine in combination with maintenance information from MDE and maintenance personnel to predict the machines maintenance demand down to component level. Prediction is carried out by feature selection of signals and using neural networks and Bayes-Models on clearly defined machine states to detect abnormal behavior.

The key features which are provided by this tool are summarized in the table below:

Table 8: Siemens Data Mining Toolbox – Feature Description

ID	Feature Description
S-MM-1	Output of change in components behavior regarding to previously defined component specific failures. Defining of thresholds to detect a breakdown of a component regarding a specific failure.
S-MM-2	Can send three different possible schedules based on different objectives to the middleware via the standard interface which are consumed by the simulation tool. 1. Solve the maintenance issues as soon as possible 2. Solve the maintenance problems as late as possible 3. An optimized solution combining the production and the maintenance tasks

Data Mining

This tool is going to use data mining techniques to predict the failure probabilities: It will collect the Alarms and maintenance data from MDE/BDE and LHnet database of Siemens via the middleware and probably using machine sensors data (not yet defined and confirmed by the time of this deliverable). The data mining techniques will be applied on the historical data to build a predictive model to predict the future failure using the current Alarms and/or Sensors data. The outcome of the predictive model will be visualized by the web

Table 12 lists the key features of this tool:

Table 9: Data Mining– Feature Description

ID	Feature Description
S-DM-1	Provides a model to predict the future maintenance needs
S-DM-2	Provides a decision support tool to visualize the historical data and prediction outcomes
S-DM-3	Connects to the middleware for collecting the data from the MDE/BDE and the LHnet databases of Siemens
S-DM-4	Sends the prediction outcome to the middleware which will be received by the Scheduling tool.

Universal Web-based KPI visualization

This tool is written in JavaScript and implements a browser based, universal visualization tool with the ability to add different analysis possibilities, calculation modules and plot windows based on d3s.js.

Table 10: Data Mining Toolbox – Feature Description

ID	Feature Description
S-V-1	Can receive data from the middleware via MQTT, Websockets and OPC-UA based on the PERFoRM standard interface
S-V-2	The user interface, entities and the data to be displayed is remotely configurable by PML

Bayesian Diagnostics & Prognostics for Manufacturing Equipment

The Bayesian Analytics Tools for Siemens Case study rely on the analysis of the maintenance data, the alarm messages from the PLCs and production plan. Based on that the developed tool will be able

to isolate the fault part and provide an estimated failure sub-system based on the current state of the machine and the production plan. All features are summarised as shown in Table below.

Table 11: Siemens Bayesian Diagnostics & Prognostics – Features Description

ID	Feature Description
F-S-BD-01	Extract alarm messages from the PLC on regular bases (every hour)
F-S-BD-02	Production plan with part numbers, tools and process (milling/ drilling)
F-S-BD-03	Isolate faulty sub-system
F-S-BD-04	Show the probability for failure in each sub-system

5. Whirlpool – Test Scenarios Description

The Whirlpool use case focuses on the challenge to solve the lack of available real-time shop floor data and its integration into the production system controlling and planning.
In fact, Whirlpool use case aims at establishing a real-time monitoring system able to correlate the dynamic behavior of the factory to its Key Performance Indicators (KPI) and static Key Business Factors (KBF), and enable its fast reconfiguration.

5.1. Test Scenario Description

Whirlpool information system is composed of four main technologies, operating on different levels and interacting with each other by exchanging data and information. In this section, this system is shown and the connections among the technologies are described, using the following picture and tables.

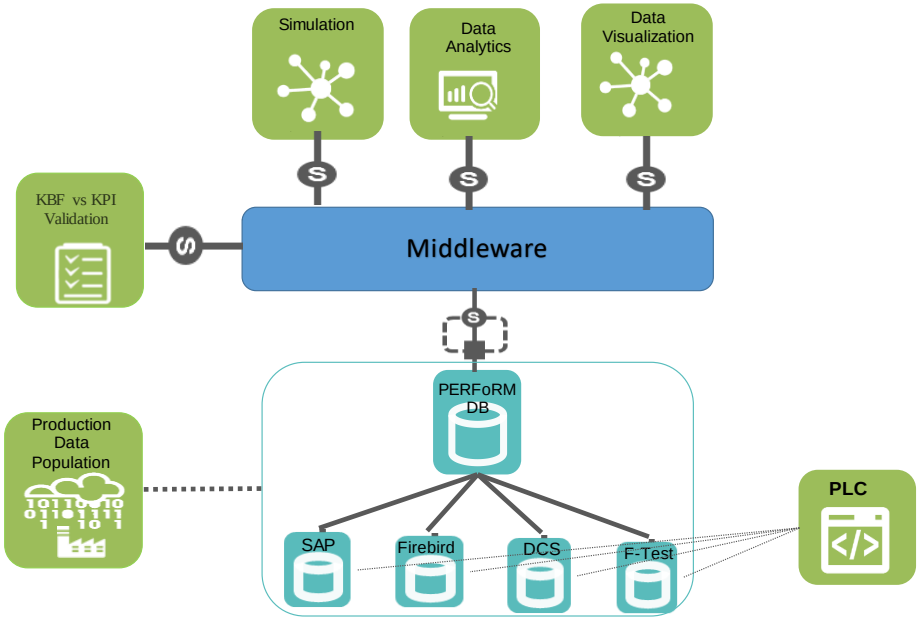


Figure 14: Whirlpool Manufacturing Information System Architecture: relationships among different technologies and different levels

From this representation, it is possible to figure out the relationships among the different technologies and modules involved within the Whirlpool architecture. In particular, the picture shows how the interaction among the IT-Modules is performed. The test scenarios and the description are summarized in the table below within the scope of T6.2.

Table 12: Whirlpool - Test scenario description

ID	Test Scenario description
W-F-01	***Interface Middleware – Adapter – PERFoRM Database (SQL):*** *Test that the PERFoRM database (MS-SQL) can be accessed by the middleware via the adapter*
W-F-02	***Interface Simulation to middleware:*** *Test that the Simulation tool can share information between the middleware with the PERFoRM standard interface.*
W-F-03	***Simulation access database via the middleware:*** *Test that the Simulation can access relevant information with are necessary for the simulation activities via the PERFORM Standard interface*
W-F-04	***Interface between the Data Analytics tool and the middleware:*** *Test that the data analytics tool can send a request to the middleware via the PERFORM interface*
W-F-05	***Collaboration between the Data Analytics and the simulation tool:*** *Test that the Analytics can receive the simulation results via the middleware in order to aggregate them.*
W-F-06	***Interface between the Visualization tool and the middleware:*** *Test that the visualization can receive necessary data from the middleware*
W-F-07	***Collaboration between the Visualization and the Analytics:*** *Test that the visualization tool can receive analysis results via the middleware in order to visualize them,*
W-F-08	***Collaboration between the Visualization and the Simulation:*** *Test that the visualization tool can receive simulation results via the middleware in order to visualize them.*

5.2. Technology Scope

In the use case of Whirlpool, the following technologies are in the scope to be tested or demonstrated with the test scenarios described above:

- The **Adapter** to the PERFoRM MS-SQL database which provides an interface to the middleware for accessing the data.
- The **PERFoRM Database** (SQL) with offline data.
- The capability of the **Middleware** to route the information to the involved components.
- The **PERFoRM Standard Interface** between the Middleware and the solutions used in this use case
- The **Tool-Solutions**
  - Excel-based KPI Value Stream Model
  - *Excel-based KPI* Simulation
  - KPI Monitoring and Visualization Tool
- The **Collaboration** between the used solutions.
- The **Information Model** which provides the exchange format between the Middleware and the Tools and the database.

5.3. Description of the components and features to be tested

5.3.1. Architecture Technologies

Middleware

In the Whirlpool Use Case, the Middleware is used for similar purposes as described in 4.3.1. The main difference is that different systems which have to be linked up in this use case. The main requirement for Whirlpool is to be able to retrieve production data from one specific database, which includes all relevant production data. Furthermore, simulation, data analytics and data visualization tools need to be integrated and be enabled to exchange data.

Table 13: Whirlpool Middleware - Key Features

ID	Feature Description
F-W-M-1	Middleware enables the data transfer from the PERFoRM DB to the Simulation tool
F-W-M-2	Middleware enables the data transfer from the Simulation tool to the Data Analytics tool
F-W-M-3	Middleware enables the data transfer from the Data Analytics tool to the Data Visualization tool
F-W-M-4	Middleware enables the data transfer from the Simulation tool to the Data Visualization tool

Standard Interface

Regarding the standard interface between the tools seen in Figure 15 and the middleware, since the focus is centered mainly on data acquisition and simulation, the methods highlighted in Figure 15 are particularly relevant to be tested.

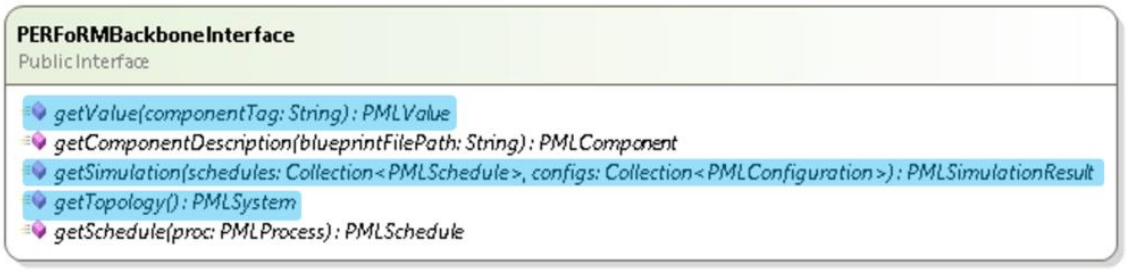


Figure 15: Backbone Methods to be tested for the Whirlpool Scenario

The *getValue* method is required by each of the tools to acquire specific data (from both the PLM repository and the shop-floor), in order for them to perform their respective tasks. On the simulation side, *getTopology* serves as the means to obtain the current system state, thus enabling the simulation tasks to be executed. Finally, *getSimulation* is used to trigger the start of a new simulation run, hence being used in the optimization tasks.

In terms of the lower-level machinery interface, the following methods represented in Figure 16 are of relevance:

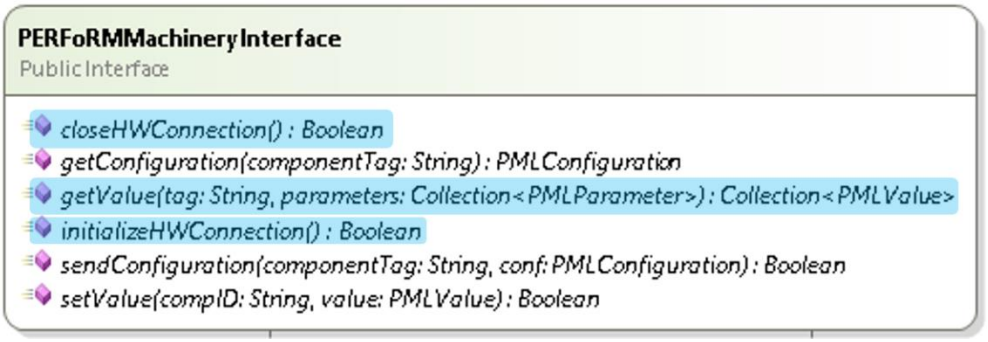


Figure 16: Machinery Methods to be tested for the Whirlpool Scenario

InitializeHWConnection and *closeHWConnection* should be used to open and close the connection to the machinery level, respectibely. The *getValue* method should be used to complement its counterpart in the backbone layer, thus enabling the acquisition of data from the machinery level.

Adapters

Legacy systems considered in the Whirlpool scenario for the implementation of the adapters are a database containing data for KPIs calculation, the PLM software and the robot used for the leakage test.

Table 14: Whirlpool Legacy Systems

Use Case	Objective	Legacy Systems
Whirlpool Microwave Ovens	➢ Implementation of a KPI real-time monitoring system ➢ Reconfiguration of the path of the robot for the leak test	➢ PERFoRM DB (MS SQL Server DB) ➢ PLM Repository (txt file) ➢ Leak Robot Station (UR10 Controller)

In order to reduce the number of adapters needed and select the legacy systems which are most important, a questionnaire has been prepared and completed by the end-users. The questionnaire takes into account the following advantages which can be achieved by the integration of the legacy system through an adapter:

- Improve Cost Effectiveness
- Reduction in Obsolescence Risk
- Improve Overall User Satisfaction
- Improve Overall Systems Capability
- Enhance System of Systems (SoS) Integration

According to the results of the questionnaires, for the Whirlpool scenario the legacy systems which have the higher scores are the PERFoRM DB and the Leak Robot Station. For these two production resources two different adapters must be implemented.

The PERFoRM DB is a MS SQL Server DB which contains the data for the calculation of the KPIs to be monitored. The DB adapter permits to connect this database to the PERFoRM middleware. In particular, the activities of the DB Adapter consist of the following steps:

1.Implement the PERFoRM's standard interface and methods

2.Connect to the database using the proper driver

3.Send queries and update statements to the database

4.Retrieve the results from the database

5.Present the results according to the PERFoRMML data model

Table 15: WHR Adapter - Key Features

ID	Feature Description
F-W-A-1	Adapter enables the middleware to query and update the PERFoRM-DB (MS SQL DB)
F-W-A-2	Adapter translates the database information to the PERFoRMML data model

5.3.2. Simulation

Finite State Automata – KPI Excel-based Simulation

The KPI-Excel based Simulation tool is a structured support tool for the measurement, the analysis and the improvement of manufacturing performances through the design and use of a Finite State Machine–based system. This tool is able to identify all temporal/manufacturing states which can be observed during operation and which can impact on production inefficiency (i.e. time losses, quality losses and performance losses). Structured methodology for the measurement and the analysis of OEE based on finite state machine can be applied, replicating the production process with standard functional blocks in order to model machine behavior to define the operation execution time and to calculate buffer capacity.

Table 16: KPI Simulation - Key Features

ID	*Feature Description*
F-W-KS-1	An Excel-based tool will be configured in order to perform simulation activities on Cassinetta production plant according with Whirlpool Use Case. Data collection can be carried out directly from field (Industrial PLCs) or from PERFoRM Data Base.
F-W-KS-2	Finite states machine will be analyzed to evaluate time based performances by analyzing Finite states machines.
F-W-KS-3	Supports Finite State machines to carry out a product based analysis, evaluating the overall product lead time for each product and calculating how different item (product flexibility) can impact on production efficiency.

5.3.3. Decision Support

Excel-based Value Stream Model and KPI Simulation

The Excel-based Value Stream Model is in conjunction with the KPI Simulation a Real time monitoring system, based on Value Stream Mapping (VSM), which is able to correlate dynamic behaviour of the factory to both its Key Performance Indicators (KPIs) and the static indicators such as Key Business Factors (KBF). The correlation between those indicators and the visualization of KPIs is one of the main requirements from Whirlpool.

Table 17: Value Stream Model - Key Features

ID	Feature Description

F-W-VS-1	Validation process will be carried out testing the decision maker capability to select and to adjust a new set of KBFs in order to meet the process optimization.
F-W-VS-2	The reconfiguration solutions will be tested to identify the corrective actions to be applied at shop floor level (i.e machine disposition, dispatching management etc.) in order to improve the overall production activity.
F-W-VS-3	This tool has a strategic position within Whirlpool Work Flow as it is responsible of closing the loop control system. In fact, it is designed to achieve and maintain the desired output (KPIs provided by Simulation, Data analytics and Visualization tools) by comparing it with actual condition (KBFs).

Automatic Monitoring and Visualization of KPIs

The Automatic Monitoring and Visualization of KPIs is based on XETICS LEAN which provides automatic data collection on production and packaging lines. It captures short interrupt events, freeing line operators from manual data collection tasks, and provides immediate feedback on unexpected or trend conditions. Clear indications of line bottlenecks, visibility into real-time performance metrics, and drill down analysis tools enable stakeholders to have useful information to effectively increase line performance and operational efficiency.

Furthermore it provides real-time monitoring of production lines and combines with tracking of equipment utilization for a full line performance solution. Tracking work order execution provides an adequate categorization of unplanned downtime versus product change or downtime or other scheduled events.

Table 18: WHR Xetics Monitoring and Visualization - Key Features

ID	Feature Description
F-W-XM-1	Implemented KPI service which stores all calculated KPI gathering from events like movein/moveout/sensor/statechanges and storing in own table/database

KPI Monitoring with What-if-game Functionality Tool

The KPI Monitoring with What-if-game Functionality Tool is a Web based visualization tool to support decision-making strategies, by KPI monitoring and by performing what-if-game based on the variation of several KBF. Main features are: visualization and monitoring of KPIs, detection of trends and deviations and what-if-game capability.

Architecturally the tool is composed of the following modules: Data Service Module, KPI Calculation & Statistical Quality Control Module, What-if-game Module, Data Handler Module and Visualization Module (represented in Figure 10).

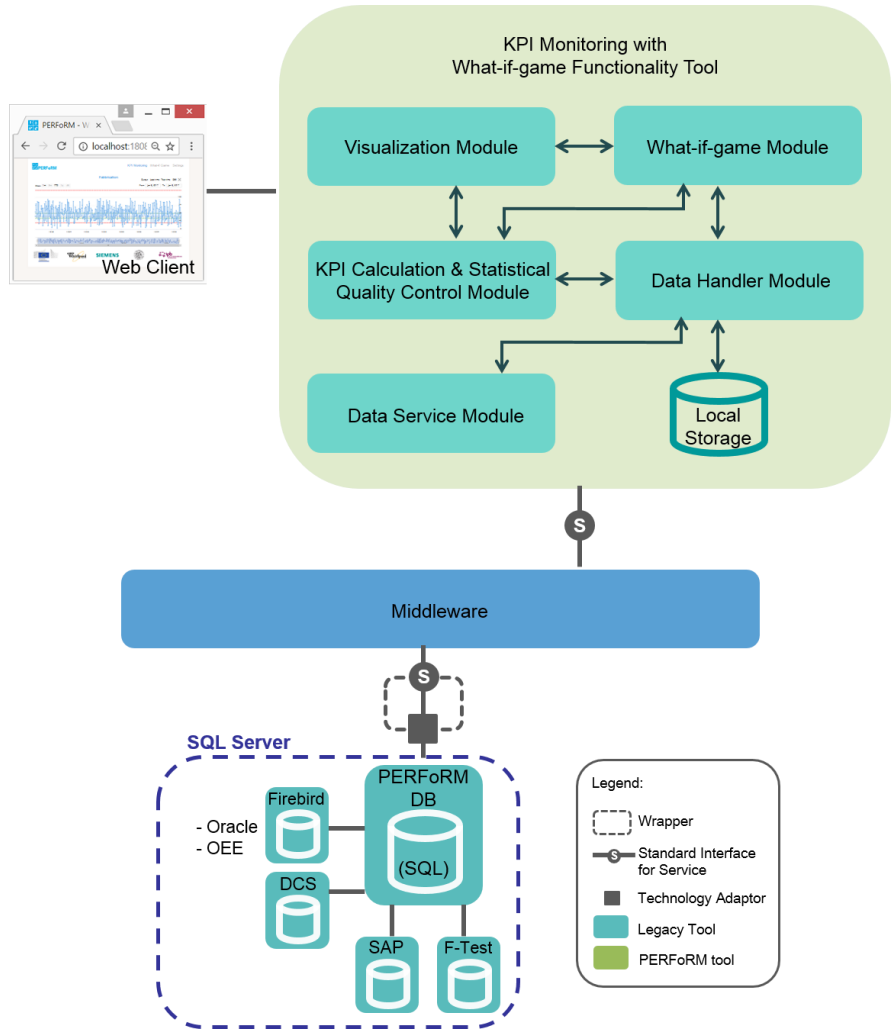


Figure 10: Tool architecture

The Data Service Module, operating on a request–reply or publish–subscribe pattern basis, is responsible for receiving data via middleware (generic KBFs and by process). Remote and local data is required to support the continuous operation of the KPI Calculation & Statistical Quality Control Module. To handle these needs the Data Handler Module manages the local repository and evaluates whether there is a need to trigger data acquisition via the Data Service Module. The KPI Calculation & Statistical Quality Control Module is responsible for the calculation of KPIs using KBFs. It's also responsible for the statistical quality control of the temporal evolution of the KPIs. The What-if-game Module uses internal info and the KPI Calculation & Statistical Quality Control Module to generate what-if games with the KBFs vs KPIs. Finally, the Visualization Module renders the web pages to be presented to browsers directly accessing the tool. Table 19 summarizes the features of the tool to be tested in T6.2.

Table 19: KPI Monitoring – Feature Description

ID	Feature Description
W-KPI-1	Receives Data from the Middleware via MQTT standard interface including the PML description
W-KPI-2	The tool displays the KPIs after processing the PERFoRM DB

The tests to be performed within T6.2 are devised to test the tool connectivity to the needed data sources. Particularly, the testing of the tool to the MW and to the data sources will be performed. The first, will be performed using a MQTT publish/subscription mechanisms that will then inform the KPI monitoring tool of changes in the relevant data values (KBFs), namely *Working Day*, *Hour/shift*, *Shift/ Day*, *Total Demand*, *Cycle Time*, *Op-Mach*, *NC* (non-conforming product), *Availability*, *Performance*, *Price*, *Variable Cost*, *Fix Cos*t, *Margin*, *Set up time*, *Batch size*, *Op/set up*, *Operator Cos*t, *Dim trolley*, *Pz/trolley*, *Stackability*, *Interest rate*, *WIP Value*, *Occupation Cost*, *Overheads*.

6. GKN – Test Scenarios Description

In terms of PERFoRM project, GKN is developing an innovative reconfigurable robotic process cell that will maximize the flexibility and agility of the production process. The reconfiguration mechanism will be able to detect the modules that have been plugged in into the cell and will connect those information with new software solutions that will allow data visualization, simulation mechanisms and optimization of the production schedule (please see Figure 17) An OPC-UA layer will allow the communication between the IT level and the asset level. Interfaces that will enable the communication among the different PLCs and the middleware solution will be used, if required.

This chapter describes the test and demonstration scenarios for GKN's use case, as this is described in D10.1 [D10.1].

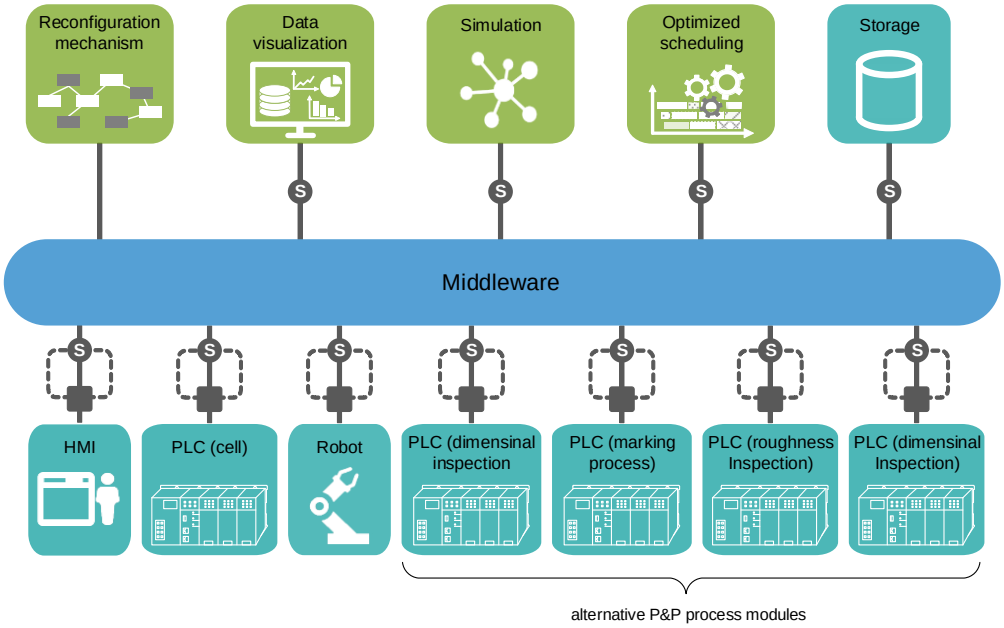


Figure 17: GKN System Architecture

6.1. Test Scenario Description

All the new software solutions that are going to be implemented in to GKN's IT infrastructure level will be tested and demonstrated. All the related test scenarios can be seen on Table

Table 20: GKN - Test scenario description

ID	Test Scenario description
G-F-01	*Verify that the robot program can access OPC-UA/ cell Middleware*
G-F-02	*Verify that the interaction between the ERP, SPS and OEE systems can be done*
G-F-03	*Verify that the data from the ERP, SPS and OEE systems can be transferred to the OPC-UA/ cell middleware through adaptors*
G-F-04	*Verify that the reconfiguration tool works properly*
G-F-4.1	*Verify that the reconfiguration tool can detect the plugged modules (simulated data)*
G-F-4.2	*Verify that the reconfiguration tool can detect when a new process module is being plugged into the cell (simulated data)*
G-F-4.3	*Verify that the agents can access OPC-UA*

6.2. Technology Scope and Test Items

For the use case of GKN and the scope of T6.2, the following technologies are in the scope of the testing and demonstration activities.

- The **Adapter** between the PLCs and the middleware, which is in this case OPC-UA. Therefore the technology scope lies in the integration of legacy controller like PLCs to OPC-UA.
- The connection between OPC-UA and the **Middleware** solution.
- The **PERFoRM Information model to support** the reconfiguration on the cell level.
- The **PERFoRM Standard Interface** to the machinery level and to the IT-Level.
- The **Reconfiguration Tool** which detects the topology and interacts via OPC-UA with the robot and process modules.

6.3. Description of the components and features to be tested

6.3.1. Architecture Technologies

Middleware

Within this use case, the choice for the communication technology on the shop floor level is OPC-UA. A centralized database with OPC-UA interfaces is used to collect data from various connected machines. Therefore, a Middleware is not necessary for the different machines to communicate with each other. The reconfiguration tool is using OPC-UA as well and uses its discovery mechanisms to achieve the dynamic reconfiguration. Nevertheless, the various other tools to be implemented don't necessarily have an OPC-UA interface, making a Middleware necessary for any communication between these tools and the shop floor.

It is also necessary to allow the various management level systems (ERP, SPS, OEE) to get and transfer data to the data collection database. These systems are already using a Middleware solution (Microsoft biztalk), which needs to be adapted to allow communication through OPC-UA.

Table 21: GKN Middleware - Key Features

ID	Feature Description
F-G-M-1	Biztalk Middleware enables the data transfer between ERP, SPS and OEE systems
F-G-M-2	OPC-UA Biztalk component enables communication between ERP/SPS/OEE and cell database
F-G-M-3	PERFoRM Middleware enables the data transfer from the cell DB to the agent based simulation
F-G-M-4	PERFoRM Middleware enables the data transfer from the cell DB to the Simulation tool
F-G-M-5	PERFoRM Middleware enables the data transfer from the cell DB to the Planning & Scheduling tool
F-G-M-6	PERFoRM Middleware enables the data transfer from the cell DB to the Reconfiguration management tool

Standard Interface

Regarding the standard interface between the tools seen in Figure 18 and the middleware, since the main goals are related to reconfiguration, simulation and scheduling, the methods highlighted in Figure 18 are particularly relevant to be tested.

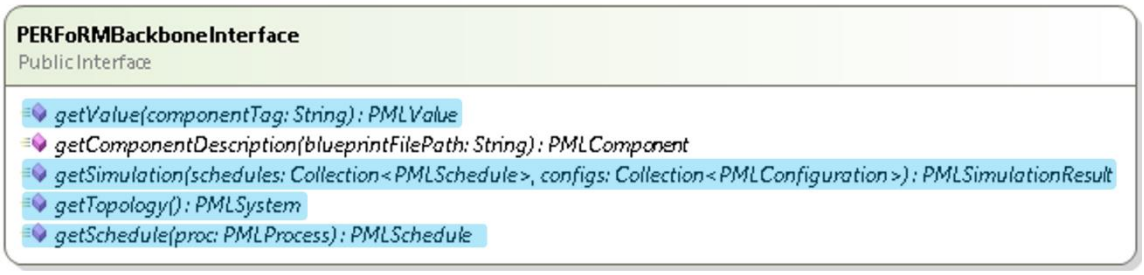


Figure 18: Backbone Methods to be tested for the GKN Scenario

The *getValue* method is required by each of the tools to acquire specific data, in order for them to perform their respective tasks. On both the simulation and scheduling side, *getTopology* serves as the means to obtain the current system state, thus enabling these tasks to be executed. The *getSimulation* method is used to trigger the start of a new simulation run, and finally *getSchedule* serves a similar role, interfacing instead with the scheduling tool.

In terms of the lower-level machinery interface, the following methods represented in Figure 19 are of relevance:

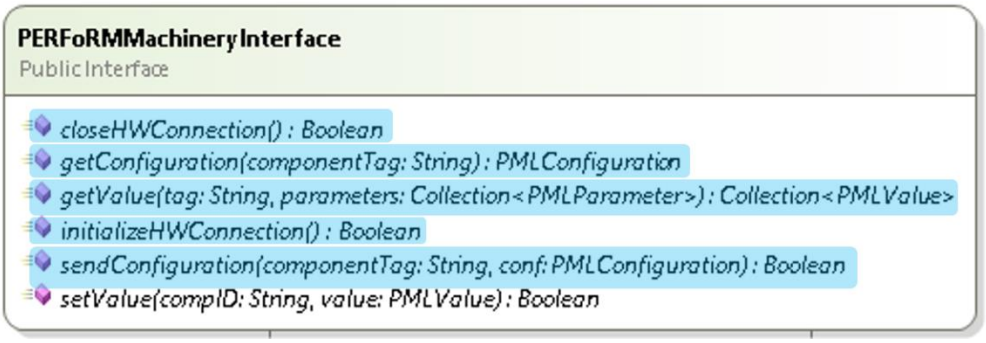


Figure 19: Machinery Methods to be tested for the GKN Scenario

InitializeHWConnection and *closeHWConnection* should be used to open and close the connection to the machinery level, respectibely. The *getValue* method should be used to complement its counterpart in the backbone layer, thus enabling the acquisition of data from the machinery level. Additionally, the *getConfiguration* and *sendConfiguration* methods support the reconfiguration tool, enabling it to get current machine configurations as well as to send new configurations as necessary, respectively.

Adapters

Legacy systems considered in the GKN scenario for the implementation of the adapters are the PLC controlling the robotic cell and the measurement sensor used for checking the roughness of the component.

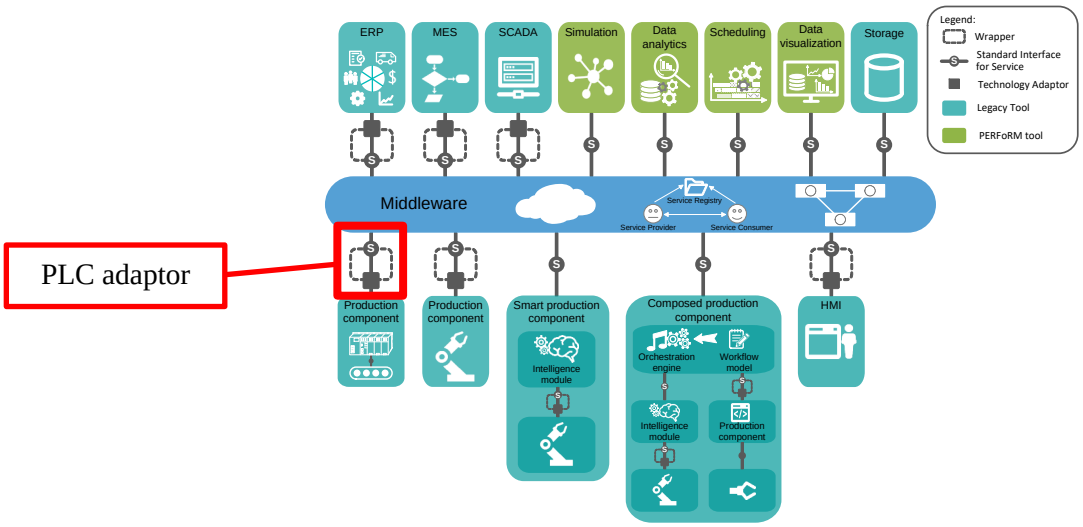
Table 22: GKN Legacy Systems

Use Case	Objective	Legacy Systems
GKN Turbine Vanes	Construction of a reconfigurable robotic cell	➢ Robotic Cell PLC ➢ Roughness Process (Mitutoyo

		SJ-210)

For these two legacy systems two different adapters must be implemented: PLC adapter and sensor adapter.

The PLC adapter permits to connect the PLC controlling the robotic cell to the middleware.



The PLC adapter is software based and communicates over a socket connection to the Siemens PLC. The communication to a S7-CPU is realized with an existing .Net library called S7.NET which is used for the adapter. With this library it is possible to send/receive a data structure consistent to a C# class with the same data structure. This communication is very fast and data safe. The adapter connects on the other side to the PERFoRM DB and communicates with the middleware standard interface. More details about the PLC adapter can be found in Deliverable 3.1.

The sensor adapter permits to connect the roughness sensor (Mitutoyo SJ-210) to the middleware. In order to enhance the reconfiguration capability of the GKN robotic cell, the adapter is able to substitute the existing wired serial communication between the roughness sensor and the PC with a WiFi connection. For this purpose, the adapter has hardware and a software part.

Regarding the hardware, a box containing a serial-WiFi converter, a battery and an electronic chip for recharging it, has been designed in order to make the converter free from any supply wires and the sensor released from the serial cable. The converter transforms the existing RS 232 communication in a TCP/IP one.

From the software side, the adapter implements an OPC-UA server which can be used to give commands to the roughness sensor and to read data from it. Besides any OPC-UA client can access the server, reading data provided by the sensor or controlling it.

The client can be created on the Middleware as a web service that interfaces with the OPC-UA server running on the PC.

The OPC-UA protocol allows exchanging information between Real Time systems and it is also very focused on data safety: communication between server and client can be permitted only if authorization certificates are provided.

Table 23: GKN Adapter - Key Features

ID	Feature Description
F-G-A-1	Adapter enables the middleware to query and update the PERFoRM-DB (MS SQL DB)

F-G-A-2	Adapter translates the database information to the PERFoRMML data model

6.3.2. Simulation

Agent Based Simulation (RMS and AMS in discrete event environment)

This tool provides a agent-based system simulation within an industrial discrete event simulation tool (e.g. Tecnomatix Plant Simulation). The Agent-based Simulation is used in the case of GKN to layout the production with micro-flow-cells and analyze their behavior (based on material and other flows) in production with focus on specific, to be defined, questions to verify the cell concept. The solution will be used "offline" in the current plan.

Table 6: Agent Based Simulation – Key features

ID	Feature Description
F-G-AS-1	Verification and layout of a flexible and reconfigurable manufacturing cell concept

Multi-objective planning & scheduling for dynamic flexible manufacturing systems.

The multi-objective planning & scheduling for dynamic flexible manufacturing systems is a tool that can be used to generate new scheduling plans for the dynamic flexible manufacturing systems in a dynamic system environment. Given the current state of the factory (the state of the target cell, machinery, product orders, available workers,…), the planning & scheduling tool can automatically generates a new schedule which optimise the objective function based on the current setting of the factory.

For the GKN scenario, the reconfiguration management tool's interfaces provide the following functionalities which must be tested in the test scenarios described above:

Table 24: GKN Multi-objective Planning & Scheduling - Key Features

ID	Feature Description
F-S-SE-1	Planning & scheduling can accept a command via standard interface
F-S-SE-2	Planning & scheduling can create/execute new optimised schedule of the dynamic manufacturing system
F-S-SE-3	Planning & scheduling can send command via standard interface
F-S-SE-4	Planning & scheduling can interpret command from Simulation Environment

Reconfiguration management in flex. manuf. sys.

The reconfiguration management in flexible manufacturing system is a tool that can be used to execute the reconfiguration management tasks for the target GKN cell. Given the current state of the factory (the state of the target cell, machinery, product orders, available workers,…), the reconfiguration management tool automatically generates a new schedule which optimise the objective function based on the current setting of the factory.

For the GKN scenario, the reconfiguration management tool's interfaces provide the following functionalities which must be tested in the test scenarios described above:

Table 25: GKN Reconfiguration management - Key Features

ID	Feature Description
F-S-SE-1	Reconfiguration Management can accept a command via standard interface
F-S-SE-2	Reconfiguration Management can create/execute new optimized schedule of reconfigurable system
F-S-SE-3	Reconfiguration Management can send command via standard interface
F-S-SE-4	Reconfiguration Management can interpret command from Simulation Environment

6.3.3. Planning Logic

Agent Based Reconfiguration Tool

The agent based reconfiguration tool consists in a logical re-organization of the micro-flow concept cell, as depicted in Figure 20.

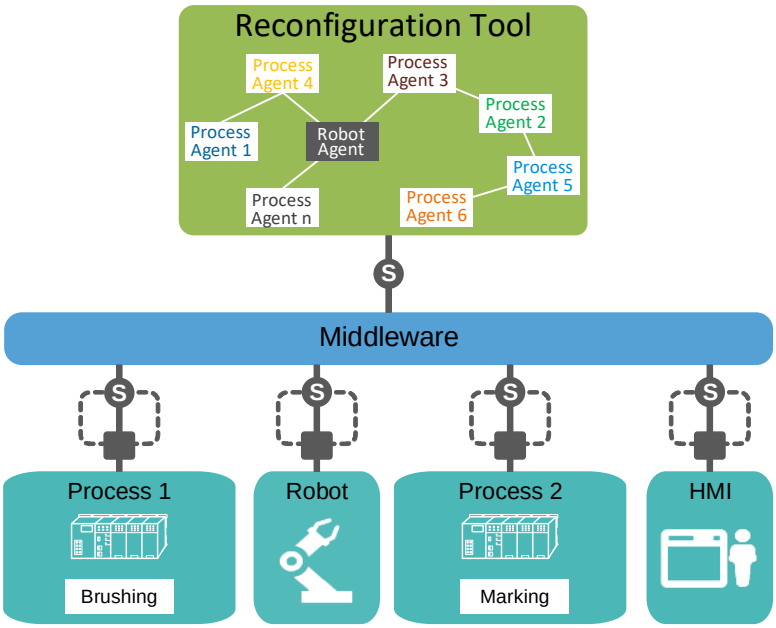


Figure 20: Overall picture

The agents are in control of the logical re-organization of the micro-flow-cell by means of getting data from OPC-UA Servers, through adapters, and exchange messages among themselves according to the FIPA (Foundation for Intelligent Physical Agents) message structure specifications.

Shortly, the MAS are composed by two types of agents, namely the “Robot Agent” and “Process Agent”. The robot agent is linked with the cell configuration and safety procedures, while the Process Agent represents the individual available processes.

At the beginning, the process agents are in *sleep mode*, switching to an active mode as soon as its associated process is plugged-in. At this stage, the Process Agent informs the Robot Agent of its new state, allowing, in this way, the robot agent to always know the cell configuration. The same happens when a process is plugged out.

The testing procedures to be developed in the present task focus on the plug-in and plug-out detection status by the MAS and the agents' ability to interface with the PERFoRM middleware.
Table 6 summarizes the features of the tool to be tested in T6.2.

Table 6: GKN Agent Based Reconfiguration – Key Features

ID	Feature Description
F-G-ABR-01	Agents read, write and subscribe a PMLValue through the middleware using OPC-UA technology.

7. IFeVS – Test Scenarios Description

I-FEVS information system is composed of different technologies, operating at different levels and interacting each other by exchanging data and information. In this section, this system is shown and the connections among the technologies are described, using the following picture and tables.

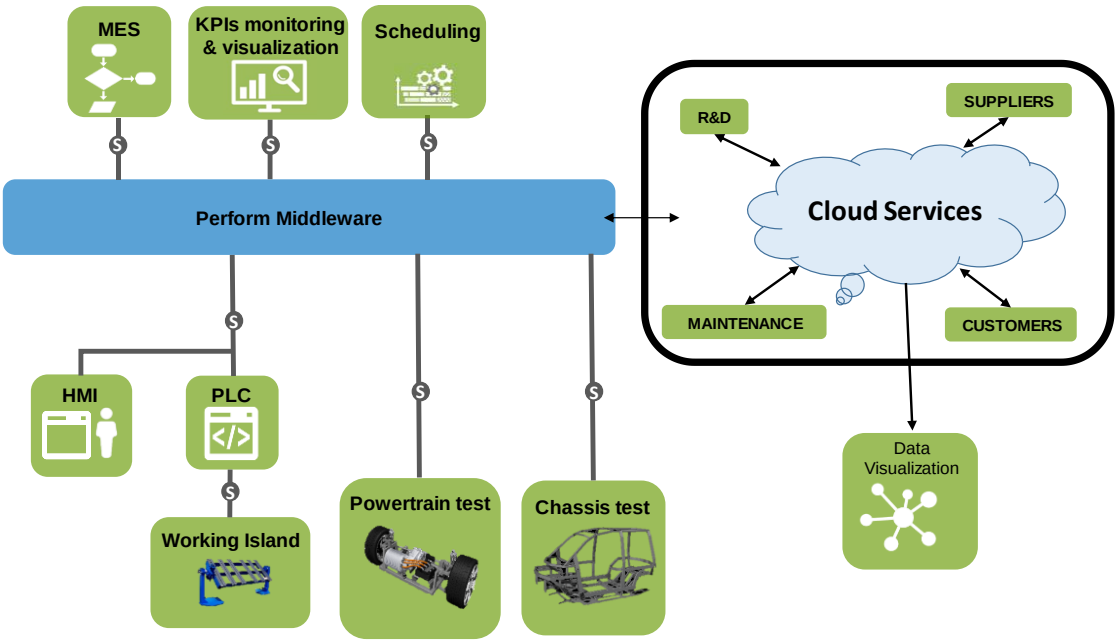


Figure 21: I-FEVS Manufacturing Information System Architecture: relationships among different technologies and different levels

From this representation, it is possible to figure out the relationships among the different technologies and modules involved within I-FEVS architecture. In particular, the picture shows how the interaction among these modules is performed.

7.1. Test Scenario Description

In the matrix reported below, the different sources and destinations of communication activities are listed down respectively on the left and on the top side. In particular, the intersection of a certain row and a particular column indicates the specific object (message, action, query etc.) resulting from the communication exchange.

The description of each connection is reported below with a focus on the IT-Level in the scope of T6.2.

Table 26: IFeVS – Test Scenario Description

ID	Test Scenario Description
IF-F-01	Summarizing and retrieving data from PLC
IF-F-02	Product identification is obtained through RFID and barcode that provide information about internal component production, their serial number, part-list and finished good
IF-F-03	Production Plan details are provided to perform the scheduling activities
IF-F-04	Display performance indicator in order to support decision maker activity
IF-F-05	KPI results are provided to each operator

IF-F-06	Each data is collected in the cloud in order to speed up the information procurement. Principally finished and semi-finished good declarations are communicated.

7.2. Technology Scope and Test Items

For IFeVS the technology scope has a focus on the following points for T6.2:

- The capability of the **Middleware** to route the information to the involved components.
- The **PERFoRM Standard Interface** between the Middleware and the solutions used in this use case
- The **Information Model** which provides the exchange format between the Middleware and the Tools and the database.
- The **Solutions** and **Interaction** with the Middleware:
  - KPI Monitoring and Visualization
  - Optimization and Rescheduling.

7.3. Description of the components and features to be tested

7.3.1. Architecture Technologies

Middleware

The Middleware foreseen for the E-District use case is enabling the communication between each involved component, similar to the ones described in 4.3.1 and 5.3.1. It needs to connect data from shop floor level systems, such as PLCs with the MES, the scheduling tool and the KPI monitor. Additionally, the aggregation of data is requested. In this specific case, also a link between the Middleware and external cloud services needs to be implemented.

Table 27: IFeVS Middleware - Key Features

ID	Feature Description
F-IF-M-1	Middleware is able to aggregate PLC data
F-IF-M-2	Middleware enables the data transfer from the shop floor systems (PLC, Powertrain test, Chassis test) to the MES
F-IF-M-3	Middleware enables the data transfer from the shop floor systems (PLC, Powertrain test, Chassis test) to the Planning & Scheduling tool
F-IF-M-4	Middleware enables the data transfer from the MES to the Planning & Scheduling tool
F-IF-M-5	Middleware enables the data transfer from the shop floor systems (PLC, Powertrain test, Chassis test) to the KPI monitoring tool
F-IF-M-6	Middleware enables the data transfer to the cloud

Standard Interface

Regarding the standard interface between the tools seen in Figure 22 and the middleware, since the IFEeVS use case focuses mainly the reconfiguration and scheduling activities, the methods highlighted in Figure 22 are particularly relevant to be tested.

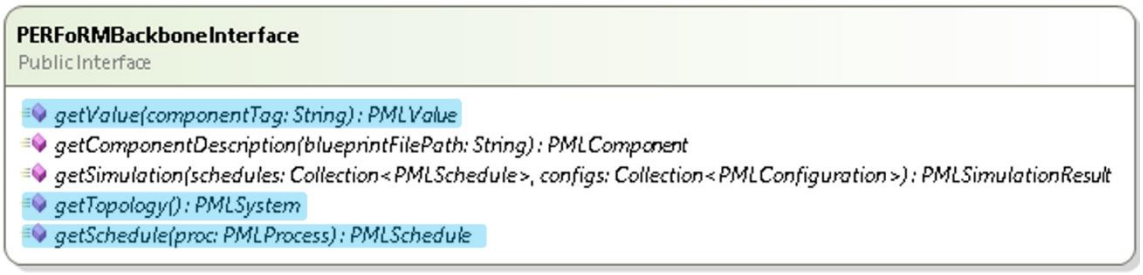


Figure 22: Backbone Methods to be tested for the IFeVS Scenario

The *getValue* method is required by each of the tools to acquire specific data, in order for them to perform their respective tasks. Moreover, *getSchedule* is used to trigger the request for a new schedule to be provided. As such, *getTopology* enables the scheduler to acquire the current system state, as deemed necessary by its internal operations.

In terms of the lower-level machinery interface, the following methods represented in Figure 23 are of relevance:

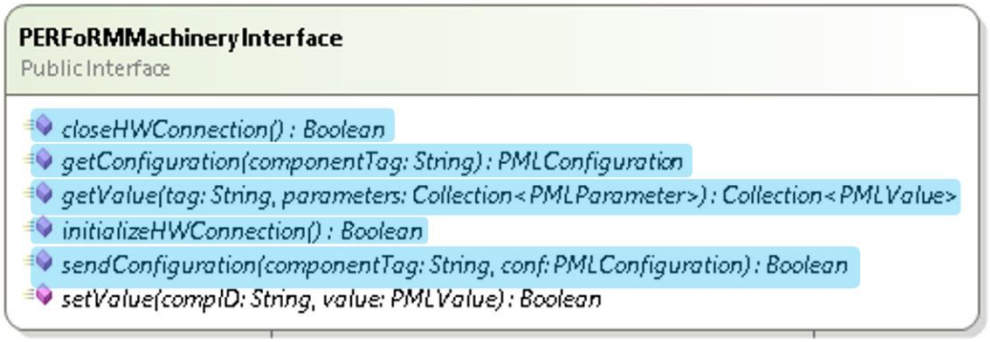


Figure 23: Machinery Methods to be tested for the IFeVS Scenario

InitializeHWConnection and *closeHWConnection* should be used to open and close the connection to the machinery level, respectibely. The *getValue* method should be used to complement its counterpart in the backbone layer, thus enabling the acquisition of data from the machinery level. Furthermore, the *getConfiguration* and *sendConfiguration* methods support the reconfiguration tool, enabling it to get current machine configurations as well as to send new configurations as necessary, respectively.

Adapters

Legacy systems considered in the IFeVS scenario for the implementation of the adapters are welding robotics cells as well as a powertrain testing station (rolling bench test to check the functionalities of the motorized axle frame) and a chassis testing station (geometrical test of the chassis to check if all the assembly complies with the design). Both robotic cells and testing stations are controlled by a Siemens PLC IM-151. In order to connect these systems to the MES a PLC adapter has been implemented.

The PLC adapter is practically identical to the one described under chapter 6.2.2 in this document. It is software based and communicates over a socket connection to the Siemens PLC. The communication to a S7-CPU is realized with an existing .Net library called S7.NET which is used for the adapter. With this library it is possible to send/receive a data structure consistent to a C# class with the same data structure. This communication is very fast and data safe. The adapter connects on the other side to the PERFoRM DB and communicates with the middleware standard interface.

Table 28: IFeVS Adapter - Key Features

ID	Feature Description
F-IF-A-1	Communication with a S7-CPU via S/.NET
F-IF-A-2	Reflects the data structure to a C# class

7.3.2. Planning Logic

Energy based planning with rescheduling

The energy based planning tool is based on XETICS LEAN which helps to incorporate energy saving measures in accordance with DIN EN ISO 5001. Collect and display your energy consumption in detail. Identify the "energy consumption" and the process-induced peaks in consumption referring to the products, raw materials used, machines and tools. Furthermore it helps to introducing specific measures to improve energy efficiency and develop new planning strategies.
In detail:

- Management of energy meters and hierarchical structures
- Collection, visualization and monitoring of energy consumption
- Automatic targeting system for infringing target values
- Analysis of consumption in correlation with other production parameters
- Consumer Profiles to Identify Power Peaks
- Energy KPIs and Planning Strategies for improving energy consumptions

Table 29: IFeVS Energy based planning with rescheduling - Key Features

ID	Feature Description
F-IF-E-1	XML Standards, JSON Standards, REST Standards, UPnP 1.0 Standards, Java 1.8

7.3.3. Decision Support

Automatic Monitoring and Visualization of KPIs

The OEE for equipment are calculated based on a formula that can be dynamically changed and entered in a script form. Furthermore it can be changed according to the defined priorities and only based on availability.

The OEE gives a quick overview of the effectiveness of the machines. The XETICS OEE is calculated from the events recorded by the XETICS LEAN Android app. This overview allows you to interpret the calculated values.
All values refer to a time interval, for example, a day or hour, and a machine or plant. For several machines or systems, for example a line, mean values are specified.

The OEE (Overall Equipment Effectiveness) indicates the effectiveness of a machine over a period of time:

$$OEE = Availabilty * Utilization * Throughput * Quality$$

The individual factors are explained in detail below.

Availability

The equipment status is used according to the SEMI E10 standard to calculate availability. This provides the following status:

PRODUCTIVE - machine works a job.
STANDBy - machine is ready to edit a job, but does not
REPAIR - machine is repaired unplanned
RAMPUP - machine is retrofitted
MAINTENANCE - Machine is scheduled to be serviced
NONSCHEDULED_TIME - Machine is not scheduled for operation

Availability is calculated as follows

$$Availability = \frac{timeIn(UPTIME)}{timeIn(SCHEDULED)}$$

Utilization

The Utilization specifies the load of the machine, ie the time at which a job is processed in relation to the time at which a job is edited or edited:

$$Utilization = \sum_{TIMESPAN \in timespanWithSameNumberOfJobs(UPTIME)} \frac{jobsOnEquipment}{equipmentCapacity} * \frac{length(TIMESPAN)}{UPTIME}$$

Throughput

$$Throughput = \frac{\sum_{OPERATION \in OPERATIONS(TIMESPAN)} getProcessTarget(OPERATION).maxNominalDuration()}{\sum_{OPERATION \in OPERATIONS(TIMESPAN)} actualTime(OPERATION)}$$

The throughput is calculated as follows

There are two special cases:
1. A processing is only partial in one interval and partly in another:
2. A processing is still running "now":

Quality

$$Quality = \frac{\#\ okJobsInTimeSpan}{\#\ jobsInTimeSpan}$$

Table 30: IFeVS - Automatic monitoring and Visualization - Key Features

ID	Feature Description
F-IF-AM-1	An OEE can be calculated with his own formular to get useable statistics data.

8. Testing and Demonstration Approach

The following chapter describes the testing and demonstration approach to de-risk the PERFoRM technologies and to demonstrate the Plug & Play capabilities of the PERFoRM components in reconfigurable environments. Therefore subchapter 8.1 is first describing the environments or demonstrators used in this task, which are the *Modular Demonstration Line* (Chapter 8.1.1) and the *"Mini"-Flexible Cell* (Chapter 8.1.2), a Test Environment which is built for the PERFoRM project, in the SmartFactory. Chapter 8.1.3 describes a demonstration environment of IPB - the *Small-Scale Production System* which is also used as a pre-industrial test bed with a focus on the agent based reconfiguration tool.

After the description of the main test environments for T6.2, the virtual test approach for the integration test and the demonstration approach for the validation in reconfigurable and physical environments are explained in chapter 8.2.

8.1. Test and Demonstration Environment

8.1.1. Modular Demonstration Line

Figure 24: Modular production system in the SmartFactory

The demonstration line in the SmartFactory shall demonstrate the feasibility of the PERFoRM-Architecture in a reconfigurable environment and gives a platform for testing and demonstrating the PERFoRM-Tools.

This demonstrator is built in a highly modular and flexible way to make modularity not only available on the assets level, but also on the infrastructure and the IT-Level. Thus, it is very suitable to provide a platform for demonstrating the capabilities of the PERFoRM-Tools and their collaboration in a realistic Plug '&' Produce environment.

The demonstrator can be divided into three parts as depicted in Figure 25. There is the production level consisting of individual production modules, a modular infrastructure which provides the modules with energy, communication and safety functionalities, and a modular IT-Architecture, where different kind of modules can interact via an integration layer with the physical level. The three parts will be explained in the following subsections.

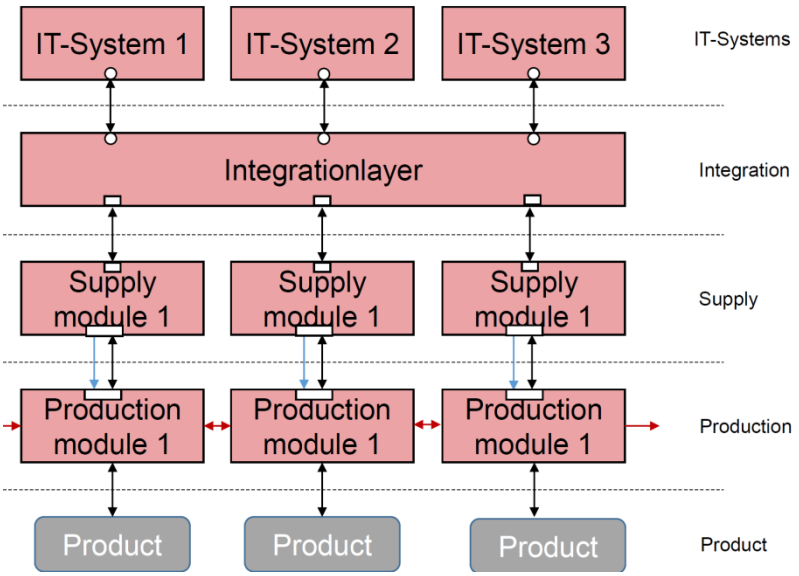


Figure 25: System Architecture

The manufactured product is an individualized business card holder (Figure 26). It consists up to four components (base plate, clip, lid and an inlay), which can be personalized by engraving parts or choosing different colors of the lid.

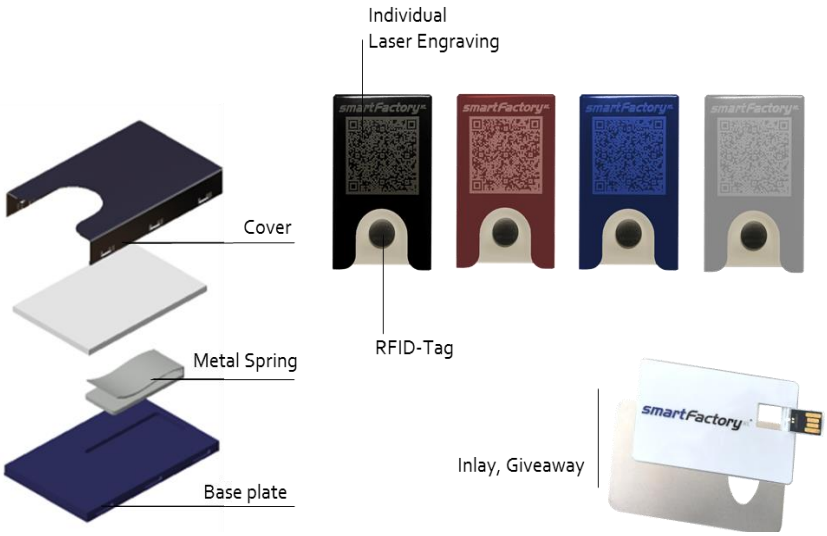


Figure 26: Product - Personalized business card case

Production Modules

The production line currently consists of 9 individual modules with different functionalities as explained below. Those modules can be combined in any way to provide the necessary production functionality, but they can also work individually, e.g. if only one module functionality is required.

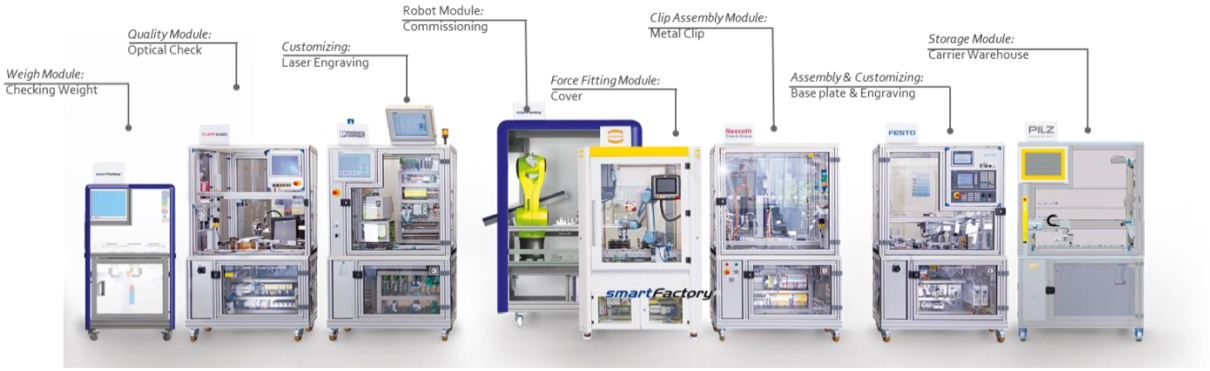


Figure 27: Reconfigurable and modular demonstration environment

1. The **Storage Module** serves as an intelligent storage system for the workpiece carriers: Sensors register each carrier when passing the module and identify it. If the workpiece carrier is empty, it is removed from the production flow. If a new order is reported and thus the need of an additional carrier, this module brings an empty carrier from the storage into the process.
2. The **Assembly & Customizing Module** starts the production flow by providing a new plastic bottom with an integrated RFID-Tag and it initializes the digital production memory with the production order from the ERP-System. Depending on this production order an engraving is added to the bottom in this module.
3. The **Clip Assembly Module** mounts a metal clip on the plastic bottom with a pick and place robot.
4. The mounting of the two housing parts takes place in the **Force Fitting Module**. According to the production order, a cover with one of two colors is mounted to the base plate, by embossing both parts together with a robot.
5. The **Robot Module** is a function-expanding module. It can be docked to the force fitting module or to the quality check module (see module 7) to provide additional lids in different colors on the one hand or to insert a give-away into the assembled business card holder on the other. If the module is docked to one of the mentioned modules, it is automatically detected and included into the process to provide its functionalities.
6. The **Laser Marking Module** puts an individual laser engraving on the cover of the business card holder. Depending on the production order it puts the name and a digital business card with an engraved QR-Code.
7. The end control and the output of the product take place in the **Quality Module**. The quality check is done by a vision control system. A scalar robot takes the product if it is finished and removes it via a product slide.
8. An in-process quality control is provided by the **Weigh Module**. The control is done with an integrated scale which checks the weight of each product and compares it with a reference.
9. Almost each module can be replaced by the **Bridge Module**. It can discharge a product for the replaced production step and can then be further processed on a *Manual Work Station*.

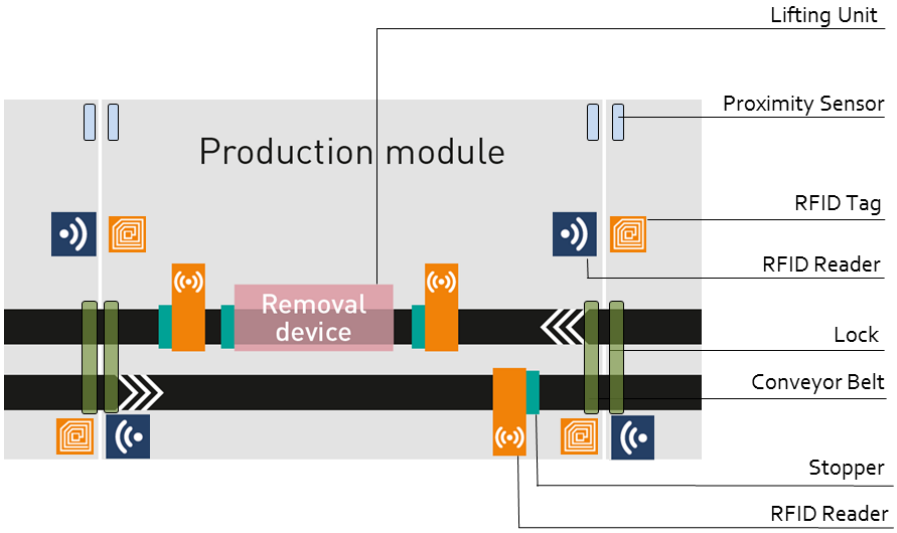


Figure 28: Basic layout and functionalities of a module

The basic layout and the mandatory components of a module are shown in Figure 11. A module consists of a conveyor belt transporting the workpiece carriers to the neighbor module or leading the carrier back, when there is no neighbor module, by closing the locking door. The recognition of neighbor modules is done by a combination of induction sensors and RFID-Readers which are mounted on each side of a module interface. Modules close to each other read the ID of adjacency modules. After verifying the id with the information of a central topology manager, the locks are opened on the side with the connected module.

For the identification of production steps, each module has three RFID-Reader mounted, which reads the production information from the production memory of the product, when the product enters a module and writes the production process to the production memory when the product leaves the module. A third reader is placed on the way back, which helps localizing products. Stoppers help in these processes by stopping a product close to the RFID-Reader.

The production process step starts, when the next step of the order can be done by the current module. In this case, a lifting unit is used to excavate the workpiece carrier in a way that the individual manufacturing process can be done while other products, which enter the module, can still pass, e.g. if the process step was already executed, it is not necessary, or if a high priority order is coming next.

Modular Infrastructure and Connectors

The infrastructure level encompassing all essential infrastructure functionalities for the production modules enable the universal combination of modules with minimal configuration effort. The infrastructure functionalities include the energy supply, the communication or data routing and the control of safety functions. These functions are provided by the so called infrastructure boxes, which can be setup in the same modular way as the production modules. Figure 29 shows one of them, together with a universal connector, based on HAN-Modular® which provides all the necessary supply by means of power, air pressor, Ethernet and connects the module to the emergency stop loop.

Figure 29: Modular Infrastructure Box (left) and Socket/Connector (right)

IT-Architecture

The IT-Architecture is basically setup around an integration layer – the IBM Integration Bus, which enables the vertical integration and provides a loose coupling between tools and the physical modules or infrastructure boxes as depicted in Figure 30. The integration acts as an Enterprise Service Bus (ESB) and thus it can provide different interfaces like MQTT or Web-services, routing protocols and transforming data models. This enables that different tools like an ERP System, a database, a simulation tool or a dashboard can easily access the system, without the need to adapt the interfaces or touch the implementation.

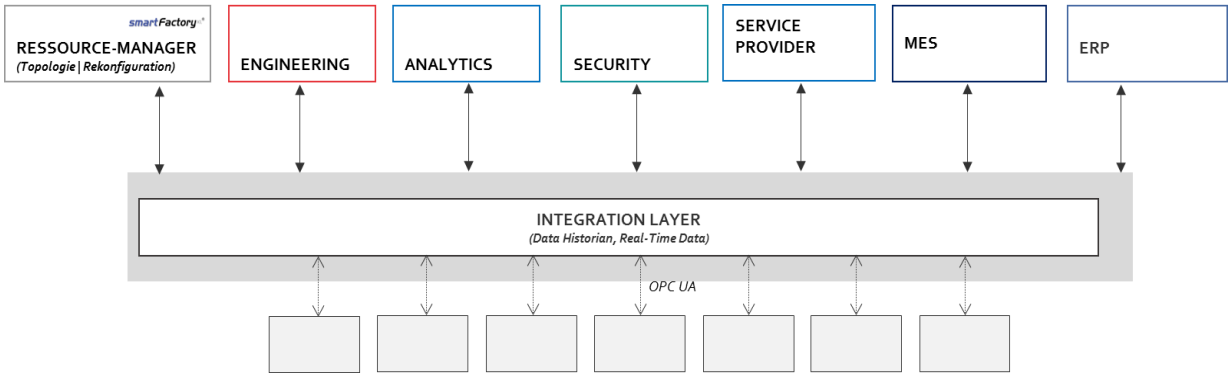


Figure 30: Current IT-Architecture in the SmartFactory demonstration line

The interaction with the modules and the infrastructure boxes is provides with OPC-UA. For this, each asset is encapsulated with OPC-UA as indicated in Figure 31 and Figure 32. Each module or box has an integrated OPC-UA server and provides the information within the specified information model.

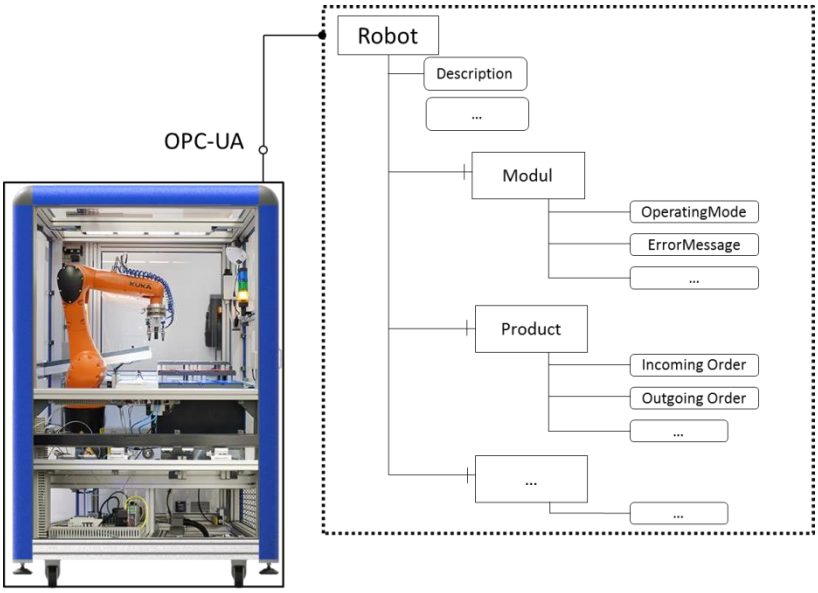


Figure 31: Encapsulation of a module with OPC-UA

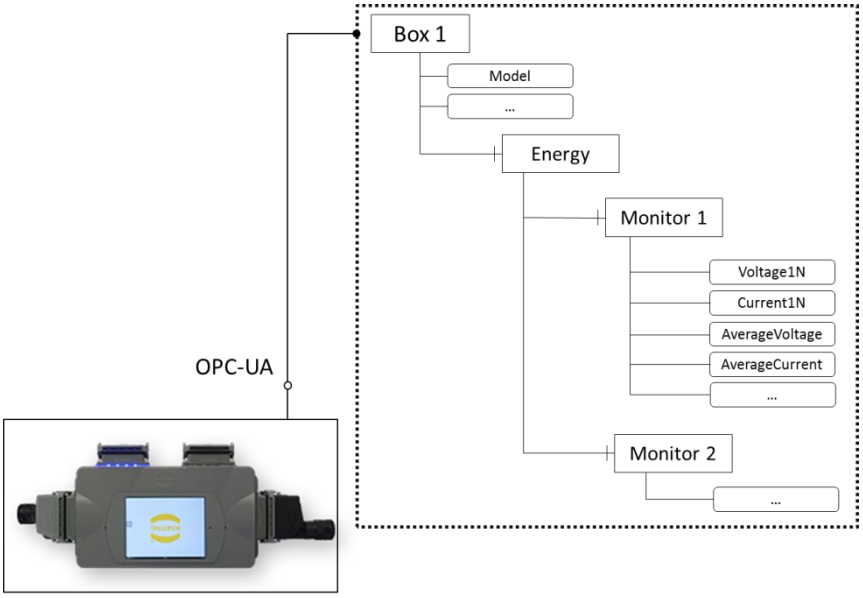


Figure 32: Encapsulation of a infrastructure model with OPC-UA

Process

Figure 33 illustrates the standard process for a business card holder. It involves the following modules with its associated skills and production steps:

1. Order from the ERP system
2. Topology manager checks if the order can be performed with the available modules (not illustrated)
3. A empty workpiece holder is provided by the *Storage Module.*
4. The *Assembly & Customizing Module* provides a base plate, engraves it if necessary, and writes the production steps to the product memory.

5. The *Clip Assembly* adds a metal spring to the base plate.
6. The Fore Fitting Module grouts the cover with the base plate
    a. If an inlay is ordered, the *Robot Module* will add it to the product in the same step
7. The *Laser Marking Module* engraves the individual name and marker on the cover
8. The *Quality Module* checks the product and release the complete product out of the process
9. The *Weigh Module* checks the weight of the product if required. Only required if an inlay was ordered.
10. The empty workpiece holder is put back in the *Storage Module*.

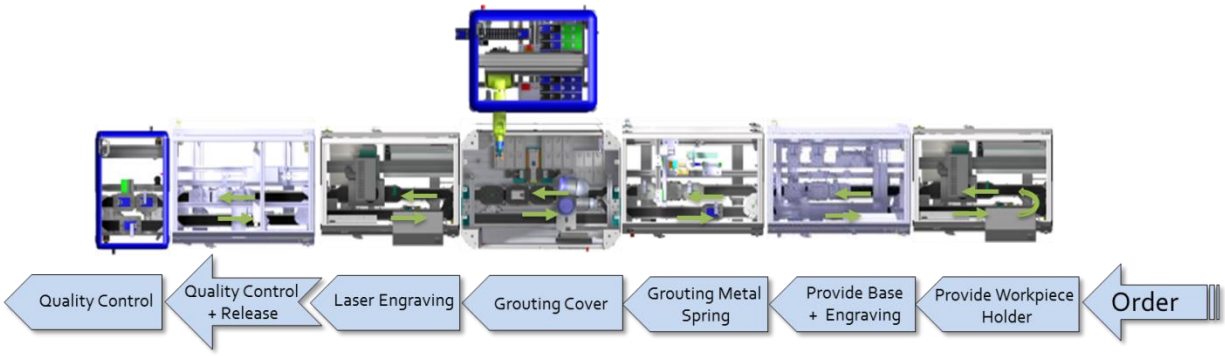


Figure 33: Standard process for a business card holder

The cycle time for each production step and the total cycle time for the complete production depend on the degree of individualization; it depends in particular on the color and the inlay. An overview of the cycle time is given in Table 31. Please consider that the cycle time given in this table for the complete production is only true if the modules are put together in the order which reflects the order of production steps. If the modules are positioned in another order, the total production time is higher, because an incomplete product must run a loop.

Table 31: Cycle time per module and product

Modul	Black Cover	Grey Cover	Blue Cover	Red Cover	Black Cover with USB Inlay	Black Cover with bottle opener	Grey Cover with USB Card	Grey Cover with Bottle Opener
Storage	45	45	45	45	45	45	45	45
Assebly & Customizing	20	20	20	20	20	20	20	20
Clip Assembly	30	30	30	30	30	30	30	30
Force Fitting	50	60	60	60	50	50	50	50
Laser Marking	50	60	120	180	50	50	60	60
Quality	40	40	40	40	80	80	80	80
Weigh	20	20	20	20	20	20	20	20
Total	**255**	**265**	**335**	**395**	**295**	**295**	**305**	**305**

8.1.2. "Mini" Flexible Cell

The "Mini" Flexible Cell is a demonstrator which is currently planned and setup as a test and demonstration environment specifically for the PERFoRM-Project. Within the scope of this Project the idea of this demonstrator is to show Plug & Play concepts and technologies, but also to reflect the use cases as best as possible, this mini flexible cell is based at the idea of the GKN use case (comp. Chapter 6 or D10.1) of a flexible cell including a robot and different process modules. Figure 34 shows a first draft of the CAD-Modell of the demonstrator. The red line is supposed to encompass the area that can be reached by the robot. The demonstrator is still in a planning phase; therefore not too many details can be given at this time. The basic setup will include a collaborative robot module (probably a Universal Robot 3) and two process modules. Furthermore, the current idea is to demonstrate modularity not only on module level, but also on subsystem level.

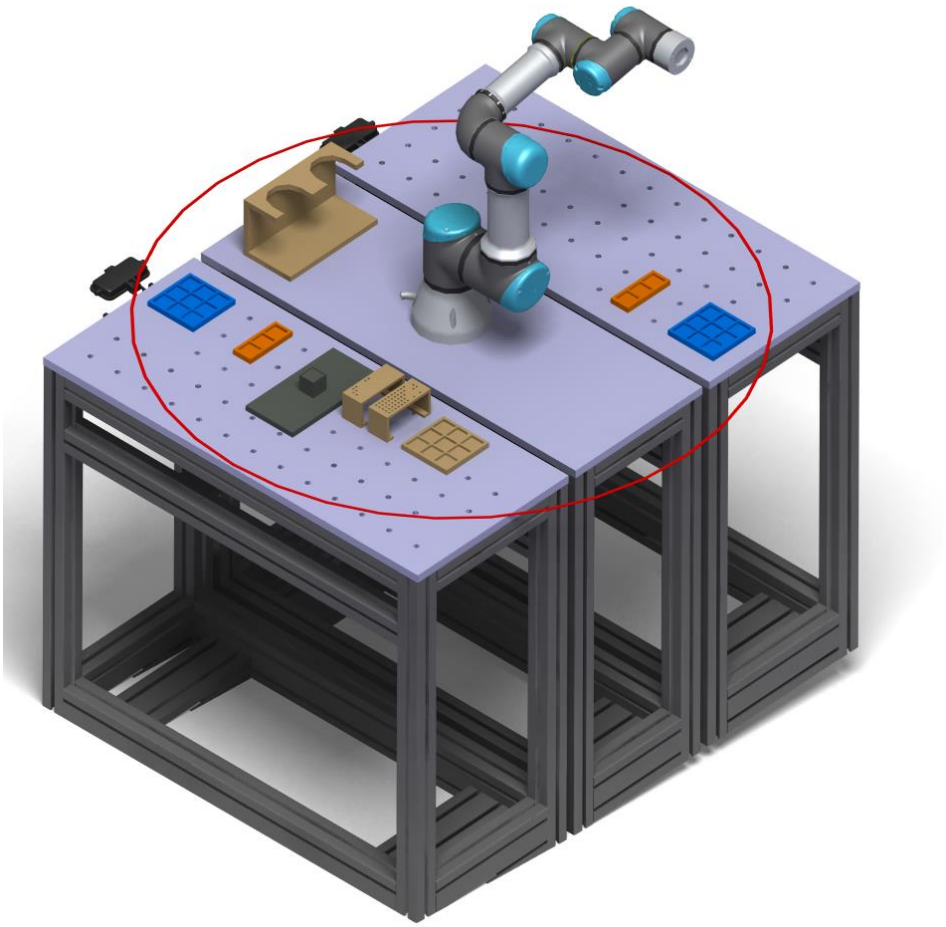


Figure 34: Current CAD-Model of the „Mini"-Flexible Cell demonstrator

Electromechanical concept

As previously mentioned, the flexible cell will be composed of 3 modules:

- A collaborative robot module (in the middle)
- An automated process module with active subsystems (on the left side)

- A manual process module (on the right side)

The product to be assembled by the cell is a classical dice, in which screws will have to be inserted by the robot on the automated process module side and bolts will have to be inserted manually on the manual process module side. This should allow showing the production of individual products. The dice is pictured in Figure 35.

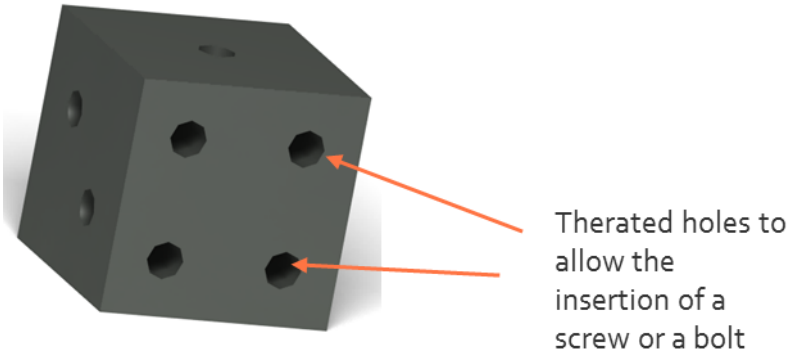


Figure 35: Dice which need to be completed by the demonstrator

The generic layout of the automated process module is shown in Figure 36. The manual process module will be built in a similar way. The dice storage will receive the empty dices to be assembled. The assembly fixture will be designed as an active component, encapsulated as an autonomous CPS, and allow the insertion of the screws in the dice by working together with the robot. The dice buffer will allow the storage of unfinished dices which have to be stored until the worker takes over the next steps, e.g. the manual insertion of bolts on the manual process module. The ejection board will receive the finished dices. Single sensors and actuators will be controlled by the module PLC.

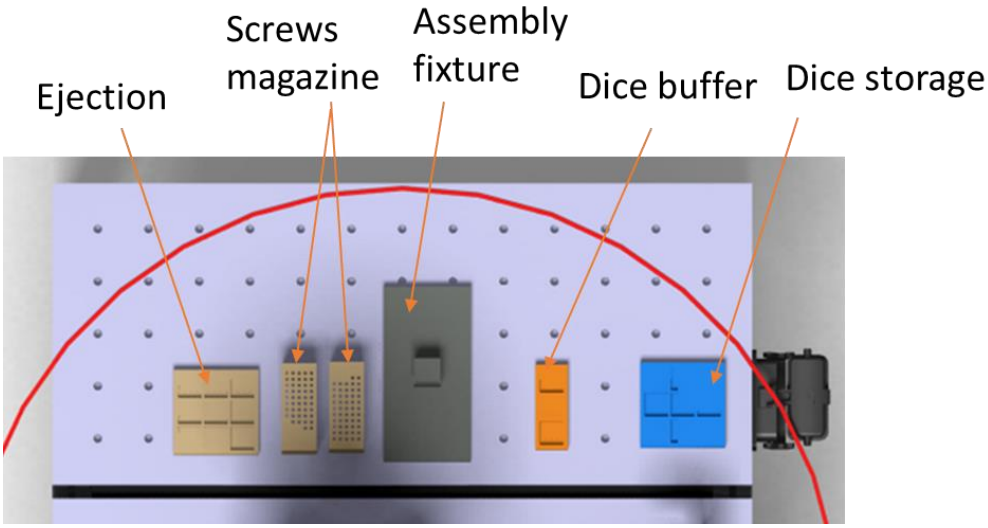


Figure 36: Generic layout of a process module

Thanks to the perforated plate, all these subsystems will be easily exchangeable according to the Plug and Play principle, allowing modularity on the subsystem level. To show this modularity, other subsystem will be designed to build another version of the dice, e.g. a bigger one. At this time, the supply interfaces between the subsystems and the module have not been designed yet.

To allow the Plug and Play ability on module level, a HAN-Modular® -based connector will allow the connection between the process module and the robot module. Hence, the robot module will host the

control box that is necessary for the power, pressed air and Ethernet supply as well as the emergency stop loop connection. Furthermore, the robot module will dispose of the tools storages which will contain the different tools that are needed by the robot to complete the assembly tasks.

The coupling between the robot module and the process module is realized by a detection and locking mechanism (Figure 37). On the robot module, this one is composed of a reed sensor for the detection of the process module, an RFID reader for the identification of the process module as well as two cylinders to allow the locking with the process module. On the process module, a magnet allows the detection through the reed sensor, an RFID tag allows the identification through the RFID reader and two hooks allow the locking with the cylinders in order to maintain the two modules together.

Furthermore, each process module will dispose of its own HMI for operation and visualization.

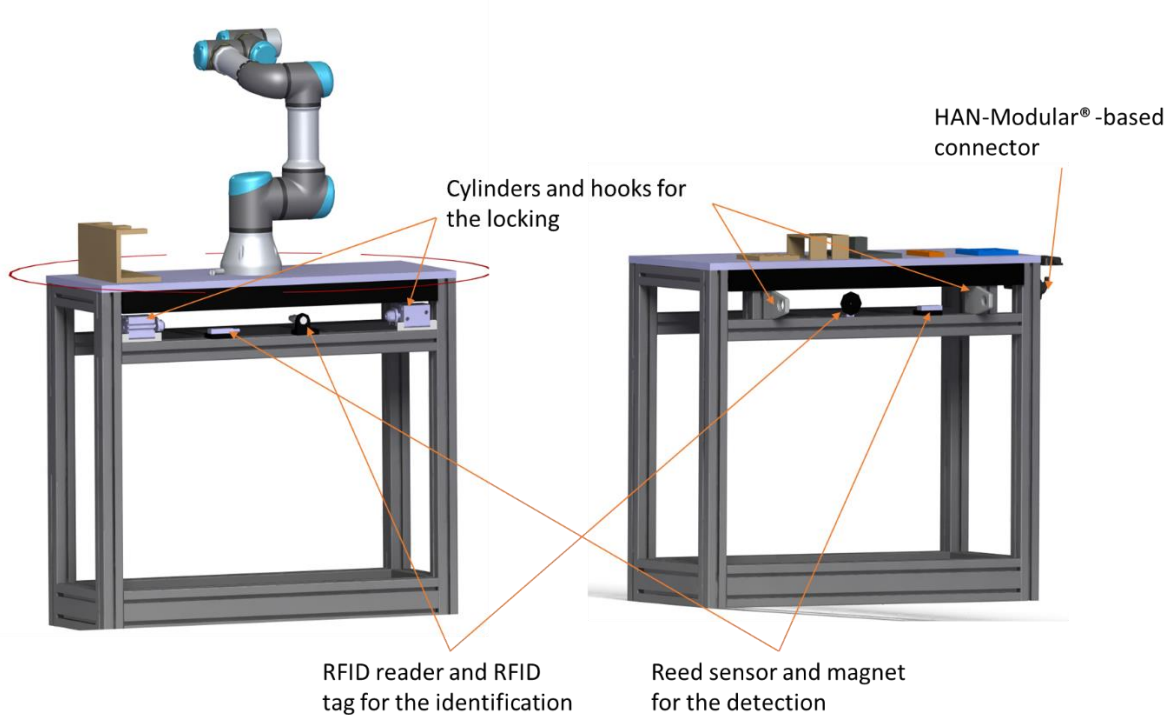


Figure 37: Supply, detection and locking mechanisms of the demonstrator

IT concept

The first definition of the IT-architecture, which is shown in Figure 38, is based on the concepts of PERFoRM, including two middleware solutions, OPC-UA at cell level and a PERFoRM compliant Integration Layer at the factory level. This setup enables a high coverage of required tests and demonstrations.

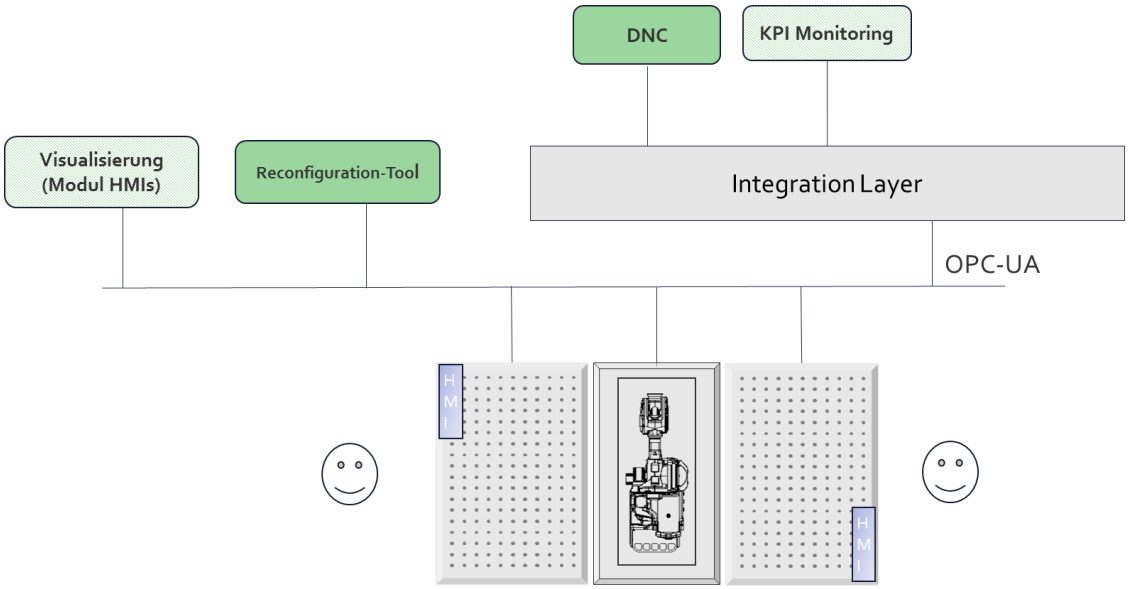


Figure 38: Basic System-Architecture of the "mini" Flexible Cell

The current concept to enable reconfiguration is by integrating the machinery level with OPC-UA. Therefore the module controller has to be upgraded with an OPC-UA adapter, which is in line with the use case of GKN. Each module will be encapsulated with OPC-UA as depicted in Figure 39. The reconfiguration tool enables the Plug '&' Play capabilities of the cell by reading and writing to OPC-UA nodes or using OPC-UA functions. Other tools like a DNC-Tool for the robot program download, simulation or monitoring tools can be attached to the integration layer by using the PERFoRM-Interfaces and a PERFoRM compliant middleware. This approach also enables the virtual tests within a PERFoRM-Architecture which is explained in the following chapter 8.2.

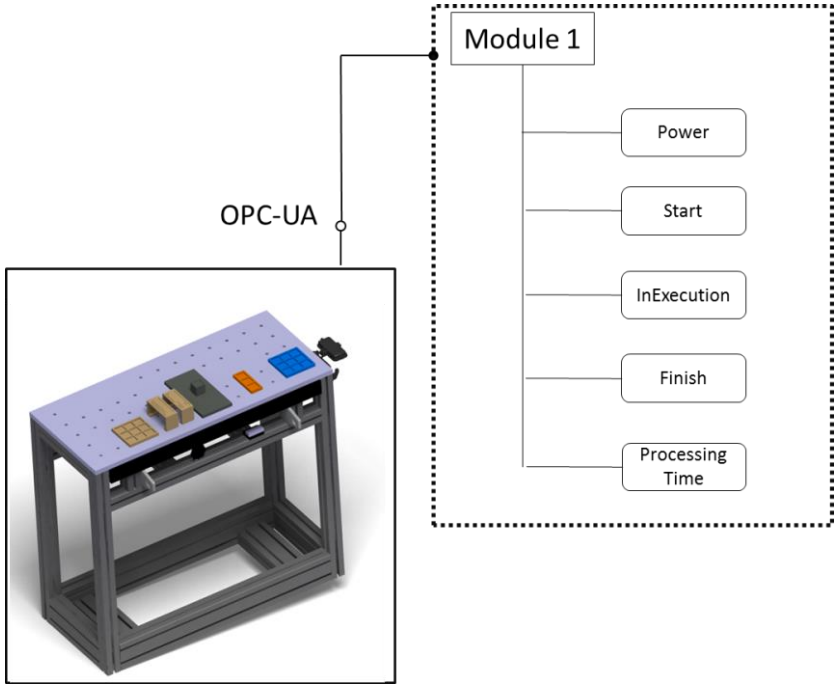


Figure 39: Module-Integration with OPC-UA

8.1.3. Small-Scale Production System

Polytechnic Institute of Bragança has a small-scale production system that allow the simulation of production processes, being currently mainly used to test distributed and decentralized manufacturing control approaches and for the development of standard interfaces concepts, as also for the support for the diverse courses taught.

The real small-scale production system is composed by one IRB 1400 ABB robot, two punching machines, two indexed lines, one pneumatic station, one RFID reader and a storage system, as illustrated in Figure 40. The circulation of the products being produced within the flexible production system is tracked by RFID readers (note that each product has associated a different process plan to be executed). The machines are supplied by Fischertechnik™ and constitute a hardware platform that provides the necessary experimental environment.

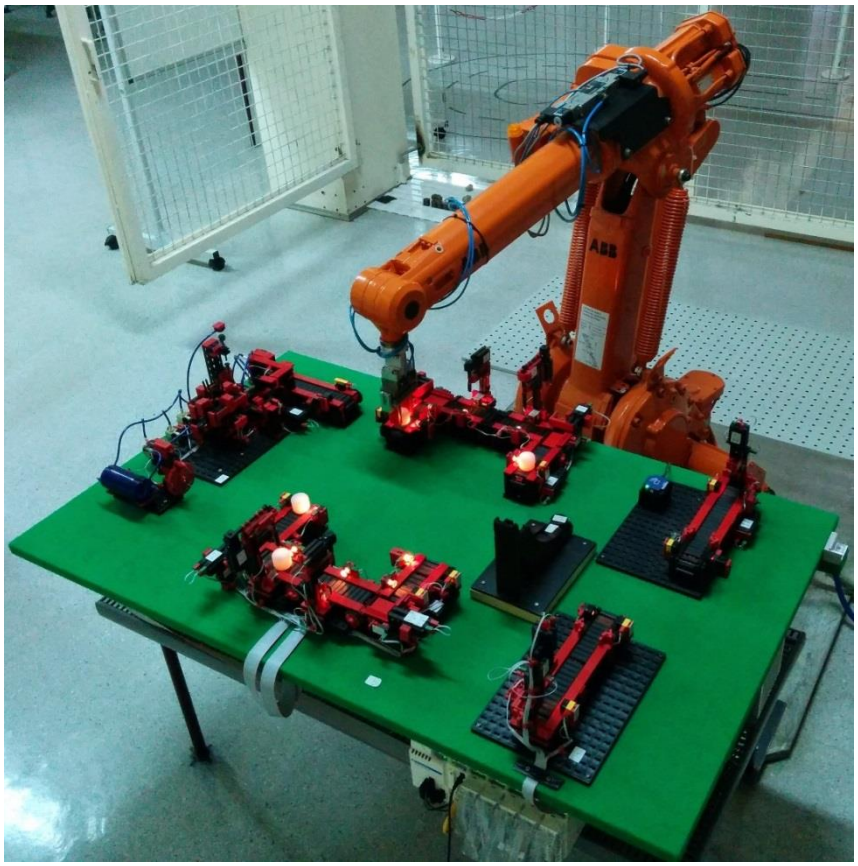


Figure 40: IPB's small-scale production system

An example of a set of possible skills and process plan for the products are depicted in the following tables.

Sequence	Part "A"	Part "B"
#1	punch_1	drill_1
#2	pneumatic	drill_2
#3	drill_1	punch
#4	drill_2	inspection
#5	inspection	-

Table 32: Process Plan for the Catalogue of Parts

Resource	{Skill, time}
Manipulator robot	{transfer, 3}
Punching machine A	{punch_1, 5}
Punching machine B	{punch_1, 7},
Indexed line A	{drill_1, 7}, { drill_2, 6}
Indexed line B	{drill_1, 5}, { drill_2, 9}
Pneumatic	{pneumatic, 10}
RFiD reader	{read, 2}
Inspector	{inspection, 3}

Table 33: Resources Skill Matrix

The punching machines are composed of two infrared sensors to detect the parts in the beginning of the conveyor and in the punching position, and two switch sensors to detect the end of the movement of the punching device. The conveyor and the punching are moving through two DC 24V motors. The capacity of each cell is to process one part at each time.

The indexed lines are composed of two workstations interconnected by several conveyors disposed in U shape, allowing processing four parts simultaneously. The conveyors and machines use eight DC 24V motors (four for the motors, 2 for the machines and 2 for the embolus). Four switch sensors are used to determine the range of movement of the embolus and five infrared sensors are used to detect the presence of the parts in the conveyors and in the processing positions.

The pneumatic station is composed by one air compressor, two infrared sensors to detect the presence of the parts at the start and end positions, one turntable to move the parts, receiving the parts from the start position and moving it to the processing positions. This turntable also transfers the parts to the conveyor exit. There are 3 switches, one that is responsible to align the turntable and the others to detect the parts in the operation positions. When the switch responsible for aligning the turntable is active simultaneously with the other sensors, the turntable stops and the operations transfer in, processing and transfer out are executed. The transfer of the parts is performed using one of the two existing cylinders that are controlled by solenoid valves.

The low-level logic control of these machines is implemented as IEC 61313-3 programs running in a Modicon M340 PLC. The data access to the PLC memory can be performed by external components using the Modbus communication protocol, allowing for example to read and write memory spaces (e.g. related to input or output signals).

A human operator performs visual inspection operations to verify if the processing operations are performed according to the specifications. The human operator interacts with the system through the NS8-TV10-V1 Omron Human-Machine Interface (HMI) display, connected to the system using a C200HG PLC from Omron. The industrial manipulator robot executes the transfer operations between the stations using proper RAPID programs and is externally accessible through the ABB S4 DDE Server.

All the signals from the above system description are accessible by means of a OPC-UA server installed in a cell computer.

The presented system will be used internally by IPB as a first phase validator, where all the necessary software development, integration and operational testing will be performed. Naturally, the described machine skills and products are merely indicative and will not be considered as-is. Yet, the machines, e.g., will be used to represents the processes in the GKN's use case where plug-in and plug-out will be emulated by turning-on and turning-off their power.

8.2. Test and Demonstration Approach

Two test/demonstration approaches are taken into consideration to de-risk and test the technologies and concepts of PERFoRM; firstly a virtual test approach for the specific use cases with a focus on the integration of the IT-Level, where real offline data from the use cases will be used to reflect the machinery level (see section 8.2.1) and secondly a demonstration approach within a physical environment, which are described in section 8.1, to demonstrate the feasibility of the concepts and the technologies developed in PERFoRM in a reconfigurable environment (see section 8.2.2). Naturally those test labs can't reflect an industrial and production environment in all aspects, but it can show how the technologies work in general, which can then be adapted to the specific requirements of a real industrial factory.

Regardless of the approach the following steps must be done for all integration tests:

1. Setup of the PML to describe the modules and/or plant/factory.
2. Implementation of the required interfaces in the middleware solution.
3. Setup of the required DMS and the datasets.
4. Integration of the adapter to the DMS.
5. Testing of the interconnection/interfaces between the adapters and the middleware solution
6. Integration of the solutions from WP4 step by step.
    a. Test the interfaces between the tools and the middleware.
    b. Test the collaboration between the tools via the middleware.

8.2.1. Virtual Test Approach

Working with virtual test environments are state of the art when it comes to testing software products, because the environments can be easily setup and adapted to the needs of the tester. When it comes to integration tests of the IT-Solutions developed in PERFoRM within a PERFoRM compliant architecture, this approach can be used to test parts of the uses cases with in a realistic way; esp. when the use cases are working with "offline" data. Such data can not only be used to test the static behavior of a system but also to simulate new machinery states or production data by inject simulated values which reflect relevant test conditions. This approach is also independent of test environments; therefore this can also be setup directly in the use case factory for the tests, as long as the system doesn't influence the real production

Figure 41 and Figure 42 depict the Siemens and the Whirlpool use case. Both of them can work with "offline" data, which can be accessed from the database systems. In the Siemens use case, which is about the implementation of a predictive maintenance system (cmp. chapter 4 or D10.1), the tools, in particular the Data Analytics tool, is accessing the data via the database systems and is not directly connected to a machine. Therefore this approach can be used to test the workflow on the IT-Layer.

A very similar concept, also the use case has a different goal, is the one from Whirlpool (see Figure 41). This use case is also providing all relevant data by a database which can be accessed by the tools via the middleware.

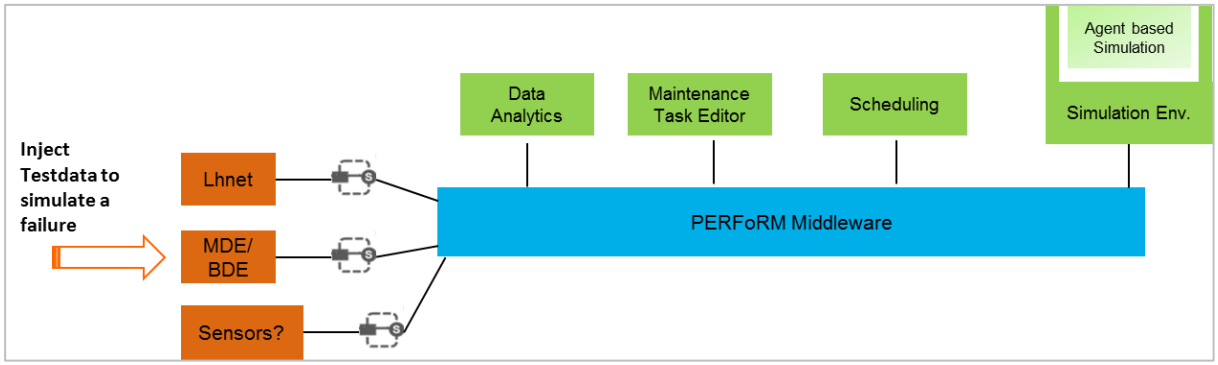


Figure 41: Injection of test data to the Siemens workflow

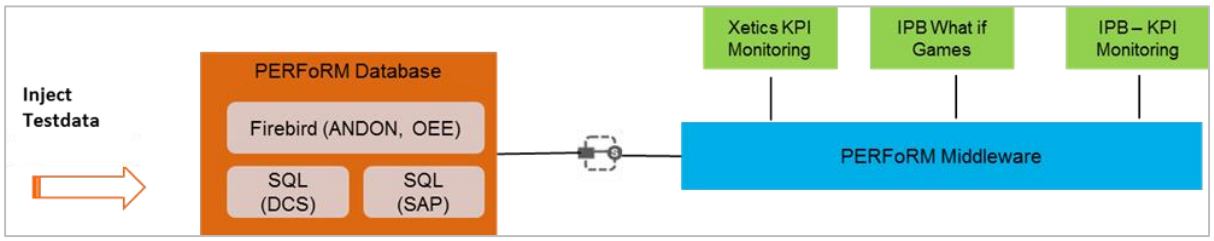


Figure 42: Injection of test data to the WHR workflow

8.2.2. Demonstration Approach

The Demonstration Approach is focusing on the demonstration of the concepts and technologies of PERFoRM. In contrast to the Virtual Test Approach, modular and reconfigurable demonstration environments will be used, where the IT-Level is connected to the shop floor. Because the Mini Flexible Cell is currently developed for the PERFoRM-Project, the major demonstration scenarios will be mapped to this cell, but also the other environments (see chapter 8.1) will be used to demonstrate the different aspects of PERFoRM.

Different possible demonstration architectures using the mini cell are depicted in Figure 43 to 46. It shows how the architecture elements are integrated and how the different solutions can be demonstrated. Those solutions will be partly adapted e.g. to visualize the state or KPIs with data produced by the demonstration environment and not with the data from the industrial use cases. But it also acts as a real test study, e.g. by using a PLC used by GKN and the PLC-adapter to show the integration of legacy PLCs to OPC-UA or by using the reconfiguration tool for the reconfiguration ability of the cell, which will also be used at GKN.

A more detailed description of the demonstration, the setup of the demonstrator and the documentation of the results will follow in the next deliverable of this task (D6.5: *"Self-Adaptive Highly Modular and Flexible Assembly Demonstrator Documentation and Results"*)

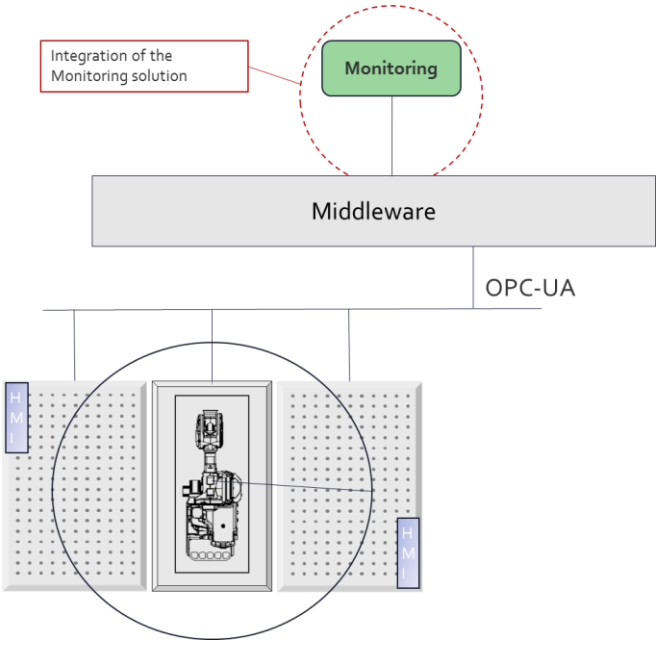


Figure 43: Possible Demonstration Architecture with the Mini Flexible Cell for the Use Case of IFEVS

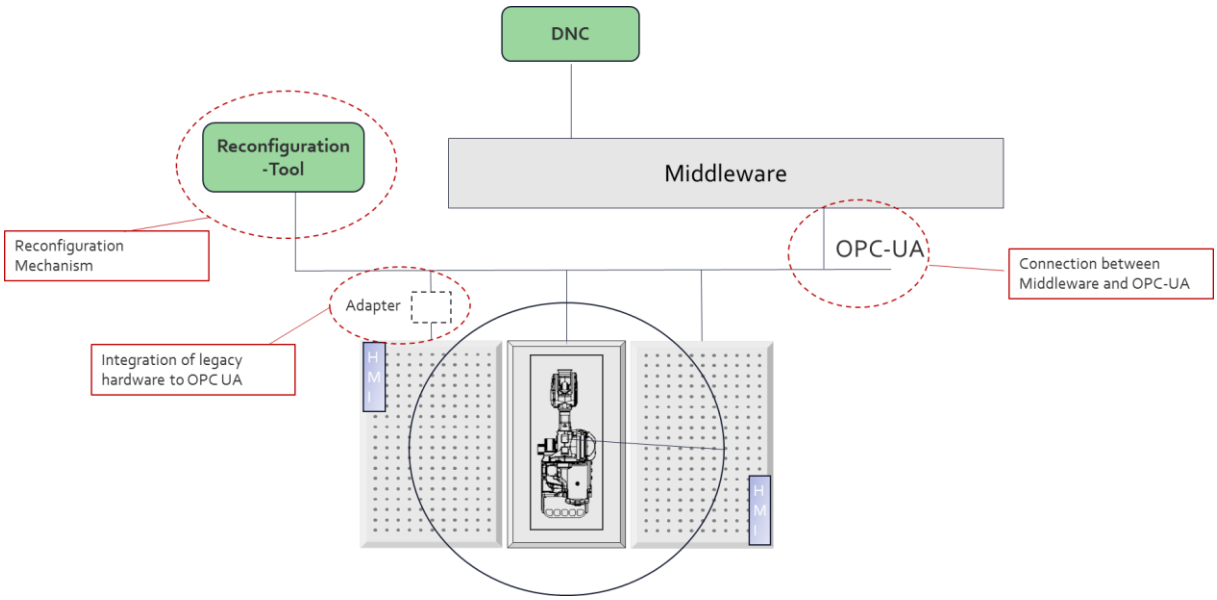


Figure 44: Possible Demonstration Architecture with the Mini Flexible Cell for the GKN Use-Case

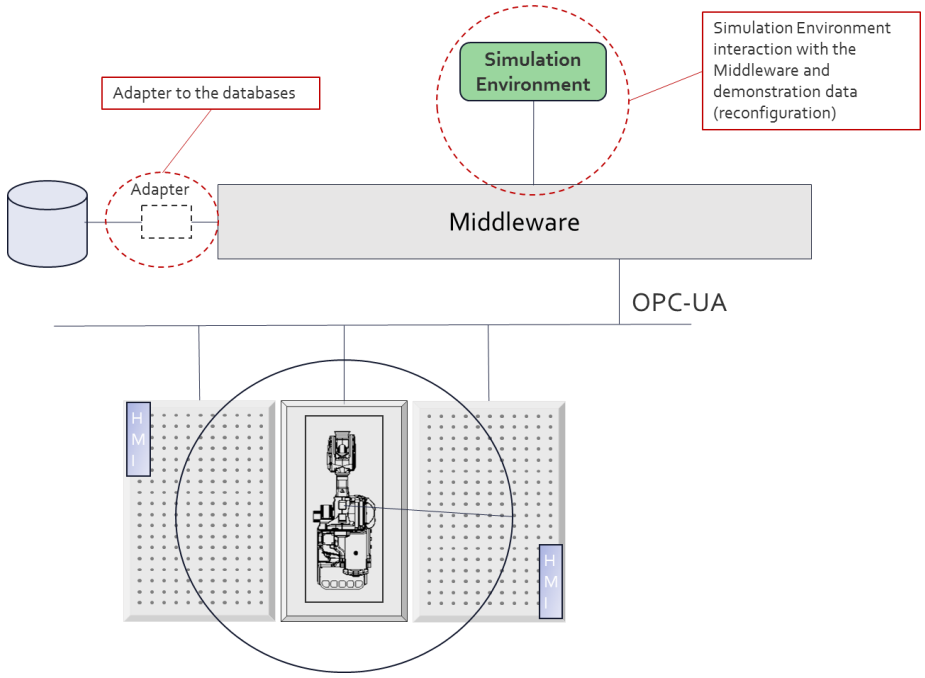


Figure 45: Possible demonstration architecture with the Mini Flexible Cell for the Use-Case of Siemens

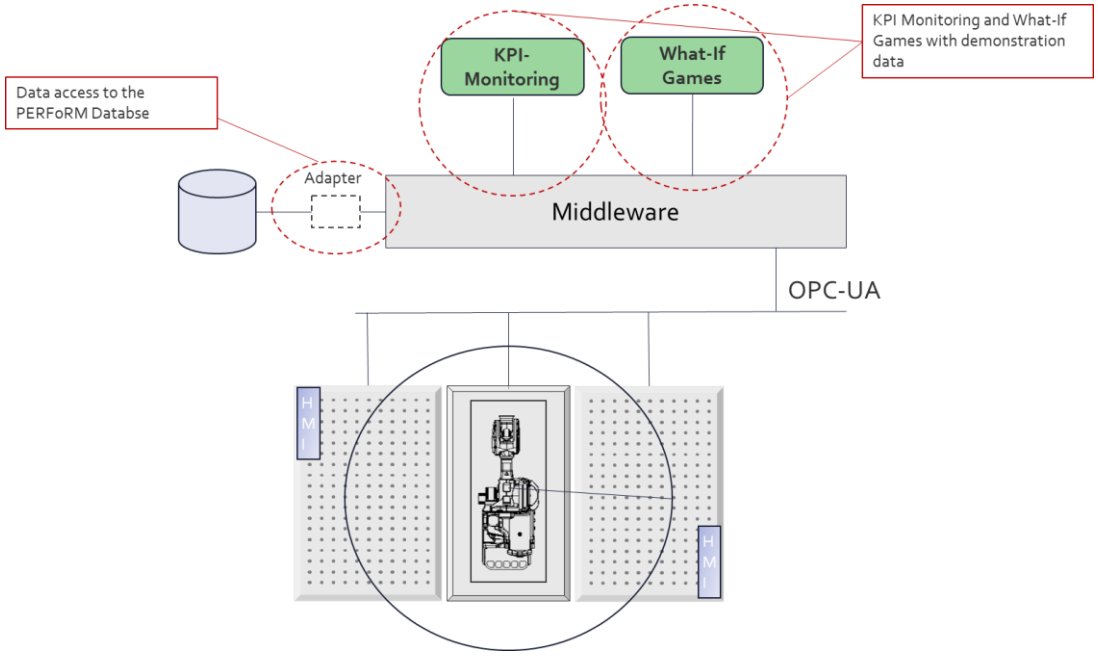


Figure 46: Possible demonstration architecture with the Mini Flexible Cell for the Use-Case of Whirlpool

9. Test Management

9.1. Environment Requirements

To reflect the use cases environments with a test or demonstration environment it is necessary to collect the required IT-Technologies like DBMS, protocols, tools or others which are used in the specific Use Case. For this and among others, a technical questionnaire was setup up in cooperation with WP2, WP3 and WP6.

The technical questionnaires and answers are attached in Appendix III.

Whirlpool:

- Tools deployed in **Virtual Machines** (VMWare)
- **Database – SQL 2000** (Update rate from the shop floor in hours. The update rate might influence the virtual tests, in particular the injection system)

IFeVS:

No requirements regarding the IT-Level and tests.

GKN:

- **OPC-UA** as the cell level middleware
- **Biz-Talk** as the factory middleware
- **S7-300** for the Cell Control. This is not directly related to the scope of T6.2 but might have an influence when testing the IT-Integration
- A **DNC system** (HI-Fit) to download robot programs to the robot controller from a repository

Siemens:

- **Oracle database** for the LHnet data.
- **SQL database** for the MDE/BDE data.
- **SAP APO**

9.2. Test Risks

The following points give an overview of risks during the test process and in the approach of testing the technologies and concepts for the industrial use cases in a pre-industrial testbed.

Development is not completed and blocks test and demonstration activities. Within the scope of this task, the following points must be ready before integration tests can be performed:

- Middleware must be ready and deployed to be able to test the integration of other tools
- Standard interface must be defined and implemented

    - PML descriptions must be ready and accessible via the middleware

- **Required environments are not available.** The previous chapter described some environment requirements to be able to test or demonstrate the results in this project. Those requirements must be available during the testing and demonstration activities.
- **Data and the process in the test and demonstration labs are not relevant for the use cases** and therefore doesn't de risk the technologies. Mainly research labs are used to test the use cases, while the use cases are from the industry and represent four different domains. Naturally not everything can be reflected to a test environment.
- **Demonstrator or test environments are shared between other activities which might influence the test results.** Test environments which are not specifically setup and only for the project, might have some dependencies which influence the test results.

10. References

[D1.1] Deliverable D1.1, Report on decentralized control & Distributed Manufacturing Operation Systems for Flexible and Reconfigurable production environments

[D2.1] Deliverable D2.1, "Guidelines for Seamless Integration of Humans as Flexibility Driver in Flexible Production Systems", PERFoRM project, March, 2106.

[D2.2] Deliverable D2.2, "Definition of the System Architecture", PERFoRM project, Sept. 2016.

[D6.1] Deliverable D6.1, "Self-Adaptive Machines Demonstrator Design and Set-Up", PERFoRM project, Sept. 2016.

[D7.1] Deliverable D7.1, "Siemens Description and Requirements of Architectures for Retrofitting Production Equipment", PERFoRM project, 2016.

[D8.1] Deliverable D8.1, "Micro Electric Vehicles Description and Requirements of Architectures in View of Flexible Manufacturing", PERFoRM project, 2016.

[D9.1] Deliverable D9.1, "Description of Requirements and Architecture Design", PERFoRM project, 2016.

[D10.1] Deliverable D10.1, "Use Case goals/KPIs and Requirements Defined", PERFoRM project, 2016.

[IEEE29119] IEEE 29119 – IEEE Standard for Software and systems engineering – Software testing, 2013-2016.

[IEEE829] IEEE 829-2008 – IEEE Standard for Software and System Test Documentation, 18 July 2008.

[ISA95] ANSI/ISA-95 Enterprise-Control System Integration, 2000-20013

11. Appendix

I. Acronyms

Abbreviation	Explanation
IT	Information technology
SOA	Service-oriented architecture
MAS	Multi-agent system
HW	Hardware
SW	Software
OPC-UA	OLE for Process Control - Unified Architecture
RDBMS	Relational Database Management System
SQL	Structured Query Language
VSM	Value Stream Mapping
KPI	Key Performance Indicator
KBF	Key Business Factors

II. Template for collecting the solution description, features and required tests

Tool name (contributor)	<Name of the solution>
Use Case	<Use Case where the solution is used>
Tool description	*<A description of the adapter which includes its functionalities and how and why it is used in the use case architecture / workflow>*

ID	Key Feature description
1	*<A feature which is developed to fulfill a requirement for the corresponding use case; This could be a functional feature of the adapter or the interaction with the middleware (interface)...*
2	

III. Technical Questionnaires and Answers

The main goal of this questionnaire is to provide deeper (and localized) technical questions that allow a better specification/design for the middleware, standard interfaces and adapters. The answers may also allow the refinement of the tests to be performed in WP6.

Whirlpool's answers to technical questionnaire

1. Considering the specificities of your use case, which data and its structure is foreseen to be used?

Data type	Yes / No
Product	Y
Process	Y
Resources	Y
Sensors	Y
Others (specify)	

2. Can you estimate at which interval rate the data will be updated?

Interval	Yes / No
Weekly	N
Daily	Y
Hour	Y
Seconds	N
Others (specify)	

3. How is this process managed? E.g., automatic script for data dump?
4. How can the data be accessed? (E.g. through the access to a DB? Which?)

Type of data model	Yes / No
Proprietary Data Base	Y
PERFoRM dedicated DB	Y
Other(s):	

5. Are you using any of the following "standardized/proprietary" way of representing data?

Type of data model	Yes / No
Proprietary data model	Y
Ontology	Y
Unspecified	

Standardized data model	
OPC-UA	N
BatchML (IEC 61512)	N
B2MML (IEC 62264)	N
XMPLant (ISO 15926)	N
CAEX (IEC 62424)	N
AutomationML (IEC 62714)	N
Other(s):	

6. Is it possible for you to describe your data structure? This can be very useful to understand which data we need to account for and in order to provide data translations into the PERFoRM data structure.

   YES

7. What IT infrastructure is present (or can be provided) for the PERFoRM system installation?

   Virtual Server running VMware.

Whirlpool specific questionnaire

1. Which types of databases are used for collecting the information of the production systems (DCS, ANDON, F-TEST, OEE, etc.)? In case of Firebird, which version.

   SQL 2000

2. Have you already defined which database management system should be used as the PERFoRM database? Are you open to suggestions? Are there any requirements or company policies that limit the choice?

3. Which is the model of Universal Robot used for the leakage test? UR5 or UR10? Which software version is currently installed on the robot controller?

   UR10

4. Which is the model of the ABB robot used for the glue dispensing?

5. Are PLCs used in robotic cells for the leakage test and the glue dispensing? Are they connected to the robot controller? Which types of PLCs are used? Which firmware is used?

E-district's answers to technical questionnaire

1. Considering the specificities of your use case, which data and its structure is foreseen to be used?

Data type	Yes / No
Product	Yes
Process	Yes
Resources	Yes
Sensors	Yes
Others (specify)	

2. Can you estimate at which interval rate the data will be updated?

Interval	Yes / No
Weekly	Yes
Daily	Yes
Hour	Yes
Seconds	Yes
Others (specify)	Shifts

3. How is this process managed? E.g., automatic script for data dump?

   By PLC, data need to be accessed by OPC. Specific scripts for data dump/acquisition need to be developed

4. How can the data be accessed? (E.g. through the access to a DB? Which?)

Type of data model	Yes / No
Proprietary Data Base	Yes (PLC)
PERFoRM dedicated DB (production, orders and test results)	Yes
Other(s):	

5. Are you using any of the following "standardized/proprietary" way of representing data?

Type of data model	Yes / No
Proprietary data model	
Ontology	
Unspecified	
Standardized data model	
OPC-UA	Yes
BatchML (IEC 61512)	
B2MML (IEC 62264)	

XMPLant (ISO 15926)	
CAEX (IEC 62424)	
AutomationML (IEC 62714)	
Other(s):	

6. Is it possible for you to describe your data structure? This can be very useful to understand which data we need to account for and in order to provide data translations into the PERFoRM data structure.

   Digital I/O, analog data (order numbers, structured data, notes and feedback by manual operator)

7. What IT infrastructure is present (or can be provided) for the PERFoRM system installation?

   Typical SME network infrastructure: PLC, Ethernet network, wi-fi, desktop pc.

E-district specific questionnaire

1. Which is/are the model(s) of the COMAU robot(s) that you are going to use in the welding cells?

   It has to be defined according to the feasibility study that will be performed during the next months. Robot controller will be C5G/-OPEN

2. Which is/are the model(s) of the PLC(s) that you are going to use for controlling the welding cells and the testing stations? Is the SIMATIC S7-300 from Siemens a good option for you?

   The S7-300 is too much for our needs. We are planning to use Siemens IM-151 family PLCs.

GKN's answers to technical questionnaire

1. Considering the specificities of your use case, which data and its structure is foreseen to be used?

Data type	Yes / No
Product	Yes
Process	Yes
Resources	Yes
Sensors	Yes
Others (specify)	

Comments:

- Product related data will need to be downloaded before processing as well as uploaded after processing to have full traceability for each individual part produced.
- Similar for process data; eg. process control programs and parameters downloaded and some uploaded afterwards as part of the logging/documentation

- Some resource monitoring may be of interest, or eg measure tool time/remaining tool life.
- Process monitoring or measurements data from sensors

2. Can you estimate at which interval rate the data will be updated?

Interval	Yes / No
Weekly	Yes
Daily	Yes
Hour	Yes
Seconds	Yes
Others (specify)	

Comments:

- Product related data will not change very often (weeks/months), but some data needs to be downloaded for each individual part that is produced.
- Similar for process data; eg. process control programs and parameters downloaded and some uploaded afterwards as part of the logging/documentation
- Some resource monitoring may be of interest, or eg measure tool time/remaining tool life, as well as process monitoring, this will need to be updated "continuously" but may not be time critical.
- Measurements data and data from sensors can be large … but can perhaps be stored locally in real time and uploaded as batch after the operation has finished …

3. How is this process managed? E.g., automatic script for data dump?

Yes, I believe the solutions are something like that. E.g. the operator uses different commands on a web interface to make SAP trasactions, access different systems/views, and to download e.g. NC / Robot programs, and upload e.g. probe data from NC to SPS system

4. How can the data be accessed? (E.g. through the access to a DB? Which?)

Type of data model	Yes / No
Proprietary Data Base	Yes
PERFoRM dedicated DB (production, orders and test results)	?
Other(s): SAP, PLM in TeamCenterManufacturing / Engineering	

There is a variety of different systems and ways to store / access data/information

5. Are you using any of the following "standardized/proprietary" way of representing data?

Type of data model	Yes / No
Proprietary data model	
Ontology	

Unspecified	
Standardized data model	
OPC-UA	
BatchML (IEC 61512)	
B2MML (IEC 62264)	
XMPLant (ISO 15926)	
CAEX (IEC 62424)	
AutomationML (IEC 62714)	
Other(s):	

Im not sure / knowledgeable enough about current solutions or preferred solutions to give you a correct answer today. This will be clarified until the Workshop at GKN (or we can probably get some more information before).

6. Is it possible for you to describe your data structure? This can be very useful to understand which data we need to account for and in order to provide data translations into the PERFoRM data structure.

   We need to investigate and specify this better to make a realistic scenario for testing/demnostration

7. What IT infrastructure is present (or can be provided) for the PERFoRM system installation?

   For the demo cell – see the appended pictures …

   (We are also working on updating our infrastructure for IT / communication at our research and test site, but I can't give a detailed / technical answer and specification at the moment)

GKN specific questionnaire

1. Have you already defined which PLC or Embedded System will be responsible for controlling the execution at the process level? Are you open to suggestions?

   What we have available at the moment you can see in the pictures above. Some new equipment we will need to by … and Yes, we are open to suggestions, but at the end we need to choose something with respect also to risk / current standards and preferences. (It will be a balance to test / challenge the future opportunities and what can be motivated from industrial implementation perspective)

2. Have you already defined the hardware for the dimensional inspection process?

   No, not yet. We can consider different options (physical probes/gauges or optical/non contact …)

3. Could you provide a more detailed schema of the hardware architecture and the data flow inside the reconfigurable cell?

   I don't have something to deliver today, but will refine this during preparations for the workshop at GKN

Siemens' answers to technical questionnaire

1. Considering the specificities of your use case, which data and its structure is foreseen to be used?

Data type	Yes / No
Product	No
Process	Yes
Resources	Yes
Sensors	Maybe
Others (specify)	Scheduling

2. Can you estimate at which interval rate the data will be updated?

Interval	Yes / No
Weekly	No
Daily	Yes for daily production schedule
Hour	Yes for minor maintenance
Seconds	Maybe if we have sensor data
Others (specify)	No – asap for immediate maintenance

3. How is this process managed? E.g., automatic script for data dump?

   Not decided yet. Depends on system implementation

4. How can the data be accessed? (E.g. through the access to a DB? Which?)

Type of data model	Yes / No
Proprietary Data Base	Yes
PERFoRM dedicated DB (production, orders and test results)	No / Maybe
Other(s):	

5. Are you using any of the following "standardized/proprietary" way of representing data?

Type of data model	Yes / No
Proprietary data model	Yes
Ontology	No

Unspecified	Yes
Standardized data model	No
OPC-UA	No
BatchML (IEC 61512)	No
B2MML (IEC 62264)	No
XMPLant (ISO 15926)	No
CAEX (IEC 62424)	No
AutomationML (IEC 62714)	No
Other(s):	

6. Is it possible for you to describe your data structure? This can be very useful to understand which data we need to account for and in order to provide data translations into the PERFoRM data structure.

   Not yet; evaluation with factory are ongoing. Will be possible in the future

7. What IT infrastructure is present (or can be provided) for the PERFoRM system installation?

   Oracle and SQL via Ethernet
   SAP via Ethernet
   Manufacturing Bus systems (may not be necessarily used if all data is available via DB

Siemens specific questionnaire

1. Which type of database is used to collect the information of the Manufacturing Data Logging System?

   Oracle, maybe also accessible via OPC???
   SAP

2. Which type of database is used to collect the information of the Maintenance Ticketing System?

   SQL, SAP

3. Is there machine data that has to be collected which is not provided via the Manufacturing Data Logging System? How is this data accessible?

   Not known yet. Atm we have no additional data, but we might need to install additional sensors. Data access depends on implementation

4. Which is/are the model(s) of the PLC(s) used for controlling the processing machines? Which firmware is used?

   No PLC → SINUMERIK 840D → Firmware not known

5. Which Sinumerik controllers will be used? Which revision is used (e.g. PCU-50 for Sinumerik 840D Powerline)? Is it possible to get data access through OPC (-DA or -UA), for example through the OPC.SINUMERIK.Machineswitch?

   SINUMERIK 840D → Firmware not known
   Data access via OPC has to be evaluated. Ongoing with factory