Production harmonizEd Reconfiguration of Flexible Robots and Machinery

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

Deliverable 5.1

The PERFoRM Integration Approach

Lead Author: SIEMENS

Document Owner:	Siemens AG [A. Calà; N. Weinert]
Contributors:	- see List of Authors -
Dissemination:	Public
Date:	28.09.2017
Revision:	V1.0

Version history

Version	Date	notes and comments
0.1	04/12/2017	Deliverable structure, ToC , first input from SIEMENS
0.1	19.07.2017	Integrated parts from WHR, e-DISTRICT, POLIMI
0.1	28.07.2017	Integrated parts from IPB
0.1	10.08.2017	Technical Solution Simulation Environment integrated from SIEMENS
0.1	21.08.2017	Integrated content outline from TUBS on Agent Based Sim
0.2	28.08.2017	Description Siemens Demonstrator Case from SIEMENS + Integrated input from MTC
0.3	31.08.2017	Section 2.1 Overview from SIEMENS + Integrated input from LOCCIONI, UNINOVA, XETICS, TUBS
0.4	31.08.2017	Preparation of document for review cycle (SIEMENS)
0.4	07.09.2017	Integrated contribution from HSEL and IPB
0.5	12.09.2017	Included contribution from GKN and PARO; Included Review results from UNINOVA, LOCCIONI;
0.5	14.09.2017	Included Update from TUBS + Review Remarks (SIEMENS)
0.6	18.09.2017	Integrated review Feeback, Chapter 7, overall editing (SIEMENS)
0.7	26.09.2017	Pre-Final version (SIEMENS)
1.0	27.09.2017	Integrated contribution from GKN + Reference (SIEMENS)

Author List:

Ali Al-Yacoub (LBORO)
Giacomo Angione (Loccioni)
José Barbosa (IPB)
Chiara Bariatti (WHR)
Filippo Boschi (PoliMI)
Lennard Büth (TUBS)
Ambra Calà (Siemens)
Manfred Hucke (XETICS)
Gregorio Iuzzolino (IFEVS)
Pietro Perlo (IFEVS)
Giulia Lo Duca (Loccioni)
João Matos (UNINOVA)
Benjamin Neef (TUBS)
Birgit Obst (Siemens)
Phil Ogun (LBORO)
Mafalda Parreira-Rocha (UNINOVA)
Ricardo Peres (UNINOVA)
Pierluigi Petrali (WHR)
Mostafizur Rahman (MTC)
Johan Vallhagen (GKN)
Nils Weinert (Siemens)
Jeffrey Wermann (HSEL)

Abstract

Over the last decades several technologies have been developed towards the distributed automation architecture for smart factories but none of them are developed in an integration-ready form. The main objective of the PERFoRM project is to integrate and harmonize existing research and innovation results and prepare standards to facilitate industrialization and dissemination of distributed modular approaches, based on a decentralized automation architecture, according to the Industry 4.0 paradigm.

Within the first two years of the project, several technology solutions have been developed by the PERFoRM consortium, based on past experience, in order to meet the different Use Cases' needs and requirements. This Deliverable intends to provide a synchronization point by integrating and harmonizing the developed results in a unique solution adaptable to each Use Case. To this end, the complete set of PERFoRM technologies is listed and detailed, focusing on their current development status and test results.

Since the final goal is to implement and validate the integrated solution in industrial environments to establish the condition for flexible and reconfigurable manufacturing systems, the planned integrated solution for each Use Case as well as their As-Is situation are described in order to provide the knowledge necessary to Task 5.2 to elaborate a suitable migration plan towards the PERFoRM integrated solution.

This Deliverable refers to Task 5.1 "Integrative approach on innovative production systems" and contains contribution from many PERFoRM partners involved in WP5.

Table of Contents

List of Figures

List of Tables

1. Introduction

This deliverable is the first outcome of WP5 “Integration & Deployment planning”. WP5’s main goal is to define a vision for an integration and deployment of tools addressing self-adjustment, correction, control of individual machines and robots, and their connection with existing production planning and scheduling systems in order to demonstrate the flexibility and fast reaction of a manufacturing environment to the rapid market changes and increasing products complexity. The planned activities can be summarized to:

- Integration and harmonization of results from WP 2, 3 & 4;
- Identification of technologies to be further tested and validated within WP 6 before they can be applied within use cases;
- Detailed planning of migration strategies and deployment of innovative production systems within the four use cases;
- Centralized control of the migration progress towards innovative production systems across all use cases.

1.1. Objective of the Document

This document contains the outcome of Task 5.1 entitled “Integrative approach on innovative production systems”, which aims to harmonize and integrate concepts from work packages 2, 3, and 4 regarding the system architecture design, the standard interfaces and the middleware, as well as the robots/resources adaptors and the solution tools developed so far within the project, defining an overall picture of how the different developments can be integrated in one holistic approach and implemented in the four project’s use cases.

In parallel, the use case implementation goals have been defined in terms of new control architecture and related use case workflow, as well as the available production environment and legacy systems, together with work packages 7, 8, 9, and 10. This analysis is fundamental to identify the AS-IS and TO-BE situations of the use cases to develop a migration strategy in Task 5.2.

The objective is to define the PERFoRM approach for each use cases but also a general future view on the deployment process of new technologies after the PERFoRM project. Understanding what the long-term vision of the use cases is will help to define the optimal migration strategy towards the next generation of plug&produce production systems and their first migration step that will be implemented within this project.

The selection of the optimal migration strategy will be driven also by the analysis of the possible gaps, risks and obstacles that typically prevent manufacturers and decision-makers to implement new technologies and migrate towards the new distributed control architecture.

1.2. Structure of the Document

This document is structured as follows.

Chapter 2 provides an overview of the PERFoRM integrated approach describing how the solutions developed within this project are harmonized and integrated into one architecture solution. Then, each of these technology solutions is described focusing on what is their purpose and development status. In addition the goal of the solution within PERFoRM is explained and a cross evaluation of solutions with similar purpose is given.

Chapter 3, 4, 5, and 6 are dedicated to the project's use cases: Siemens, IFEVS, Whirlpool, and GKN respectively. The intention in these sections is to understand the starting point and define the goal of the migration for the different use cases. The target architecture implemented within PERFoRM represents a first migration step towards their long-term vision, namely what they intend to achieve in the future beyond this project.

Chapter 7 concludes the document with a summary of possible risks and obstacles that might influence the implementation of the PERFoRM architecture solution in industry, from the project Use Cases perspective, considering both technological and organizational aspects.

2. PERFoRM Integrated Approach

2.1. Overview

In the direction of flexible and reconfigurable production environments, PERFoRM developed a distributed and modular control automation architecture where the production components and applications interact by encapsulating their functionalities as services. To this end, the architecture is based on an industrial distributed service-oriented Middleware, in which the components can register their services and discover other components' services, enabling the communication between different systems and applications. Besides the integration of new computational tools, the architecture allows also the integration of legacy systems through the use of technology adaptors where legacy data formats are converted into PERFoRM compliant data.

Figure 1 represents the overall PERFoRM architecture, in which the tools developed within the project can directly interact via standard interfaces for services through the Middleware and the same for the legacy systems with the help of a specific technology adaptor and a wrapper. The picture shows the application classes and technology adapters required by the industrial use case partners, but the architecture functionality is not limited only to these components. The Middleware enables the integration and connection of other possible new tools or legacy systems by using, respectively, standard interfaces or new technology adaptors that expose their data in a PERFoRM compliant manner.

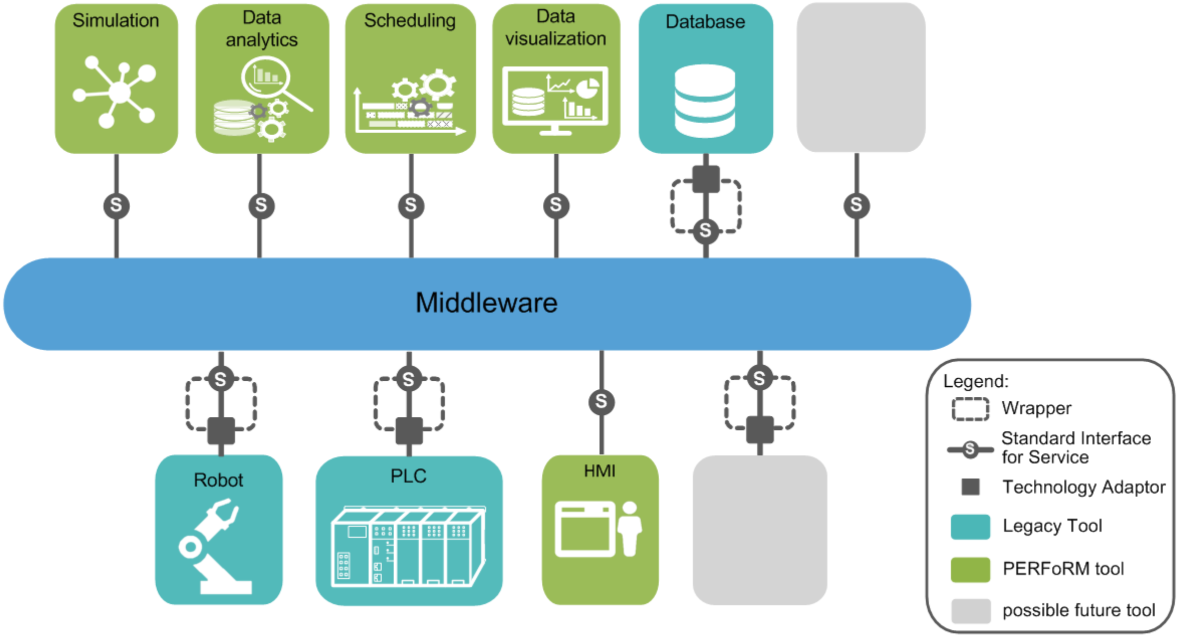


Figure 1: PERFoRM architecture

Currently, PERFoRM has developed technology adaptors for MS SQL and Oracle databases, PLCs (Programmable Logic Controllers), robots and sensors. Moreover, PERFoRM compliant tools for

Simulation, Data analytics, Scheduling, Data visualization and Human Machine Interfaces have been developed with standard interfaces. These technology adaptors, PERFoRM compliant tools and standard interfaces, as well as the Middleware, are described in section 2.2.

2.2. Technology solutions

2.2.1. Industrial middleware

2.2.1.1. Purpose of the solution

A Middleware is used in environments, where many different software applications need to work together. Since these different applications usually provide individual interfaces, making each communicate with each other can be a challenging task. For specific software it would be necessary to implement individualized interfaces to each of the other software components, which increases the necessary development exponentially with each new software inside the system. In these cases, a Middleware can act as a mediator between the applications. The applications now only have to provide one interface to the Middleware. The Middleware will then translate the information and provide it in the right syntax and semantics, which the receiving software is able to process. Figure 2 shows how the use of a Middleware reduces the amount of interfaces to be developed.

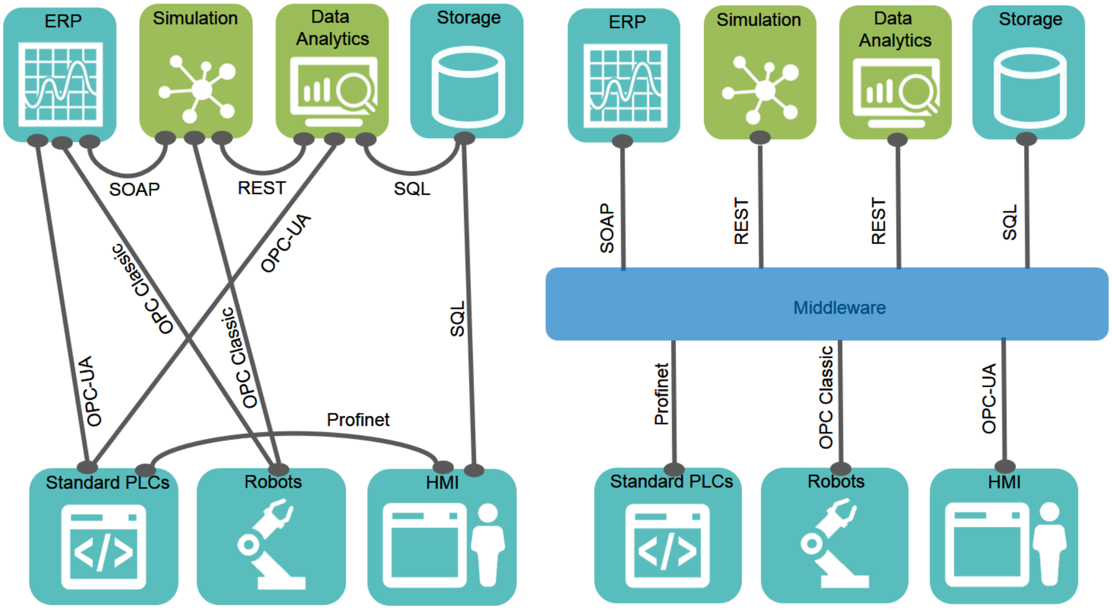


Figure 2: System without a Middleware (left) and with a Middleware (right)

By following this architecture, a lot of the complexity to establish the communication inside the system is shifted to the Middleware. It needs to provide the right interfaces to each connected component and additionally needs to be configured the right way to be able to process the information correctly. It should be an extendable system to allow the implementation of missing interfaces, where it is necessary.

Additional to just processing information and translating different protocols, a Middleware is also capable of aggregating and storing data, transform the data itself and provide additional security functionality. It decouples the sending and the receiving component, allowing the sending part to also send data, when the receiver is not available and preserving the data accordingly.

In an industrial application, these features are necessary to implement an Industrial Cyber-Physical System. Within this context, industrial components are becoming digitalized e.g. by adding intelligent software modules to machinery and need to communicate with each other to achieve an Intelligent Production System. A Middleware provides a platform to facilitate the integration of these cyber-physical components. Therefore, the Middleware is an integral part of PERFoRM's architecture.

2.2.1.2. Development goal in PERFoRM

The goal of this solution within PERFoRM is to provide a Middleware, based on existing technologies, which can fulfill the requirements of the four use cases. Because of the complexity of a Middleware system, it is not planned to develop a new Middleware system from scratch, but to evaluate the market to find fitting solutions and adjust them to meet the requirements of the architecture envisioned in PERFoRM. The focus is here to find solutions, which are also offering the necessary industrial support and is already proven to be used in industrial applications. To demonstrate the practical use of a Middleware, multiple instances are used in the demonstrators and the use cases.

As a first implementation step, the PERFoRM Middleware will be used as a basis to fulfill the different needs and requirements of the use cases. This means that the main goal is to find the right solution for each use case, detect existing gaps and propose necessary development to target these gaps, by developing missing pieces. The base for all implementations and development is the PERFoRM-compliant Middleware specification reported in Deliverable D2.4 [1].

2.2.1.3. Current state

In a first step, various existing Middleware solutions have been evaluated. Base for this evaluation were both general criteria and requirements derived from individual use cases. During this evaluation it became clear that there is not a single perfect solution, but a set of most promising solutions, which each are fitting to individual requirements.

Based on these results, the specification of a PERFoRM-compliant Middleware has been developed, which specifies, what a Middleware in the context of PERFoRM is and which functionality it needs to provide [1]. It further gives an overview about the evaluated software and suggests which to use for the individual requirements. Among many others, this can include the support of specific communication protocols or the Middleware as a distributed software.

Additionally, an instance of the Middleware based on Apache ServiceMix has been implemented and tested, which acts as a reference implementation. This installation is providing all the necessary functionality foreseen in the specification and can be used to evaluate the Middleware specification and PERFoRM architecture within the test beds. It lacks the industrial support, making it difficult to be applied in the use cases directly.

The individual use-cases are currently working on the exact definition of the Middleware to be used. Based on this, missing components will be identified and developed, to ensure that each solution is following PERFoRM's Middleware specification.

2.2.1.4. Test / Demonstration results

The Middleware has been tested and demonstrated in two iterations. The first use was presenting the Middleware specification and the Apache ServiceMix based implementation of the Middleware during project internal workshop, demonstrating the partners how the Middleware is working and how they are able to develop their own software which will be capable of communicating via the Middleware. During this workshop, a client software was developed, which was capable of retrieving data from a Database, using a Database Adapter and the PERFoRM Data model.

In addition to this, another test case has been built at the laboratories at MTC, which was demonstrated during PERFoRM's M18 meeting. In this test case, a CAD file coming from a FTP server is processed to a robot adapter using the ServiceMix based Middleware.

2.2.2. Data model and standard interfaces

2.2.2.1. Purpose of the solution

The interoperability of the elements presented in industrial environments is essential in Industry 4.0. Standard Interfaces are the key to ensure this interconnection of heterogeneous legacy hardware devices and software applications.

These standard interfaces should define in a unique, standard and transparent manner how the devices and applications must interact with each other, ensuring the transparent pluggability of these devices. For this reason, the interfaces should adopt a standard data representation and define the list of services provided by it.

The Data Model and the Standard Interfaces were designed in alignment with the use cases and their particular requirements, with two different interfaces being considered, one defining the interaction from middleware to high-level, and the other connecting the middleware to shop-floor.

In relation to the Data Model, its design was focused on being as generic as possible in order to be capable of representing both high-level and shop-floor information across a wide array of use cases. Such a generic Data Model represents a good solution to allow a great level of scalability, being able to be used in varied scenarios.

2.2.2.2. Development goal in PERFoRM

During Task 2.3 it was defined that from the main results should be a generic Data Model that can represent both high-level and shop-floor information, as well as two Standard Interfaces.

The results from the task were presented in a workshop attended by several project partners, from which a scientific publication was made, submitted and presented at FAIM 2017 [2].

2.2.2.3. Current state

The development part of task 2.3 comprised the design and implementation of the Data Model and Standard Interfaces, which were concluded in M16.

Despite the task being officially concluded and documented in Deliverable 2.3 in detail, continuous support has been given to the maintenance of the Data Model, in order to support the developments mainly from WP4. The current stage of Data Model is presented in Figure 3:

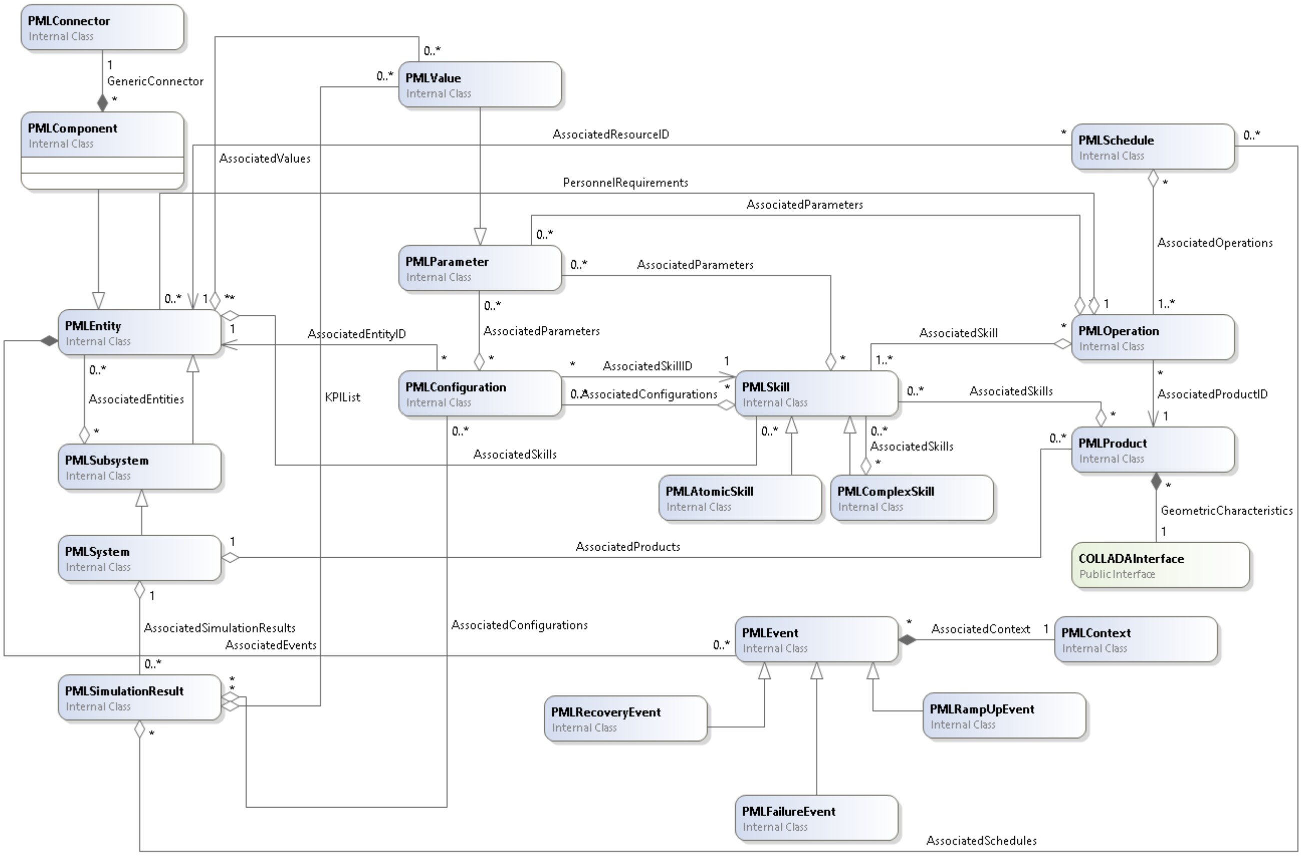
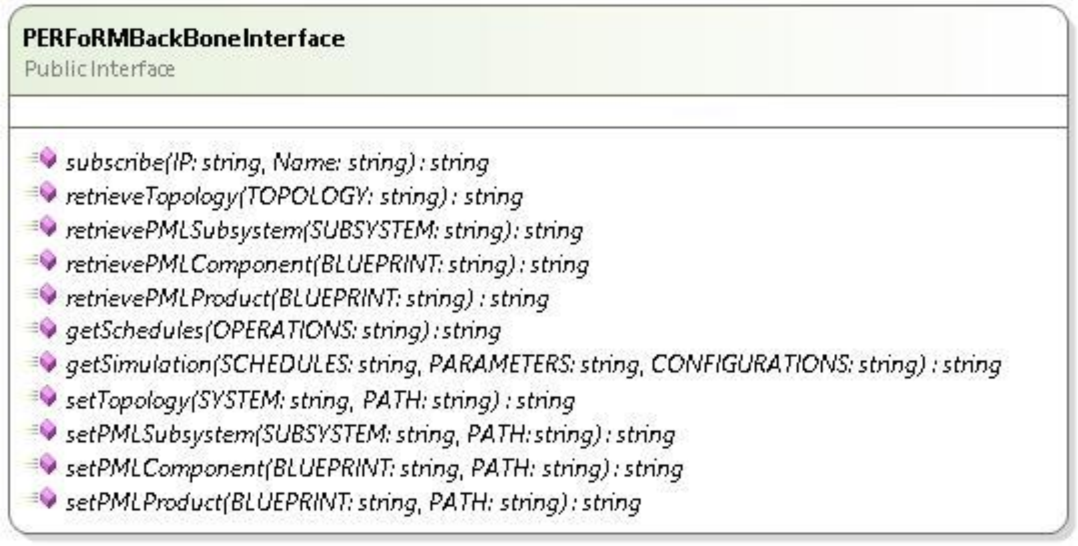


Figure 3: PERFoRM's Common Data Model

The presented Data Model was implemented using AutomationML. This format was chosen after a study done in which several available data exchange standards and formats were assessed, which is described in further detail in D2.3.

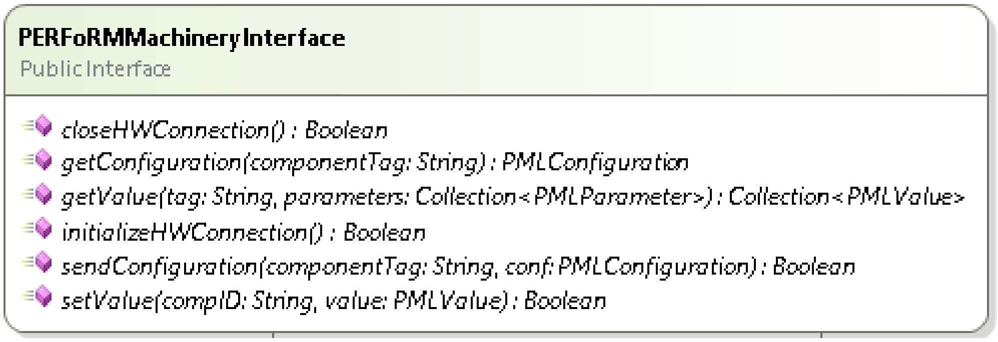
The two Standard Interfaces implemented are shown in Figures Figure 4 and Figure 5: .



PERFoRMBackBoneInterface
Public Interface

- *subscribe(IP: string, Name: string) : string*
- *retrieveTopology(TOPOLOGY: string) : string*
- *retrievePMLSubsystem(SUBSYSTEM: string): string*
- *retrievePMLComponent(BLUEPRINT: string) : string*
- *retrievePMLProduct(BLUEPRINT: string) : string*
- *getSchedules(OPERATIONS: string) : string*
- *getSimulation(SCHEDULES: string, PARAMETERS: string, CONFIGURATIONS: string) : string*
- *setTopology(SYSTEM: string, PATH: string) : string*
- *setPMLSubsystem(SUBSYSTEM: string, PATH: string) : string*
- *setPMLComponent(BLUEPRINT: string, PATH: string) : string*
- *setPMLProduct(BLUEPRINT: string, PATH: string) : string*

Figure 4: PERFoRM's Backbone Interface



PERFoRMMachineryInterface
Public Interface

- *closeHWConnection() : Boolean*
- *getConfiguration(componentTag: String) : PMLConfiguration*
- *getValue(tag: String, parameters: Collection<PMLParameter>) : Collection<PMLValue>*
- *initializeHWConnection() : Boolean*
- *sendConfiguration(componentTag: String, conf: PMLConfiguration) : Boolean*
- *setValue(compID: String, value: PMLValue) : Boolean*

Figure 5: PERFoRM's Machinery Interface

Figure 4 presents all methods needed for the communication between the middleware and the software applications connected at the high level. There are several setter and getter methods so that the tools can access/change the topology or some element or value present in the system, as well as methods to get results from specific types of tools such as *getSchedules* and *getSimulation*.

Similarly, Figure 5: lists the necessary methods for the interactions between the middleware and the shop-floor. This interface exposes methods like initializeHWConnection, that will start a connection to a hardware device, or getConfiguration, that returns the configuration associated with some component.

2.2.2.4. Test / Demonstration results

A workshop was organized by UNINOVA, IPB, HSEL and Loccioni in order to familiarize the remaining partners with the results from WP2 and T3.1, which included the Data Model in AutomationML and the Standard Interfaces. In this workshop, a description of the Data Model and Standard Interfaces was given, along with a small practical session regarding the instantiation of the Data Model itself in a running system. From this workshop resulted a scientific publication [1], which has been presented at FAIM'17.

Unfortunately the instantiation and testing in the demonstrators weren't possible before the conclusion of T2.3 due to the different timings associated with the use cases, thus these will be done at a later date.

2.2.3. Technology adaptors

2.2.3.1. Oracle DB Adaptor

2.2.3.1.1. Purpose of the solution

The Oracle DB Adapter permits to connect a legacy Oracle database to the PERFoRM middleware. In the PERFoRM project, an Oracle database is used in the Siemens scenario for collecting production data about the machining process of compressors. Data contained in the Oracle database are fundamental inputs for the predictive maintenance and scheduling tools developed in the PERFoRM project.

2.2.3.1.2. Development goal in PERFoRM

The role of the adapter is to retrieve the values from the Oracle database and translate them according to the PERFoRM's data model. In particular, the adapter performs the following activities:

1. Implement the PERFoRM's standard interface and methods.
2. Connect to the database.
3. Send queries and update statements to the database.
4. Retrieve the results of the queries from the database.
5. Transform the results according to the PERFoRM's data model.

The connection to the database is facilitated by the use of the JDBC (Java Database Connectivity) interface that provides a transparent connectivity to heterogeneous databases. Since the use case database is Oracle, the corresponding driver is used to access to the data source.

The Java code of the database adapter can be easily transformed into a web service using Apache CXF. The creation of the REST based service allows a smooth integration of the adapter into the PERFoRM's middleware, which takes care of the conversion of Java objects to and from JSON strings and permits the service registration and discovery.

2.2.3.1.3. Current state

The prototype implementation of the Oracle DB Adapter is complete and working. Future steps are devoted to the deployment and the assessment of the adapter into the Siemens scenario.

2.2.3.1.4. Test / Demonstration results

A demonstration of the database adapter was held during the workshop in Braganca (Portugal) on the 24th and 25th of November 2016. The results achieved are described also in a paper presented at the FAIM 2017 conference [1].

2.2.3.2. MS SQL DB Adaptor

2.2.3.2.1. Purpose of the solution

Similar to the Oracle DB Adaptor described in section ###, an adaptor to connect to legacy Microsoft SQL-Databases is developed. The adaptor is required to retrieve data from existing databases and provide it to tools and services through the middleware, using the PERFoRM-ML-Format. For demonstration, the MS-SQL-Adaptor will be used in the Siemens and Whirlpool Use Cases to connect with the existing Maintenance Management system.

2.2.3.2.2. Development goal in PERFoRM

Also very similar to the Oracle DB Adaptor, the Microsoft SQL Adaptors main purpose is to read data from the Database and send it back to the requesting service. While the DB request has to be conducted in the native data formats and structures, the response to the requesting service will be compiled in a PERFoRM compliant data model. Therefore, the development goal is to provide the software functionalities required in one tool, running as a connector service within a PERFoRM environment. The service will have to be parametrized at set up in order to assign a mapping of the PERFoRM data models to the legacy ones. During runtime, the services will receive request from other services via the middleware, translate the request into native MS SQL Queries, query the database and respond to the requesting service after translating the native result into a PERFoRM data format.

2.2.3.2.3. Current state

The MS-SQL-Adaptor has been defined in general, adaptations required for tailoring the solution for the Siemens Use Case are made within WP7, for the Whirlpool Use Case in WP9, respectively. The implementation also is ongoing in WP7/WP9.

2.2.3.2.4. Test / Demonstration results

The MS-SQL-Adapter will be tested in the integration test for the Siemens Use Case in WP6 and WP7.

2.2.3.3. Robot Adaptor

2.2.3.3.1. Purpose of the solution

The robot adaptor permits to connect a legacy robot to the PERFoRM middleware for sending commands/trajectories and receiving status information (positions, alarms, etc.). The adapter permits to forget about the specific programming language and low-level software library used to communicate with the robot. The purpose is to generate robot commands from a user-friendly interface which does not require a robotic expertise (e.g. CAD software tool). In particular, the implementation done in the PERFoRM project considers the requirements coming from the Leak Robot Station in the Whirlpool use case. In this scenario, a robot (UR10) equipped with a probe is used to detect microwaves leakages. The robot moves the probe following a predefined path around the microwave oven. If leakages are detected the oven is send for reparation.

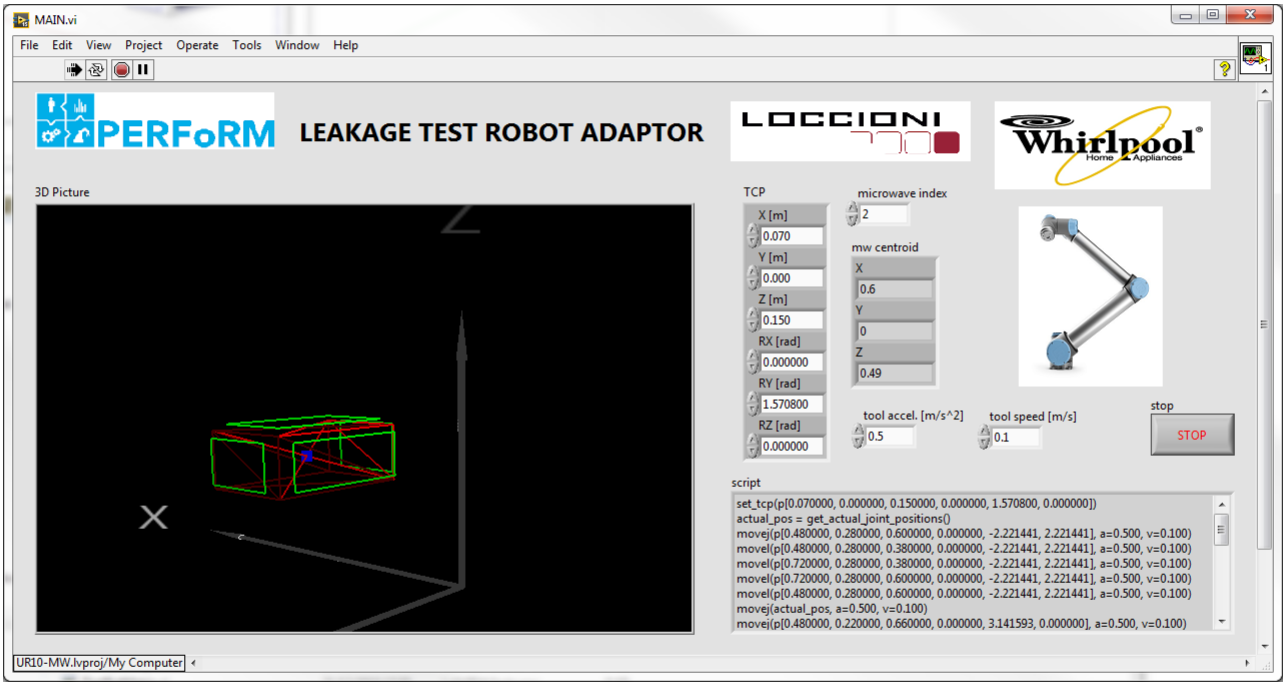
In this use case, the role of the adaptor is to automatically generate the robot commands starting from a path drawn with a CAD tool. This permits to Whirlpool product designers to directly draw the path which is more appropriate for detecting possible microwave leakages outside the oven. Moreover, it allows not to stop the production line when a new model of oven is being produced for teaching the new path to the robot.

2.2.3.3.2. Development goal in PERFoRM

For generating the robot commands, the adapter requires as input a COLLADA file containing the 3D models of the microwave, the robot and the desired path of the probe (COLLADA is part of AutomationML and compliant with PERFoRMML data model).

Since the robot of the use case is an UR10, the adaptor generates an URScript code, but this solution can be extended to generate the code of other robot programming languages (e.g. for Denso, Kuka and ABB robots).

The following figure shows the Graphical User Interface (GUI) of the Robot Adaptor developed using LabVIEW. The 3D picture displays the 3D models contained in the COLLADA file, while the controls permit to set the Tool Center Point (TCP) and the acceleration and the speed of the tool (the TCP represents the position of the probe respect to the manipulator turning disk). On the bottom-right of the GUI is shown the URScript which is automatically generated and can be sent (using a TCP/IP socket connection) to the robot controller for the execution.



TCP
X [m] 0.070
Y [m] 0.000
Z [m] 0.150
RX [rad] 0.000000
RY [rad] 1.570800
RZ [rad] 0.000000

microwave index 2

mw centroid
X 0.6
Y 0
Z 0.49

tool accel. [m/s^2] 0.5

tool speed [m/s] 0.1

script

```
set_tcp(p[0.070000, 0.000000, 0.150000, 0.000000, 1.570800, 0.000000])
actual_pos = get_actual_joint_positions()
movej(p[0.480000, 0.280000, 0.600000, 0.000000, -2.221441, 2.221441], a=0.500, v=0.100)
movel(p[0.480000, 0.280000, 0.380000, 0.000000, -2.221441, 2.221441], a=0.500, v=0.100)
movel(p[0.720000, 0.280000, 0.380000, 0.000000, -2.221441, 2.221441], a=0.500, v=0.100)
movel(p[0.720000, 0.280000, 0.600000, 0.000000, -2.221441, 2.221441], a=0.500, v=0.100)
movel(p[0.480000, 0.280000, 0.600000, 0.000000, -2.221441, 2.221441], a=0.500, v=0.100)
movej(actual_pos, a=0.500, v=0.100)
movej(p[0.480000, 0.220000, 0.660000, 0.000000, 3.141593, 0.000000], a=0.500, v=0.100)
```

Figure 6: Robot Adapter GUI

A possible integration of the robot adaptor inside the PERFoRM architecture is described in Figure 7 and consists of the following components:

- File Server: hosts the CAD files for performing the robot leak test
- Robot Service: fetches a particular CAD file from the File Server and push it to a folder that the UR adaptor can access
- Operator GUI: reads the list of CAD files from the file storage location, displays them and gives to an operator the option of running a particular file
- Middleware VM: hosts PERFoRM’s middleware architecture that comprises of an Apache Service Mix

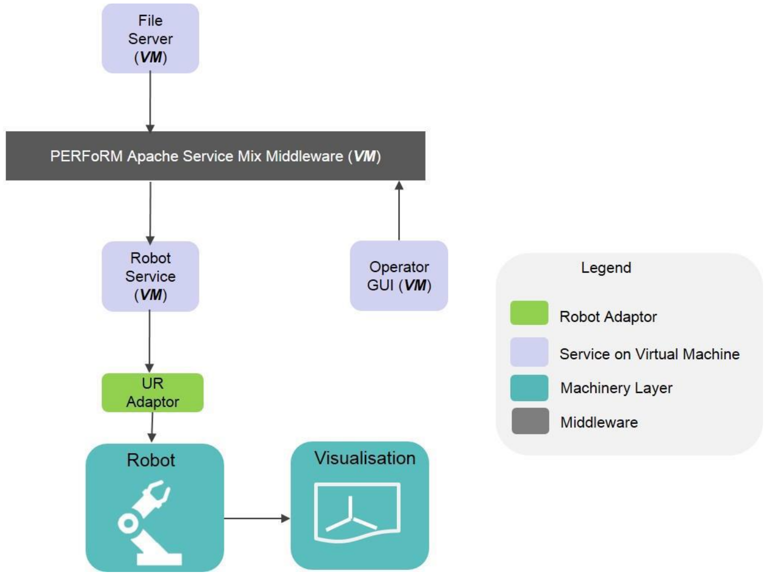


Figure 7: Integration of the robot adaptor inside the PERFoRM architecture

CAD files are stored on the File Server. When the operator selects a file through the GUI, the Robot service fetches the CAD file and places it in a specific folder through the PERFoRM Apache Service Mix Middleware. The UR robot adaptor, which is monitoring the folder, is then able to fetch the file and generate the UR script. Finally, the UR script is sent to the robot.

For testing purposes, another tool (developed by the MTC) is used to record the linear positions and rotation coordinates of the robot. The path that the robot has followed is compared to the path that it was supposed to follow and the results are visualized by a script written in Matlab.

2.2.3.3.3. Current state

The prototype implementation of the Robot Adapter is complete and working. Future steps are devoted to the deployment and the assessment of the adapter into the Whirlpool scenario.

2.2.3.3.4. Test / Demonstration results

A demonstration of the robot adapter was held during the M18 review meeting in Coventry (UK) on the 5th and 6th of April 2017. The results achieved are described in a paper presented at the INDIN 2017 conference [3].

2.2.3.4. Sensor Adaptor

2.2.3.4.1. Purpose of the solution

The Sensor adaptor permits to connect a legacy sensor to the PERFoRM middleware for sending commands/parameters and receiving measurement results and status information. The adapter permits to forget about the specific protocol, communication bus and low-level software library used to communicate with the sensor. The purpose is to manage the sensor and retrieve data offering a common software interface (e.g. OPC-UA). In particular, the implementation done in the PERFoRM project considers the requirements coming from the Roughness Process in the GKN use case. The goal is to substitute the wired serial communication of the Mitutoyo roughness sensor in order to facilitate the robot movements and the information exchange.

For the first objective (facilitate the robot movements) a serial-wifi converter powered by a battery is used, while for the second objective (facilitate the information exchange) an OPC-UA server is created.

2.2.3.4.2. Development goal in PERFoRM

The implementation of the sensor adapter has required both hardware design and software development.

From the hardware point of view, the solution proposed consists in a box, designed and realized with the 3D printer, that contains the converter, a battery (Li-Ion 7.2 V, it covers more than 8 hours), two electronic circuits: one for recharging purposes and the other to protect the load from short circuits. The box was designed to have, obviously, a hole for the antenna, a switch to start charging the battery and the space for the RS 232 connection and for the jack for the power supply (9-12 V). It also has little holes on the top to show led on the converter, in order to understand if the converter is ready to communicate, and next to the chip to monitor the charging process. On the bottom of the box there are holes for screwing it on the robot arm. In Figure 8, a preliminary scheme of the solution and result of the mechanical and electrical design are shown.

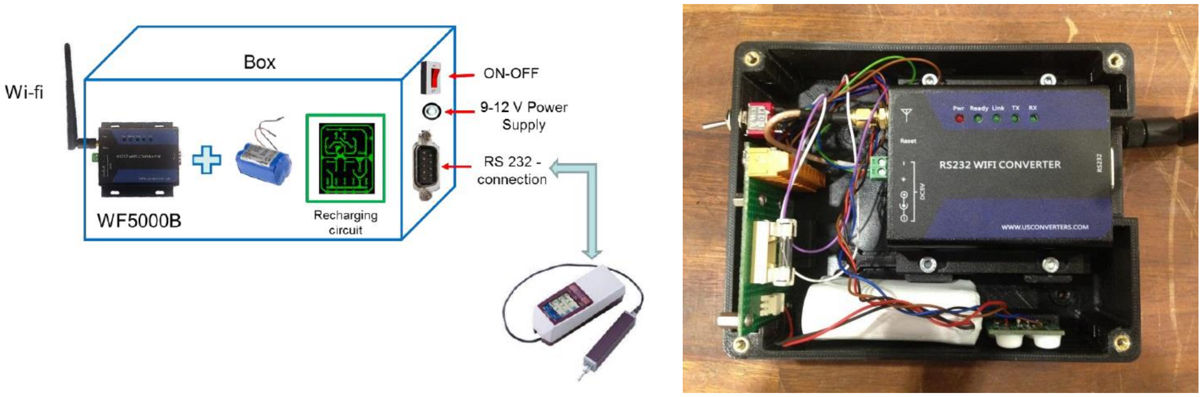


Figure 8: Sensor Adapter Hardware Implementation

From the software point of view, an OPC-UA server has been implemented together with a GUI (Figure 9) which permits to give commands to the sensor and to read measuring results and parameters from the sensor.

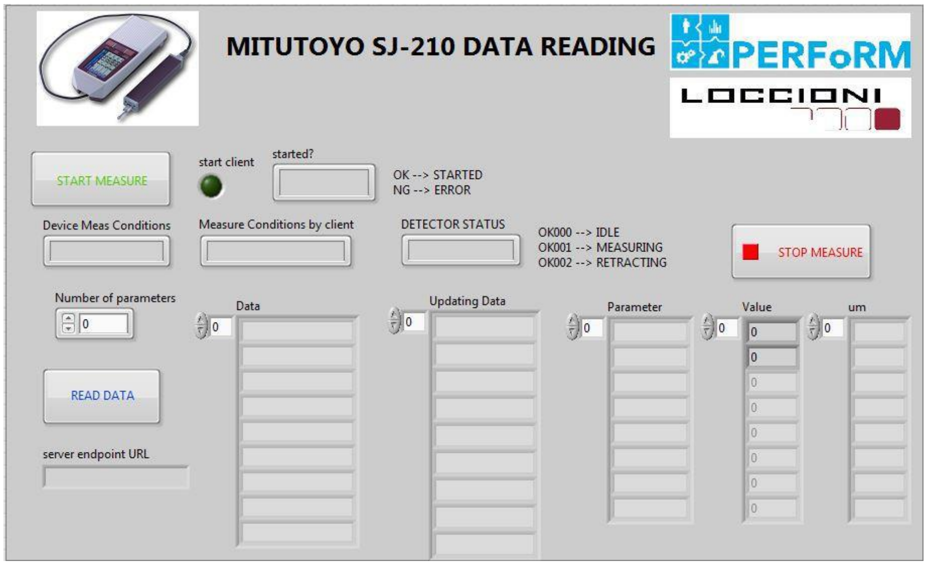


Figure 9: Sensor Adapter GUI

The OPC-UA server created can be easily integrated into the PERFoRM middleware architecture by using OPC-UA client application.

2.2.3.4.3. Current state

The prototype implementation of the Sensor Adapter is complete and working. Future steps are devoted to the deployment and the assessment of the adapter into the GKN scenario.

2.2.3.4.4. Test / Demonstration results

A demonstration of the robot adapter was held during the M18 review meeting in Coventry (UK) on the 5th and 6th of April 2017.

2.2.3.5. PLC Adaptor

2.2.3.5.1. Purpose of the solution

The PLC Adaptor provides the functionality required to connect legacy PLCs to PERFoRM Systems. While the implementation for PERFoRM will mainly be tailored for the GKN Use Case, the general requirements towards a general adapter will be met.

2.2.3.5.2. Development goal in PERFoRM

The PLC adaptor 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 adaptor. With this library, it is possible to send and receive a data structures consistent to a C# class with the same data structure. This communication is very fast and data safe. On the other side, the adaptor provides PERFoRM-conform standard interface, allowing for connections to and from the middleware, or respective other services accessible via the middleware.

2.2.3.5.3. Current state

The PLC Adapter is still being developed.

2.2.3.5.4. Test / Demonstration results

The PLC Adapter is still being developed. Specific test have not yet been made.

2.2.4. Real-time process information

2.2.4.1. Purpose of the solution

The proposed solution is responsible for not only performing the context-aware data analysis, generating predictive data that can be used to trigger the system's self-adjustment mechanisms (e.g. reconfiguration), but also for the acquisition of the data itself from the shop-floor. As such, T3.2 can be classified similarly to the technology adaptors, being however focused on the real-time aspects, serving as a soft real-time connector for tools that might operate under this time constraints.

Several requirements were imposed on the solution’s design. Aligned with PERFoRM's vision, the solution’s architecture should be generic enough to be applicable to various different scenarios, being completely decoupled and independent from the existence of a single communication protocol or standard on the shop floor, thus facilitating its industrial integration and adoption.

Moreover, it needs to be capable of adapting to changes to the process or its components in run-time, for instance in terms of both pluggability and changes to the Key Performance Indicators (KPI) to be analyzed. Furthermore, data and context representation should follow PERFoRM's common data model in order to enable the seamless interoperability and data exchange between the data analysis architecture and the remaining PERFoRM system elements and tools.

Finally, there’s the aspect of scalability. In order to ensure that the approach is applicable to a varied number of different use cases, it needs to be capable of scaling according to each use case requirements. As a system scales its complexity tends to escalate as well, as such, to tackle this challenge a layered solution was developed.

2.2.4.2. Development goal in PERFoRM

Within the lifespan of Task 3.2, during the various alignment meetings that were held, it was decided that the development goal to be achieved was to provide a framework shell of the real-time data acquisition and processing solution. This would in turn serve as a guideline for the integration and implementation of use-case specific instantiations, with all the requirements and functionalities of the different elements comprised in the solution clearly identified.

However, the results of Task 1.3 ended up going a bit further, with an example implementation of the entire framework being showcased, resulting in a scientific publication and a video demonstration which was used for the dissemination of the project’s results after M18. Being a fully decoupled solution, it can also be later on be adjusted and further refined. For instance, in regards to the data processing component several adjustments can be made as needed, such as the variation of the data processing parameters, the algorithm or even the technology used for its implementation. This can allow predictions to be either faster or slower to respond to changes in the incoming raw data values, or to mitigate possible challenges such as false positives and late detections.

2.2.4.3. Current state

The design and implementation of this solution were concluded in M13 as part of the developments related to Task 3.2. The requirements and main functionalities of the solution's different components are all documented in D3.2. The proposed implementation encompasses all the different layers shown in Figure 10.

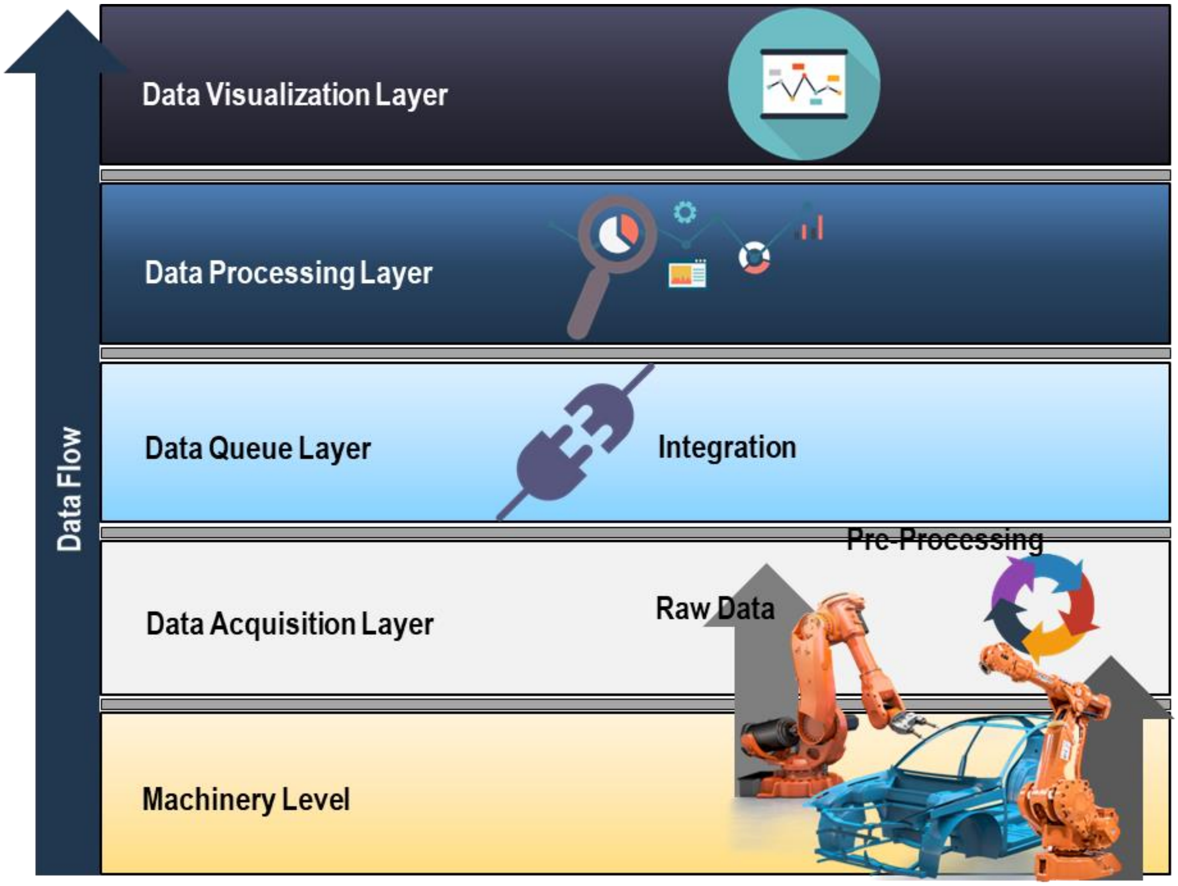


Figure 10: Solution's Layers

From the implementation of the Data Acquisition Layer resulted a Cyber-Physical System based on a Multiagent approach, while the Data Queue Layer and the Data Processing Layer were implemented through the usage of Apache Kafka and Apache Storm, respectively.

Even though the data visualization layer was not initially planned to be part of the solution, a Java-based tool was implemented in order to facilitate the observation of the results as time-series charts. This tool dynamically updates each chart in real-time when new data samples are collected or generated, as shown in Subsection X. This makes it easier to visually understand and interpret the data being outputted by the processing network, allowing it to be used for instance at the shop-floor level as a Human-Machine Interface (HMI) to support operators in run-time.

2.2.4.4. Test / Demonstration results

Since Task 1.3 concluded relatively early in the project's timeline, the validation and demonstration of its results was done internally in two distinct phases. Firstly, some preliminary tests were conducted in the laboratory with the intent to verify the executing and integration of the different layers comprised in the solution.

For this purpose, the behavior of a shop-floor gripper was emulated in order to generate test sensor data. These data stream was then passed through the different layers of the solution, with the Data Processing Layer outputting more complex processed values, such as opening and closing timespans. The preliminary tests were conducted in three different modes: Normal Operation, Spike Failure and Incremental Failure. Using the Java-based visualization tool, the results shown in Figure 11 were obtained for normal operation conditions:

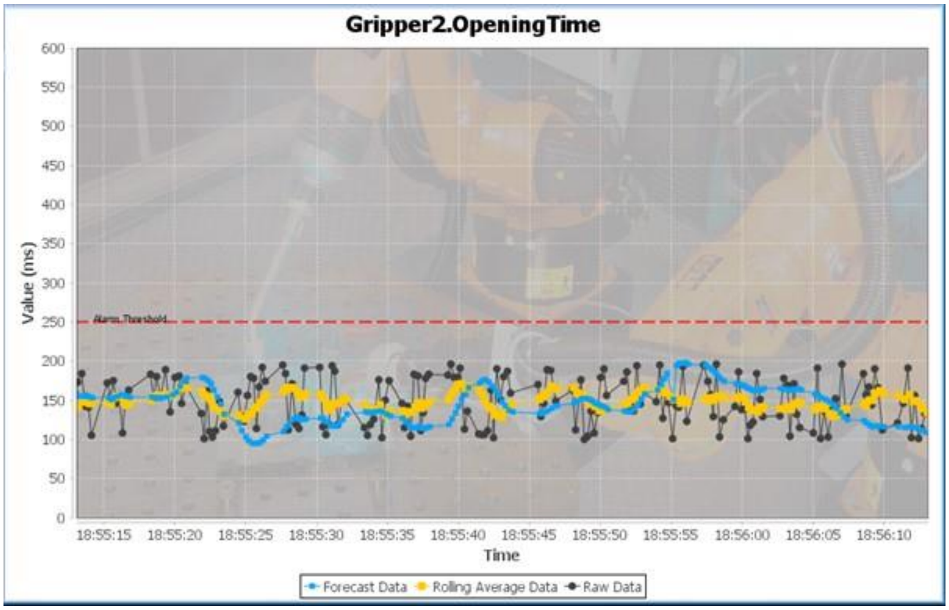


Figure 11: Preliminary Test Scenario - Normal Operation Conditions

As seen in Figure 11 three different data types are charted. The dark grey line represents the raw values coming directly from the data source, with the values ranging from 100 to 200 milliseconds. The rolling average can be seen in yellow, representing and smoothed dataset, eliminating outliers and preparing the incoming data points to be processed. Finally the forecast is shown in blue, depicting the predicted value of the *y* variable a certain number of data point entries into the future.

Upon the induction of a spike failure, the corresponding output of the solution can be seen in Figure 12:

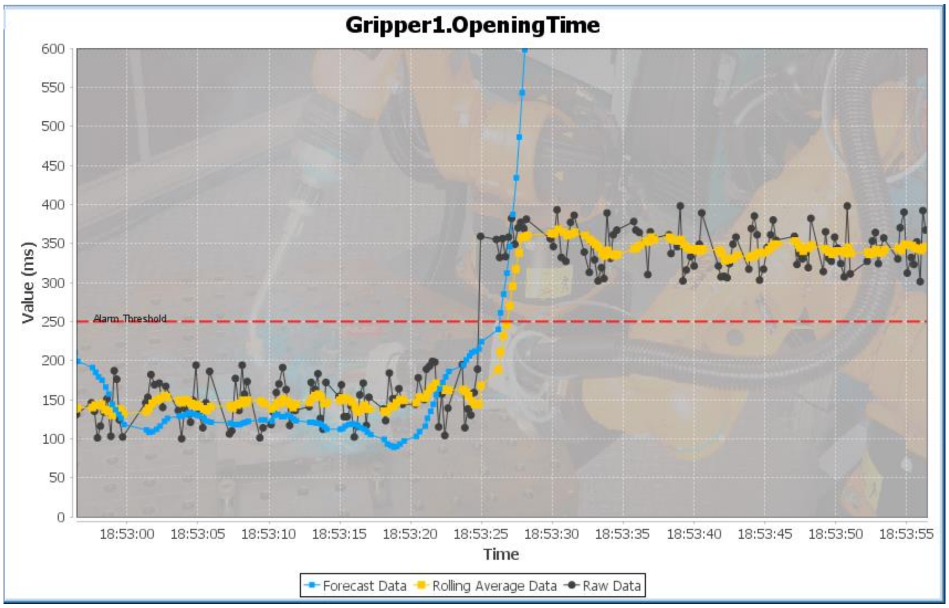


Figure 12: Preliminary Test Scenario - Spike Failure Mode

As it can be observed, the forecast values skyrocket due to the way the least squares model is estimated. This is not an ideal application scenario since there was no trend in the data that would allow the failure to be predicted. Still, this occurrence can be detected due to the forecast exceeding the alarm threshold, causing an alarm event to be emitted, triggering a possible self-adjustment or maintenance response.

A better suited scenario would be the case where there is an underlying trend in the data being analysed which can be used to anticipate such a failure or machine breakdown event. This is the case of the incremental failure mode, depicted in Figure 13.

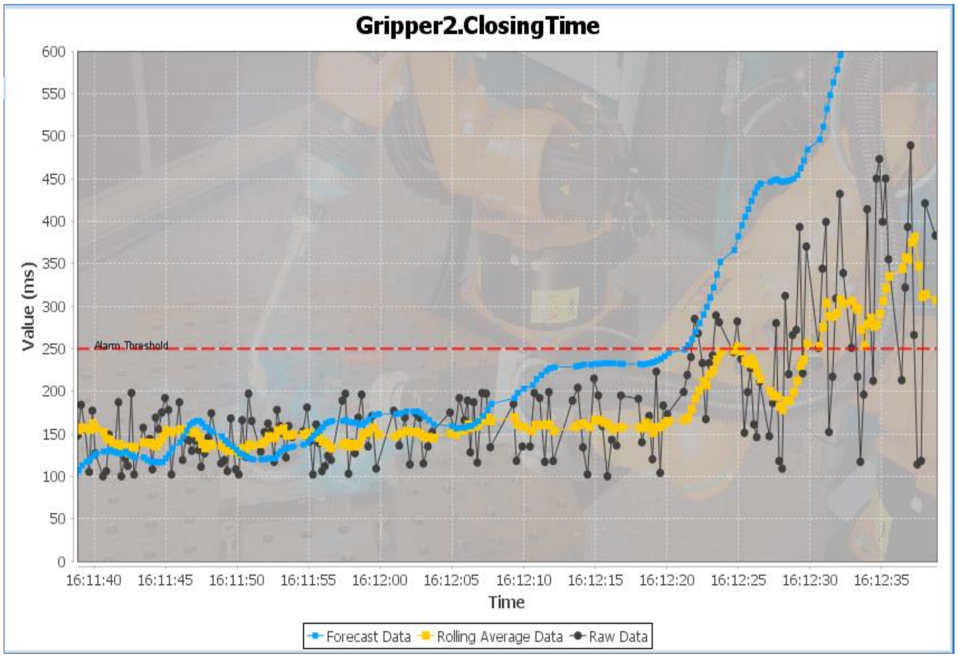


Figure 13: Preliminary Test Scenario - Incremental Failure Mode

In this case, there is no spike in the raw data values generated by the emulated data source. Instead, a gradual increase in the observed values can be detected, thus presenting a rising trend in the data. Based on this, the Data Processing Layer can predict this response and signal or trigger an intervention before a breakdown occurs.

Finally, the second phase consisted in integrating this solution with a possible implementation of the overall PERFoRM architecture, in order to showcase its feasibility to interoperate within the PERFoRM ecosystem.

For this purpose, a simulated test bed was developed in V-REP in order to emulate a simple shop-floor cell consisting of two UR5 robots, two conveyor belts, a part source and a sink. This cell performed a welding operation on every piece passing through the first conveyor, which was then picked and place onto a secondary line leading to the part sink. A overview of the simulation can be seen in Figure 14.

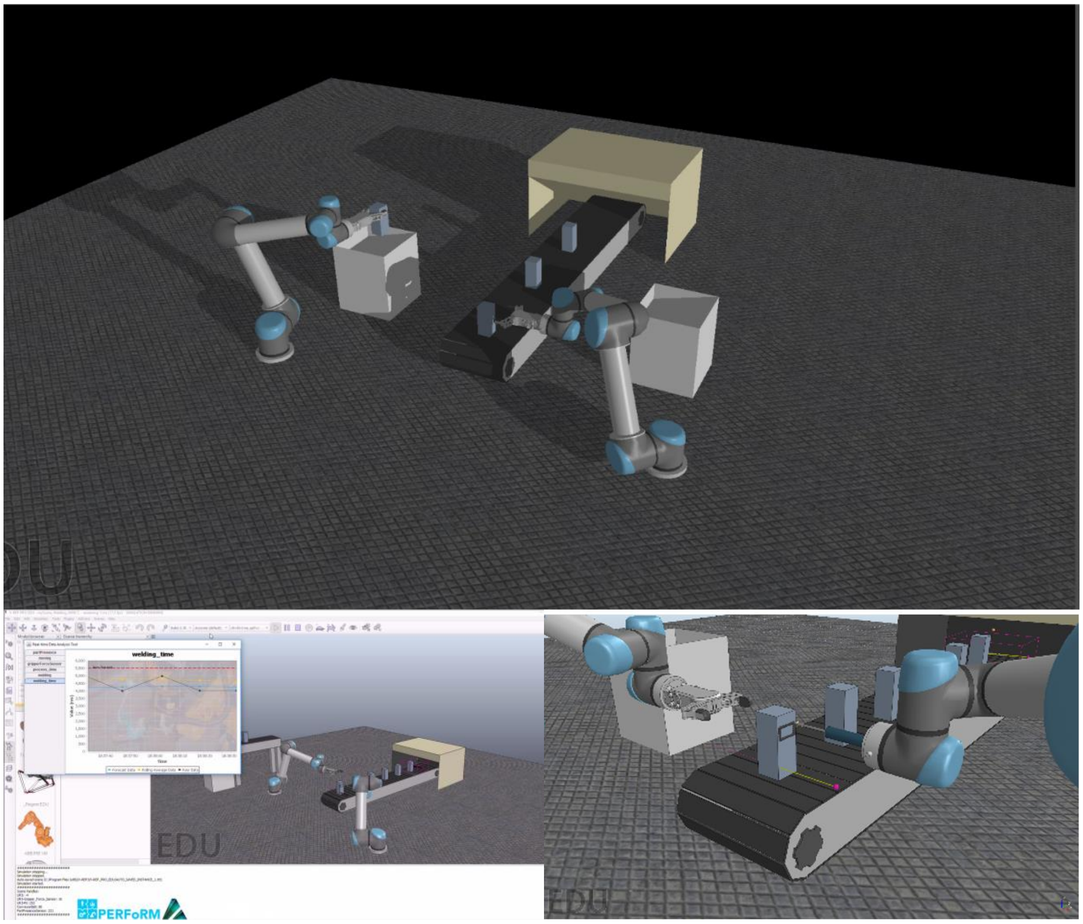


Figure 14: Demonstration of T3.2's results using a simulated test bed

The results from this second preliminary test phase culminated in a demonstration video elaborated by UNINOVA encompassing the integration of the results from Task 1.3, a middleware (WINCC OA) and the data model resulting from Task 2.3 using AutomationML. This video is hosted on YouTube and was disseminated through several social media platforms, including the project's Facebook and Twitter pages.

2.2.5. Human observation

2.2.5.1. Diagnostic and Maintenance Reporting Interface

2.2.5.1.1. Purpose of the solution

The performance of machine learning models depends on the veracity of the training data. Currently, the system used for capturing data does not enforce structured and consistent reporting of machine faults and failures by the operators. Alarm reports from operators are often vague and not informative. Different operators might also report the same event in completely diverse ways, which makes it difficult to draw lexical semantics from the data using natural language processing techniques. The purpose of this solution is to develop an interface for reporting machine failures and faults in a much more structured, consistent and machine interpretable format.

2.2.5.1.2. *Development goal in PERFoRM*

Based on the work presented in D3.4, it was noticed that the accuracy of the data-driven models depends on the operator’s textual feedback of different faults and failures. However, operates textual feedback is not informative all the time. Therefore, the goal in PERFoRM is to develop a software interface for reporting machine failures and faults on the shop-floor. goal is to force operators to structure their observations using graphical interface by clicking on the sub-system that is the root cause of a failure or fault. Additional comments can also be provided in a much more structured and consistent manner.

2.2.5.1.3. *Current state*

An information model for capturing maintenance activities has been created, although it is not reported in any of the deliverables. The information captured include failure modes, root causes and failure observables and failure events. A web interface was also designed for reporting maintenance data as part of the solution for the changeover management. Figure 15 shows an overview of the system.

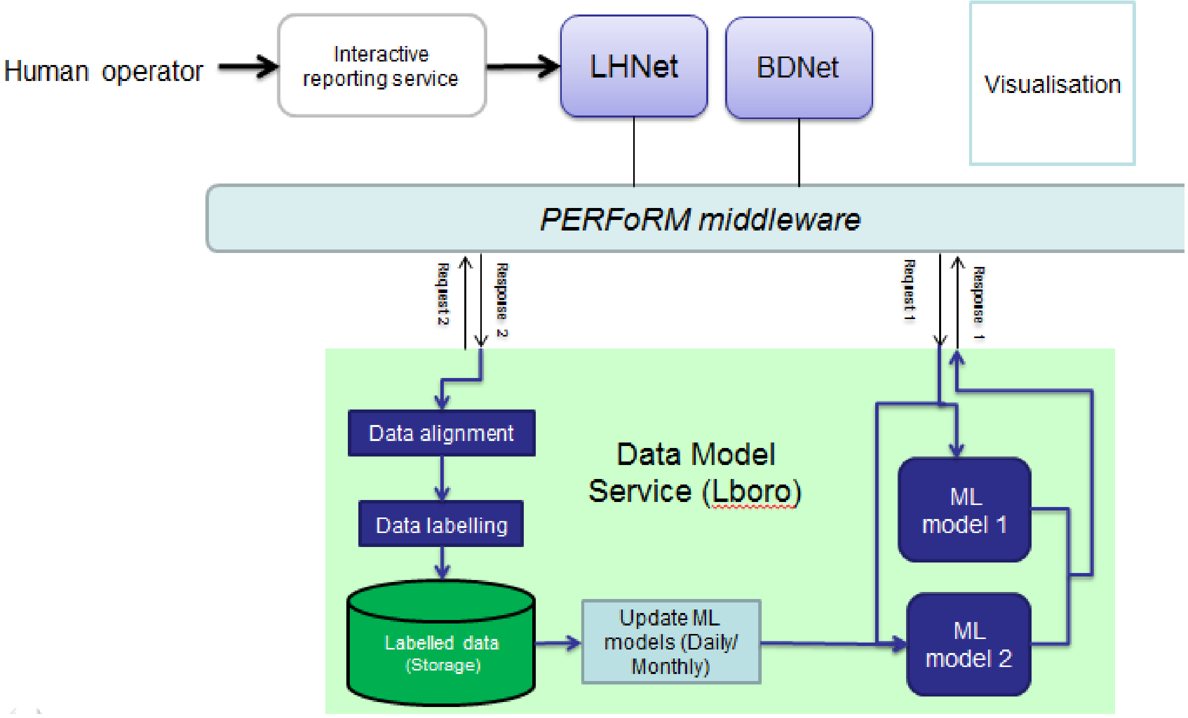


Figure 15: Human operator observation

The functionality of the developed software has been tested. However, it needs to be deployed in a shop floor environment where operators will be able to use it to report failures. The solution is well aligned to the Siemens use case. The plan is to compare the accuracies of the models derived from the new data with the ones reported in D3.4.

2.2.5.1.4. Test / Demonstration results

No test or demonstration results have been achieved yet.

2.2.5.2. Change Over Management

2.2.5.2.1. Purpose of the solution

In today’s manufacturing, the proliferation of product variants necessitates shorter production runs, which leads to more frequent changeovers. Rapid increase in product quality and quantity demand increases the pressure to maximise capacity utilisation. Therefore, a quick changeover process is very critical to the production of small batch sizes of a large diversity of products without loss in productivity.

Studies have shown that the effectiveness of a changeover process depends largely on the knowledge of the operators carrying out the changeover on the shop floor. In the absence of standard procedural guidelines, operators use their judgment and experience to select the best sequence of actions for the changeover process. Human involvement in the decision-making process introduces variabilities, which make the process to be stochastic and unpredictable.

As seen in Figure 16, changeover process consists of three phases. The purpose of this solution is to create formal models for the set-up and ramp-up phases of the entire process.

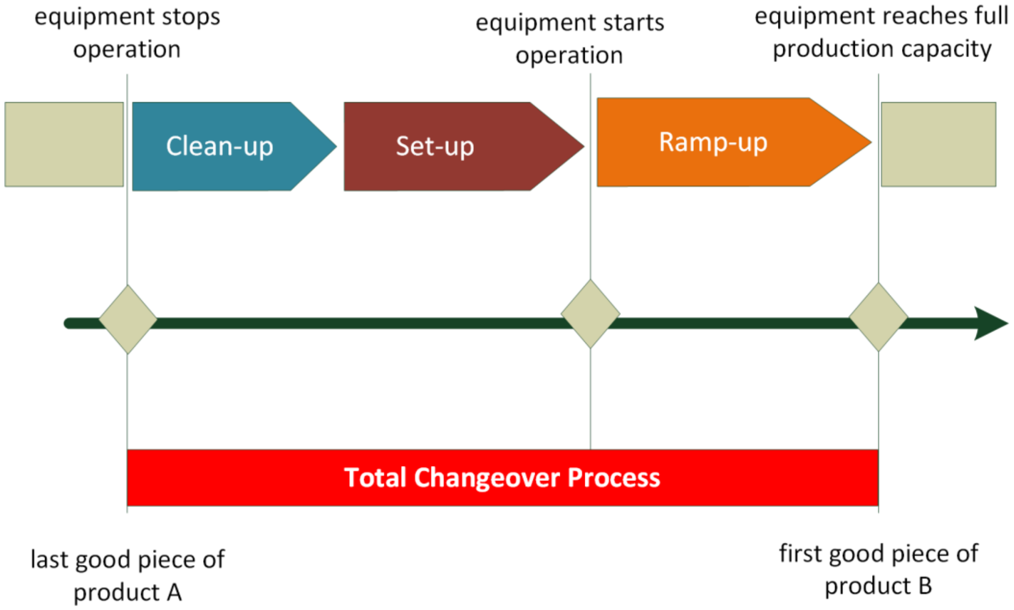


Figure 16 Total changeover process

2.2.5.2.2. Development goal in PERFoRM

In practice, coordination is achieved during set-up by using standard operating procedures (SOP) and checklists. The SOP describes the detailed actions to be performed, how to perform them and when they should be performed. A checklist is an abridged version of the SOP, which contains only a list of the actions without the details of how to perform them. One of the goals in PERFoRM is to create an electronic version of the SOP and checklists so that operators can access them on any computer and other similar devices on the shop floor. The main goal in PERFoRM is to create a formal model for the ramp-up phase.

2.2.5.2.3. Current state

A model for ramp-up in which the process is represented as a finite-state transition process consisting of a sequence of actions and observations has already been developed for new equipment introduction scenario in previous research work. The model is applied to changeover in this solution. Actions are the specific adaptations and adjustments that are applied to move a machine from an initial state to a desired target state. The observations are the observable or measurable conditions of the machine, which may change in response to the applied actions. The state of the machine is the combination of all discrete parameter sets that are used to define the performance of the machine at any point in time.

A self-learning model for decision making during ramp-up, in which the process is formulated as a Markov Decision Process (MDP), is created. This is because the decision on which action to be performed next is usually based on the currently observed state of the system. The model is solved using a reinforcement learning algorithm known as Q-learning. An overview of the learning system is shown in Figure 17.

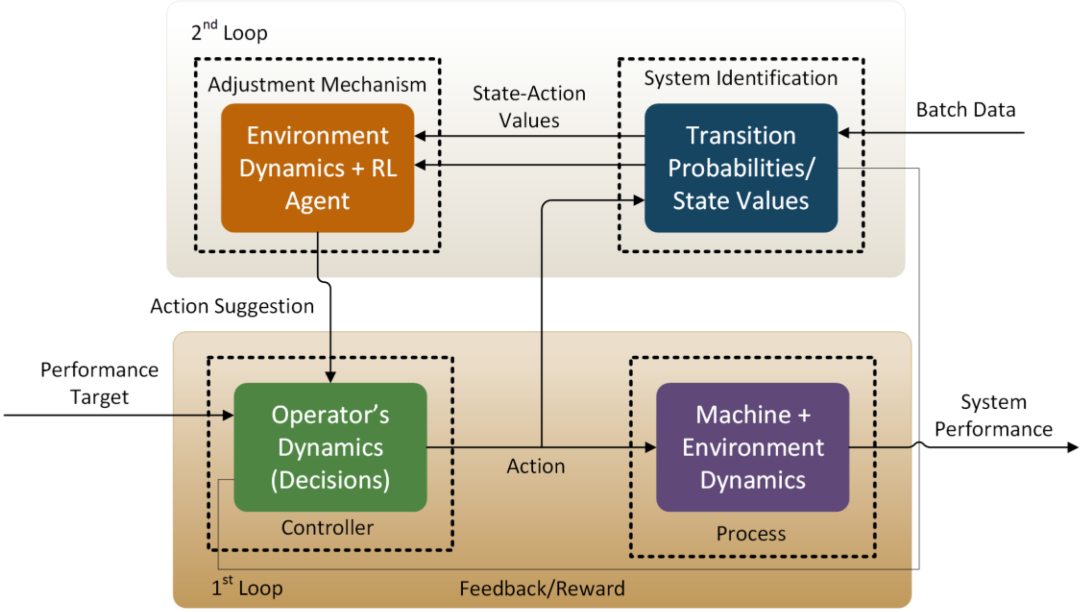


Figure 17 Reinforcement learning during ramp-up

The changeover process modelling also involves creating the static structure and the behavioural diagrams of the system. The static structure diagrams describe the information elements that exist in the system, their internal structure and how the elements relate to one another. The behavioural diagrams describe the dynamic behaviour of the system, such as the sequence of actions that a user can perform and the possible states of the system elements as events occur. The details of the changeover model and the diagrams are documented in D3.3.

2.2.5.2.4. Test / Demonstration results

One of the outputs of this task is an information model for storing the SOP for performing set-up. Instructions on how to setup an equipment provided by the manufacturer as well as the setpoints developed during the testing of the equipment are all stored in the database.

The reinforcement learning model has been evaluated using a simulated automatic production station. The states of the equipment during ramp-up has been discretized and it is assumed that operators will observe only these states. It is also assumed that the operators will only performed actions that have been predefined. A number of known ramp-up cases where generated and are used to drive some unknown ramp-up sessions. The conducted sessions presented different behaviour by accumulating different rewards and followed different paths to achieve the system ramp-up. The results obtained are very preliminary so there are plans to perform actual tests using the Whirlpool use case.

2.2.5.3. Capture of Observations

2.2.5.3.1. Purpose of the solution

Self-adaptation has become an increasingly important capability of many systems especially the ones operating in dynamic manufacturing environments. Such systems must be able to make data-driven decisions and predictions from existing causal relationships through the building of various machine learning models

In product changeover situations, sensory feedback data from machines is often limited because the operations are mainly carried out by human operators and it has been established that lack of knowledge capturing and sharing is a major cause of variable set-up and prolonged ramp-up process. Human expert knowledge about production systems and resources is often tacit and is at best expressed in natural language format. Knowledge about a system that is not represented in a structured data format cannot be fully exploited. Unstructured data is difficult to analyse and current data mining and learning techniques often miss substantial amount of useful information embedded in such data. This restriction has been a problem in real industrial environments particularly during changeover and other related activities such as new equipment introduction and maintenance.

Therefore, the focus of this solution is on the development of an effective method for capturing and interpreting human expert knowledge during product changeover process. This involves documenting actions and observations in a structured and machine interpretable format. The goal is to leverage the data by using it to formulate more effective strategies that will help in reducing total changeover time. As machine learning models are built on sample input data, the captured human expert knowledge is a valuable source of training, testing and exploration data.

2.2.5.3.2. Development goal in PERFoRM

In order to provide the necessary decision support for novice operators during changeover, it is crucial to be able to observe the process, extract and capture the experience of expert operators in a structured way. Therefore, a method for capturing changeover experience is required to enable the learning of the most effective sequence of actions for achieving the required system performance. Fundamentally, a contextual information model needs to be developed to correlate the observed state of a system with adjustment actions and the resulting system behaviour. Although many research papers have identified the need to capture and analyse the decision-making process of experienced operators, none has developed a practical or tangible solution for shop floor use. The main goal of this solution is to create a decision support system for operators' use during changeover on the shop floor.

2.2.5.3.3. Current state and Test / Demonstration results

An interactive application that is intended to be used by shop floor operators during a changeover process was developed. The functions of the application are divided into two:

- *Operational*: The software will be used to manage entities, record events, actions and observations in a contextual and structured format
- *Informational*: The software will also function as a decision support system that will assist operators in their decision-making process by presenting them with sorted, grouped or ranked lists of possible actions that can be performed at every stage of a changeover process.

One of the operational requirements of the software is to provide functionality for creating and managing changeover SOPs. At the beginning of a changeover process, an interface is provided for the operator to select the associated equipment and product. The selections are then used to guide the operator during the process according to the SOP and the ramp-up decision support system. At the beginning of a ramp-up process, the operator observes and records the state of the machine through the software user interface. Necessary action is taken based on the operator's experience and the action is also recorded. The state of the machine after the action is observed and recorded. The process is repeated until the machine reaches the desired goal state as shown in Figure 18.

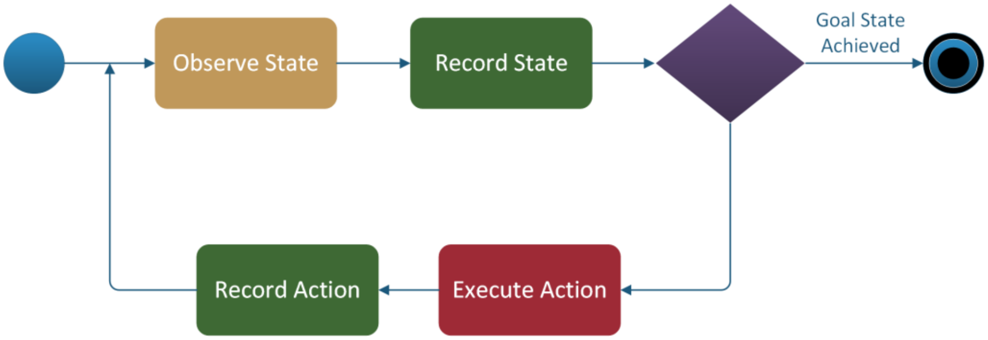


Figure 18 Ramp-up activity diagram

The captured data is converted into knowledge that is used to support future decision making via the reinforcement learning system.

In order to enforce separation of concern and services interoperability, a service-oriented architecture in which the application is composed by integrating self-contained and separately deployed software components is adopted. The backend of the system is a RESTful Web Service created using ASP.NET Core and MongoDB database. The changeover software user interface is a web application, which is powered by the ASP.NET Core backend service. The application is developed using Angular 4.0, TypeScript and Bootstrap tools. An overview of the client application and how it connects to the web service and PERFoRM middleware is shown in Figure 19.

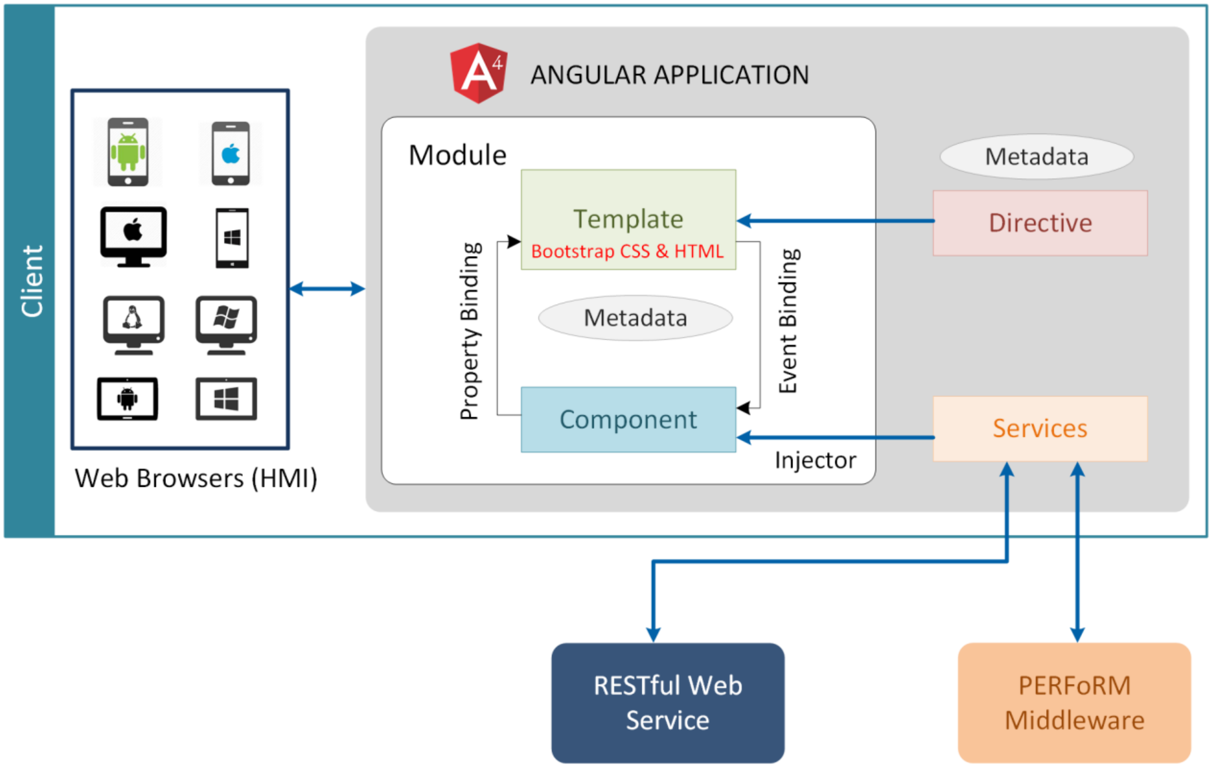


Figure 19 Client side application overview

A web application has been developed for capturing data and also providing decision support for operators during changeover. The functionalities of the software have been tested but it is yet to be used on the shop floor as intended. A study has been conducted in Whirlpool as part of this solution. This involves expert and novice operators' knowledge capturing during the task of reprogramming a collaborative robot to adapt it to new products. One of the suggestions from this study is to provide a tool that can guide the reprogramming activities both in training and on the job so that common errors can be avoided. The plan is to use the developed software to fulfil this need.

2.2.6. Simulation

2.2.6.1. Simulation Environment

2.2.6.1.1. Purpose of the solution

To evaluate the production performance regarding dynamic behaviour of a flexible production system a simulation based decision support is used. Such simulation experiments are modelled within simulation tools to generate a virtual mapping of the real plant with its material flow behaviour for schedule performance evaluation and optimization. To apply such an evaluation in an automated way a generic Simulation Environment delivers the simulation tool wrapper with all scenario defining and evaluating functionalities for use case specification, experiment configuration and integrated execution of a simulation model using simulation engines of commercial simulation tools within the industrial IT environment of a production plant.

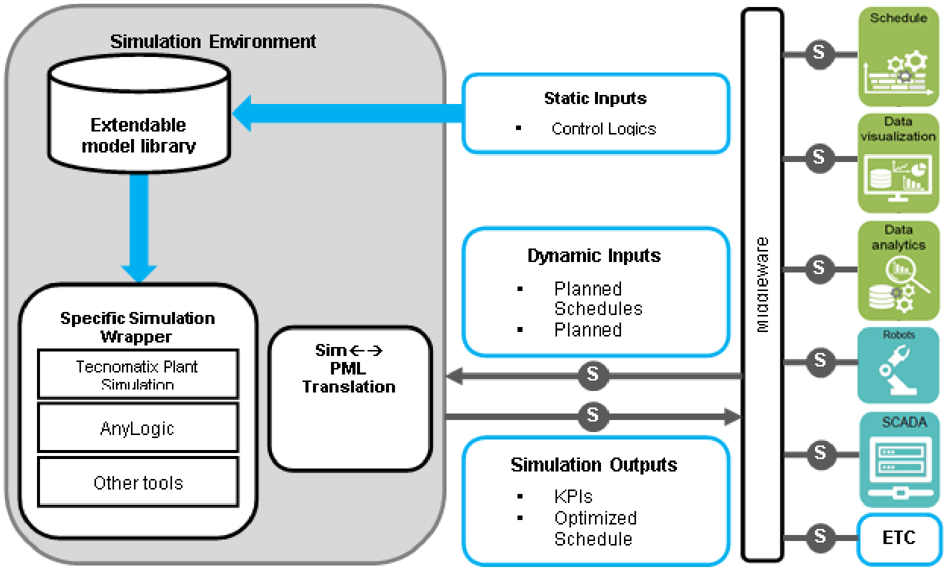


Figure 20 Simulation Environment Informational View

2.2.6.1.2. Development goal in PERFoRM

Goal in PERFoRM project is the implementation of the developed wrapper concepts and its sub modules into a Simulation Environment to subsequently demonstrate and prototype the results of the concepts. At this point it is stressed that the developed simulation techniques are built on existing simulation software, PlantSimulation and AnyLogic.

2.2.6.1.3. Current state

Prototypical implementation for Siemens use case including model generation of a PlantSimulation model based on current production schedule and current production process states and evaluating lead time. Focus lies in assessment of maintenance schedules efficiently integrated into the production schedule suggested form schedule module. The maintenance tasks result from data analytics module for predictive maintenance. Results from performance evaluation by simulation are visualized in KPI visualization module. Information exchange is done by PERFoRM ML data format and interface definition.

A so called "Milan Demonstrator" shows the automatic generation and execution of a Tecnomatix Plant Simulation model based on xml format defined in PERFoRMML data model. This shows the flexible description of machine conditions and routing schedules based on actual system state and field data or based on scenarios to be evaluated. All wrapper functionalities regarding

- preparation - read xml file as service request -,
- model generation – instantiate network model automatically -,
- simulation execution – run simulation to evaluate defined key performance indicators –
- and result availability – write results in xml file format –

are shown with this application. Four different interfaces are defined:

- getting data from the middleware into the simulation environment,
- sending data from the simulation environment to the Middleware,
- getting data from control planning logic into the simulation environment,
- as well as the internal interface within the simulation environment for execution of the simulation model itself (actual simulation tools are Plant Simulation and AnyLogic) (already shown in Milan Demonstrator!)

Next phase is to test the interfaces with PERFoRM middleware. This includes the proper generation of xml file regarding simulation request by other modules like scheduler or KPI visualization, as well as field data access for state estimation and schedule planning.

2.2.6.2. Excel-based Simulation

2.2.6.2.1. Purpose of the solution

Knowing the changeable nature of manufacturing systems that is affecting the current production environment, this solution aims at providing a new approach to model a typical production system based on a modular paradigm. In particular, the objective of this solution is to propose a novel approach to create a production model leading to speed-up the simulation of the dynamic behaviour of a flexible and reconfigurable production system. In fact, the current simulation activities adopted for production process analysis are usually based on predefined models [4]. The problem arises when the process to be analysed is not exactly replicable with one of such models. This case demonstrates the need to use elementary, modular and interoperable entities allowing to compose a typical production process in order to model any process, from the more common to the more complex one. Therefore, this concept leads to readapt the approach to be adopted before simulation activity changing the question of 'which model is most appropriate for replicating a specific production logic?' to 'what combination of standard components is the most appropriate?'.

2.2.6.2.2. Development goal in PERFoRM

Following these considerations, this solution provides an approach for the design of discrete event simulation (DES) models of large reconfigurable and flexible manufacturing systems based on a limited set of finite state automata (FSA) implementation for modular manufacturing entities and relationships.

The proposed approach implies a Discrete Event Simulation (DES) as it is based on mathematical/logic model able to point out state changes at precise points in simulated time [5]. Moreover, the proposed approach refers, also, to Agent-based (AB) simulation as its specific instance (Finite State Automata) are, in accordance with Wooldridge and Jennings, autonomous, proactive, reactive and "social" as they can communicate or interact with other systems [6]. Furthermore, this approach simplifies the process of creating new manufacturing scenarios where it can be frequently modified without any depth knowledge of simulation language by essential shop floor system parameters and logics and simplified description of relative interconnections, highlighting its capability to be applied at continuous evolving and quickly reconfigurable production systems.

2.2.6.2.3. Current state

The specific approach aiming at providing a complete modelling and simulation process of a production system can be represented according to the schema depicted in Figure 21.

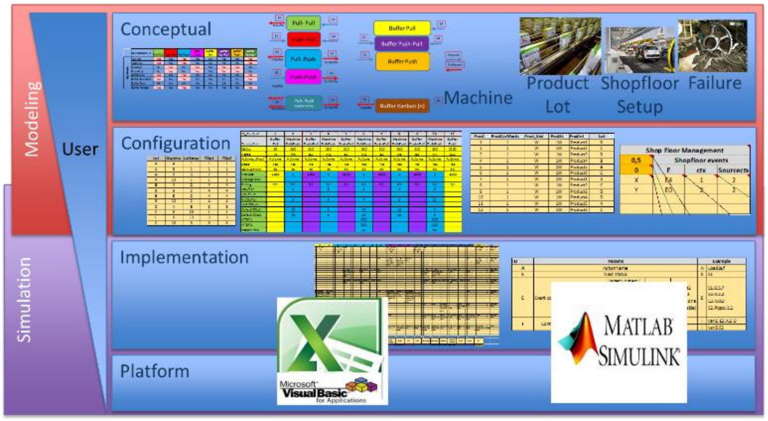


Figure 21: Reference Model

2.2.6.2.4. Test / Demonstration results

In particular, from this figure it is possible to identify the different phases in which the tool could be used.

The *Conceptual Phase* can be guaranteed by the possibility to model any (discrete) production flow as a composition of different production entities (machines, buffer, but also operators and applications). They can be described with Push/Pull logic, based on the interaction these entities have with each other, which means the way they exchange signals to coordinate material flows, and the basic working parameters as duration of operations and setups for different product types.

In this context, it is possible to state that generic production entities can be represented by 4 different machines and as many types of buffers that can be classified (see Figure 2) as:

1. Push-Push Machine
2. Pull-Pull Machine
3. Push-Pull Machine

1. Pull Buffer
2. Push Buffer
3. Pull-Push Buffer
4. Kanban Buffer

4. Pull-Push Machine

When the Push-Push Machine finishes the part, it pushes it to the downstream station or buffer and starts working on a new part only when it is pushed from its upstream machine WSi-1. If no part is available there, the machine is idle and the production is on hold. Instead, when Pull-Pull receives a request from downstream workstation WSi+1, it pulls the request to upstream workstation WSi-1 for providing a component. Once it finishes the product it releases it to the downstream requestor and waits for another request.

The third machine receives and works a part pushed by the upstream workstation WSi-1, but it releases that part only when it receives the request from downstream workstation WSi+1. Instead, the forth has the opposite behaviour: it requires a part as soon as it is available and it pushes the part as soon as that part is finished to the downstream entities.

Regarding to the buffers classification, the first one describes a raw parts buffer that provide material when it receives a request and for this reason is called "Pull"; it is the first entity of a production process. Vice versa, the Push Buffer implements a finished goods warehouse or a buffer with extracted items, it collects parts each time a Push-x workstation terminates production. The last two buffers can be used to model the Interoperational buffers. The difference is that Pull-Push Buffer is a passive intermediate entity receiving components and providing parts while Kanban buffer is an active instance able to autonomously pull items from upstream machine, in order to keep a specific inventory level. This component is call Kanban buffer to underline its role in a pull system based on upwards propagation of the demand. All four buffers are characterized by a given maximum capacity. In Figure 21 arrows specify direction of the operation trigger

Following these definitions, it is possible to model any (discrete) production flow as a composition of the different entities already described, allowing to compose a typical production process in order to model any process, from the more common to the more complex and, therefore, allowing to perform the Configuration phase

The third level refers to the *Implementation Phase* of components utilizing the Finite States Automata approach. In fact, as described in [7] a generalized sequential logic system that can be described by a number of output (n,o) which depends on the present and the past values of the input (n,i) can be formalized as a Finite State Machine (FSM). Therefore, each production standard entities, introduced above, can be considered as a mathematical abstraction that describes all the states representing each possible situation in which these entities may ever be and all inputs and outputs with defined events. Events are also utilized to synchronized internal operations like end working or fault/repair.

The lowest level deals with the actual implementation of the *Platform*. Currently 2 prototypes are implemented: one on MS-Excel/VBA and the second on MATLAB-Simulink [8], showing its full adaptability to any production context.

2.2.6.3. Agent based simulation

As described in the motivation of the PERFoRM project the manufacturing industry is subject of increasing market dynamics and growing competition. This results into the need for more flexible and reconfigurable manufacturing systems. Leading to the need for new design and control techniques to drive these more flexible and reconfigurable manufacturing systems. To foster this migration process, planning tools (here simulation) have to be enabled to model this new demands.

As an approach that is not yet adopted by industry agent-based simulation was successfully deployed for flexible and complex manufacturing systems in [9]. First studies have shown that the agent-based paradigm is a promising substitute for the classically used discrete-event simulation for (flexible) manufacturing systems [11]. However, the respective simulation models are generally built in simulation environments and as simulation software for specialists as well as without a specific manufacturing context. An implementation in an industrial grade software tool, referring to a simulation software tool that is mainly designed for manufacturing purposes and commonly used in industry, has not been introduced yet.

Thus, the in the project developed solution aims to lower the barriers of the industry to benefit from the agent-based simulation paradigm. Figure 22 illustrates the situation of generally available simulation tools, their capability to simulate in the both mentioned simulation paradigms and the “industrial grade” status of available tools leading to the gap.

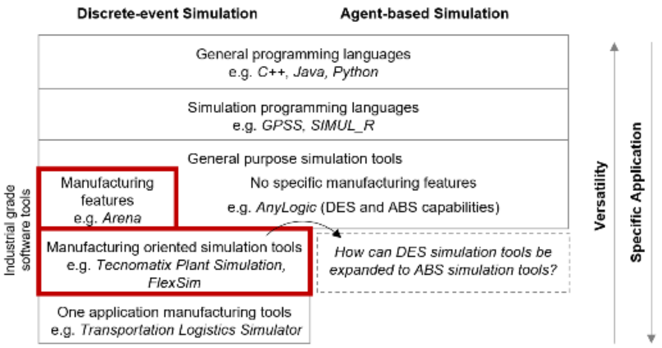


Figure 22: Classification of simulation software and research question [Büth et al., 2017]

2.2.6.3.1. Development goal in PERFoRM

As industrial grade simulation tools are mostly not capable to simulate agent-based systems: The goal within the PERFoRM project is to develop a general approach is to simulate manufacturing systems utilizing the agent-based simulation paradigm within in industrial grade discrete-event simulation tools. Furthermore, the implementation approach is tested in a case study and then used for the GKN use case to simulate the “Micro Flow Cell(s)” within a manufacturing system. Lastly, the approach is evaluated.

2.2.6.3.2. *Current state*

The implementation approach was developed, matching DES tool elements/functions with ABS requirements. See Figure 23 for an outline of the general implementation approach. A verification was successfully done, see subchapter below. The use case application is to be done.

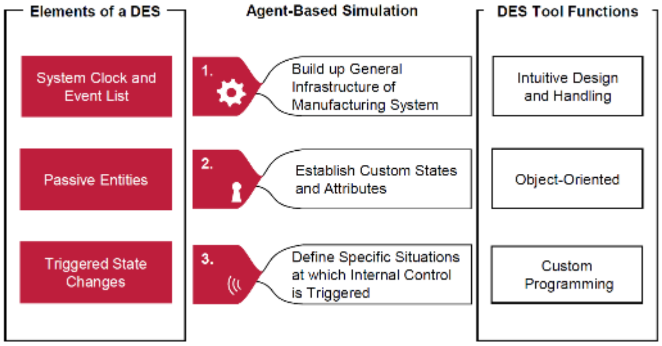


Figure 23: Three general steps for implementing an agent-based logic into a discrete-event simulation tool [Büth et al., 2017]

2.2.6.3.3. *Test / Demonstration results*

The implementation approach was verified by comparing results of an existing flexible manufacturing system simulation modelled within the multi-purpose simulation tool *Anylogic* (ABS capable) to the results of the same model within the industrial grade software tool *Tecnomatix PlantSimulation*. The results were positive with small deviations, which can be reasoned differently. For details, see the publication [13].

2.2.7. Planning logic

2.2.7.1. Multiobjective scheduling of production orders and maintenance task (bio inspired)

2.2.7.1.1. *Purpose of the solution*

This tool intends to define a new scheduling approach, combining the production tasks with the maintenance tasks that need to be performed.

This task intends to allocate the production and maintenance tasks for the designated machines which are usually defined in two different documents/platforms. Nowadays the maintenance tasks are managed separately from the production management which constitutes a problem, since most of the time the maintenance tasks allocation do not take into count the production optimization. Hence, the tool must combine the two needs (Production and Maintenance) in order to daily deliver a schedule for the designated 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.

2.2.7.1.2. Development goal in PERFoRM

The development within perform is presented in Figure 24. The idea is use the PERFoRM middleware to link all the needed sources of information. In this sense, all the structure and relevant information from the production system will be store and consulted on PERFoRM ML. Hence, the tool can know all the stations and stages available on the system and that need to be managed and all the products' descriptions, in order to know all the tasks that need to be performed and the other.

The scheduling tool will receive the production planning from the ERP and the maintenance problems from a Human Machine Interface, through the middleware, that will be developed for the test case and that allows the operators to introduce maintenance issues.

Whenever a new scheduling is required, the tool will generate several possibilities (e.g. solve all the maintenance problems as soon as possible, solve all the maintenance problems as late as possible, scheduling performing all the maintenance tasks but optimizing the production, etc).

This list of possible scheduling's will be sent to the simulation environment, being developed to be used on this demonstration also, and the simulation tool will simulate all the possible scheduling's and decide which scheduling must be applied and sent to the operators and machines.

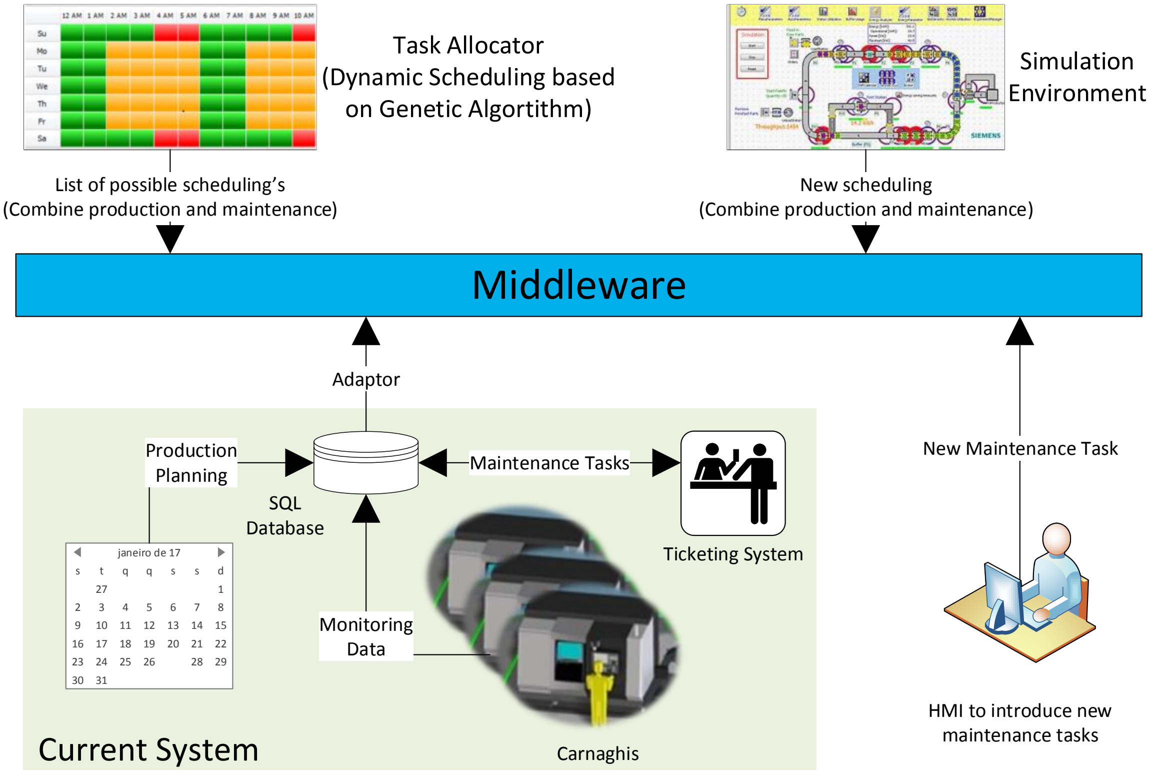


Figure 24 - Multiobjective scheduling of production orders and maintenance tasks for the Siemens use case

2.2.7.1.3. Current state

Currently the implementation is being finalized and optimized in order to put correct parameters for the optimization algorithm. The algorithm is being developed according to the specification in Figure 25.

Presently the main focuses encompass the parametrization and specification of “Calculate the fitness of individuals of new population” step, before the “Termination criteria?” question. This step is particularly important for the solution because in the proposed solution a list of possible solutions is delivered and in this step all the possible solutions are evaluated according to the objective and ranked according to that evaluation. After the evaluation the algorithm knows if it should start a new iteration or keep the already existent solutions, if the current solutions were good enough.

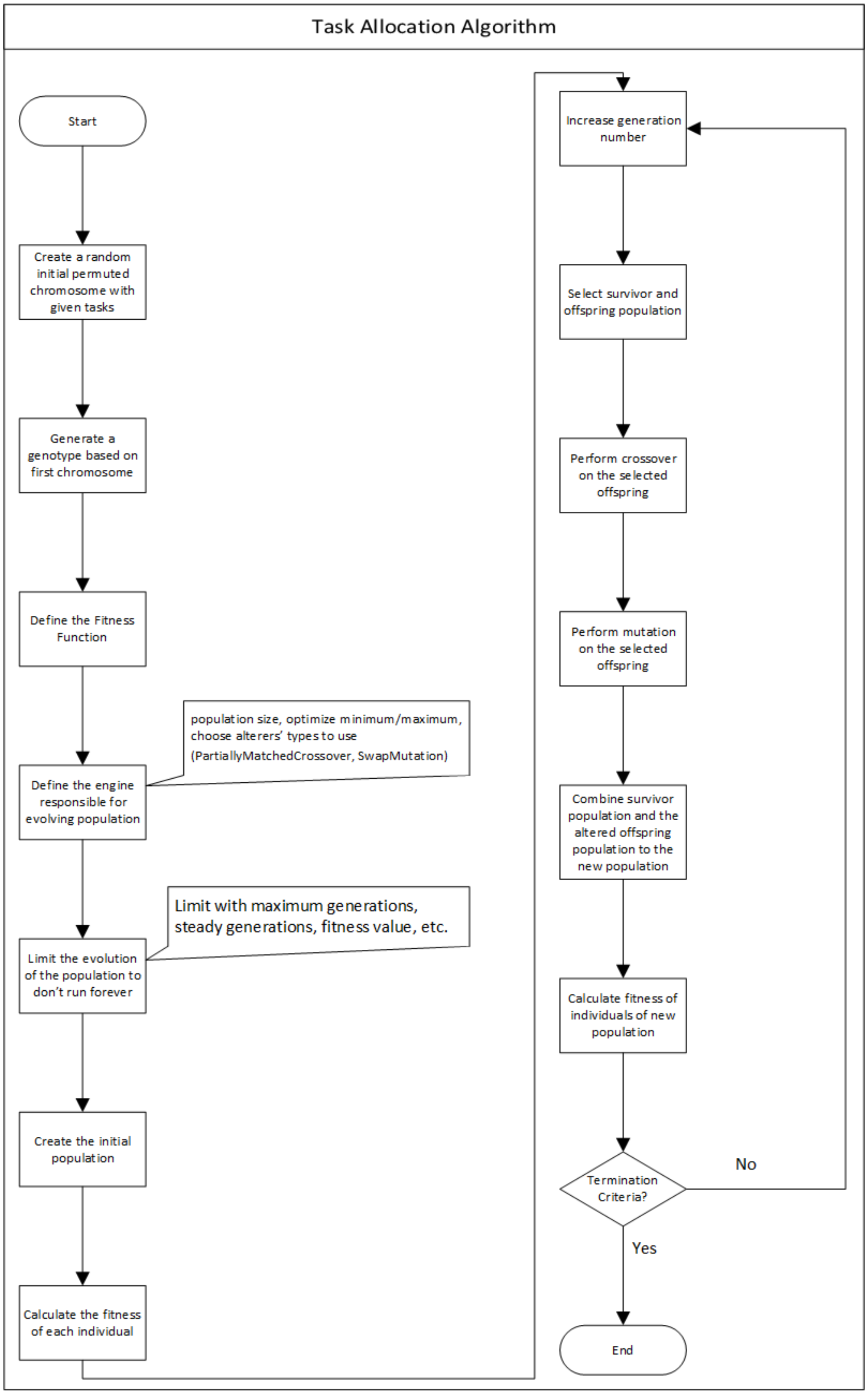


Figure 25: Task Allocation Algorithm

2.2.7.2. Multi-objective planning & scheduling for dynamic, flexible manufacturing system

2.2.7.2.1. Purpose of the solution

Traditional centralised production planning and scheduling systems can provide optimal solutions with respect to certain performance measures such as makespan, tardiness and energy consumption. However, they are inflexible and are unable to cope with real-world environments that are characterised by complexities, dynamic alterations and unanticipated changes in the production conditions. Multi-agent-based scheduling technology provides a promising way to address dynamic changes and unpredictable disturbances in the manufacturing shop floor without disruption in production. Although agent-based technology has been proven to be an appropriate solution for handling real-time and on-demand changes, optimal production schedules cannot be guaranteed. The focus of this solution is on the harmonization and standardization of techniques for achieving both flexible and optimal or near-optimal planning and scheduling.

2.2.7.2.2. Development goal in PERFoRM

The goal in PERFoRM is to combine metaheuristics and agents-based technology in addressing multi-objective job-shop scheduling problems. The chosen performance measures are makespan and energy consumption. The aim is to generate Pareto optimal schedules based on ideal production circumstances using metaheuristics such as genetic algorithm. A schedule that satisfies the subjective preference of a human decision maker is then selected. Instead of using a centralised control system to execute the schedule, multi-agent systems are deployed. Decision-making policies that follow the selected schedule are built into the agents and they will attempt to execute the preliminary policies as much as possible. The agents will change their decision rules dynamically in response to any contingencies that could lead to deviations from the initial production schedule (Figure 26).

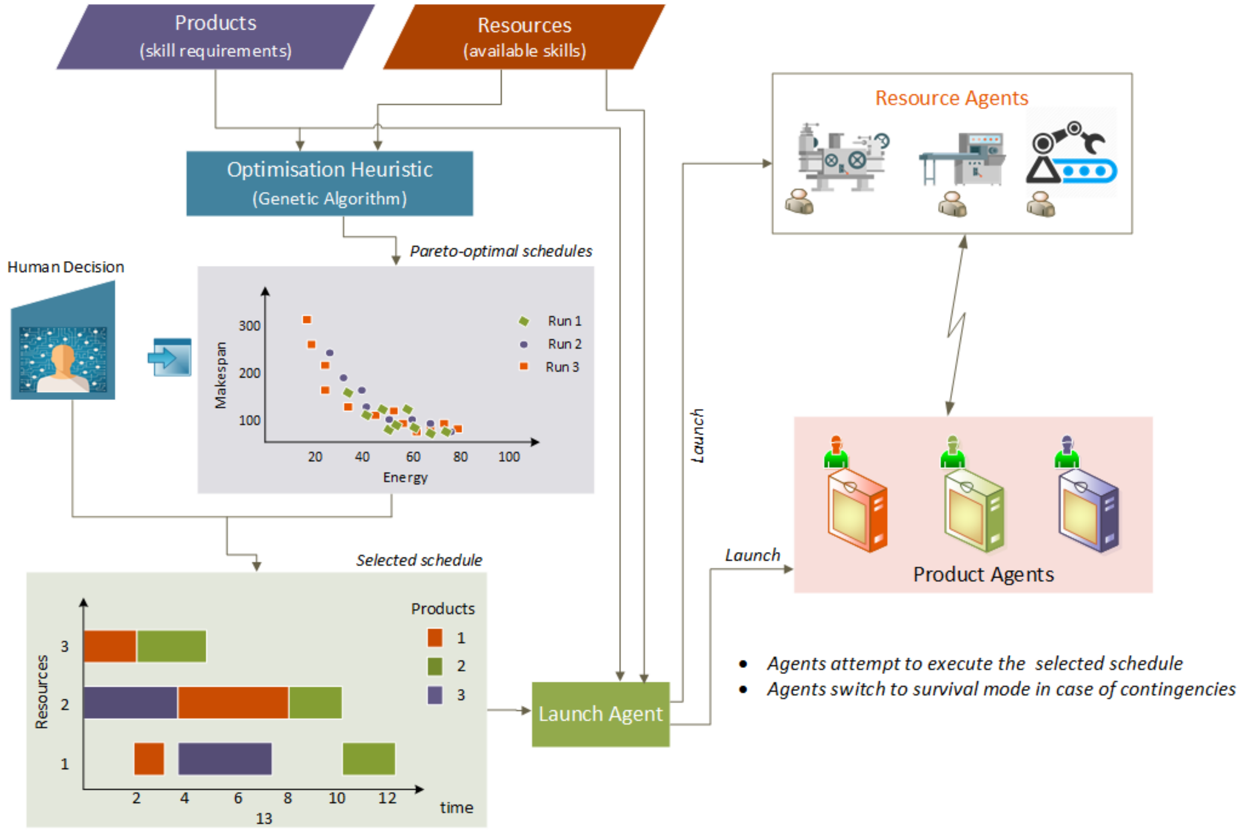


Figure 26 **Hybrid multi-objective optimization and agent-based scheduling**

In the event of machine breakdown or any other unforeseen events, the agents react quickly by allocating jobs to other available resources without disruptions in production. The agents cooperate to achieve both their individual and system-wide goals. This hybrid solution will attempt to produce optimal schedules with respect to the chosen performance measures and also respond to emerging system behaviors that cannot be precisely predicted in advance.

2.2.7.2.3. Current state

There are two aspects to this solution. The first is the design of the production scheduling system. Job-shop scheduling is one of the common combinatorial optimization problem and it can take a significant amount of computing power if there are many jobs and machines. Therefore, genetic algorithm (metaheuristic) is used to solve this problem. Although there are no guarantee that stochastic and heuristic search methods will produce optimal schedules, they are known to give satisfactory results since they do not need to evaluate all the feasible search space to extract good solutions.

A simulator for generating scheduling plans has been created using NSGA-II, which is suited to multiple objective optimization problem. The chosen performance measures to be minimized are makespan and total energy consumption. Different lot splitting plans and dispatching decisions deliver different scheduling plans. Each scheduling plan has its own performance on total processing energy consumption and makespan. The agent-based simulator for executing the selected schedule is currently under development. The simulator is being developed using JAVA Agent Development Framework (JADE), which is an open source platform for developing peer-to-peer agent based application.

2.2.7.2.4. Test / Demonstration results

Initial simulation tests have shown that the scheduling plan which reduces energy consumption does not necessarily reduce makespan. The complexity of searching for optimal scheduling plans which minimise the total energy consumption and makespan increases with the increasing numbers of jobs and machines. When the resource allocation rules are pre-defined, it is crucial to identify the optimal lot splitting plans and the orders for all sub-lots to be released to the workshop, thereby finding the optimal scheduling plans. Once the agent-based simulator is completed, full execution tests will be performed based on the GKN use case. A 48-Node Raspberry Pi Cluster has just been built to be used as a testbed for running the agents in a distributed computing environment Figure 27.

Figure 27 48-Node Raspberry Pi cluster

2.2.7.3. Reconfiguration management in flexible manufacturing system

2.2.7.3.1. *Purpose of the solution*

A cost-effective response to sudden market changes requires reconfigurable machines that can be changed rapidly (hardware and software components) and integrated in an open-architecture dynamic environment. The purpose of this solution is to develop suitable reconfigurability mechanism and framework for plug and produce machines and robots.

2.2.7.3.2. *Development goal in PERFoRM*

The goal in PERFoRM is to use the agents-based planning and scheduling tools developed in T4.2 to develop new optimisation approaches for improving the reconfigurability of a seamless production system. The results of this task will be used in a number of work packages for optimisation purposes.

2.2.7.3.3. *Current state*

A tool based on genetic optimisation algorithm has been developed for solving the reconfiguration management problem in a dynamic and flexible manufacturing system. Since the reconfiguration task has a strong link with the scheduling task in T4.2, there is the need to develop an agent-based solution. The whole solution is currently being redefined as promised in the original proposal.

2.2.7.4. *Agent Based Reconfiguration Tool*

2.2.7.4.1. *Purpose of the solution*

This solutions aims to provide a scalable and reliable logical reconfiguration of reconfigurable production cells. Therefore, the tool detects the physical changes at cell organization and represents it at a logical level enabling, e.g., a dynamic adjustment of the robot programs and, potentially, notifying also other tools in need for the cell current organization.

2.2.7.4.2. *Development goal in PERFoRM*

In PERFoRM, the goal is the full development of the solution based in agents developed using the JADE framework. Additionally, and besides being able to detect the current system organization and organization related events, the developed agents also collect process-related information, providing them, in a loose-couple manner, to a KPI monitoring tool which, in turn, allows the users to dynamically be-aware of the system current status.

2.2.7.4.3. *Current state*

In its current state, the multi-agent system (MAS) is composed by two agent types, namely the “Robot Agent” and the “Process Agent”, that are instantiated accordingly with the number of cells, defining the number of robot agents, and the physical processes, defining the number of process agents. Presently, the MAS is able to logically reconfigure the system based on the physical organization and dynamics as also to provide a real-time collection of data, using an OPC-UA approach, providing the data in a MQTT format.

2.2.7.4.4. Test / Demonstration results

Extensive tests have been conducted at Polytechnic Institute of Bragança (IPB), both with a physical and a simulated system. The first, allowed validating the MAS approach in a non-real, but physical, system. Additionally, these tests also validated the connection of the agents to the OPC-UA as also the data publication using MQTT. Finally, the MAS was also validated in a simulated environment, enabling the system validation in a multiple situations possibility (note that this also opens the possibility for scalability tests and allow the users to validate future approaches in a simulated environment (which is broadly used in industrial situations).

2.2.7.5. Energy based planning with rescheduling

2.2.7.5.1. Purpose of the solution

A possible way to increase energy efficiency in machines can be in reducing the machine fixed power demand or complete shutdown by acting on the auxiliary components according to particular machine states. The states can be provided by i.e. SERCOS Energy and Siemens PROFIenergy. These profiles provide an interface for components to communicate information about energy consumption values in different energetic states. Machines can also be set to certain status such as reducing the power consumption in e.g. reducing the power consumption or shutting off.

XETICS LEAN has a scheduling module "Planning Board" for determining operating times for stations. The planning board supports the production planner by optimizing the worker and machine allocation to orders and production steps. It allows to optimize machine utilization, throughput and adherence to delivery dates. As a result the planned operating times of a machine can be determined and provided as a formatted Schedule.

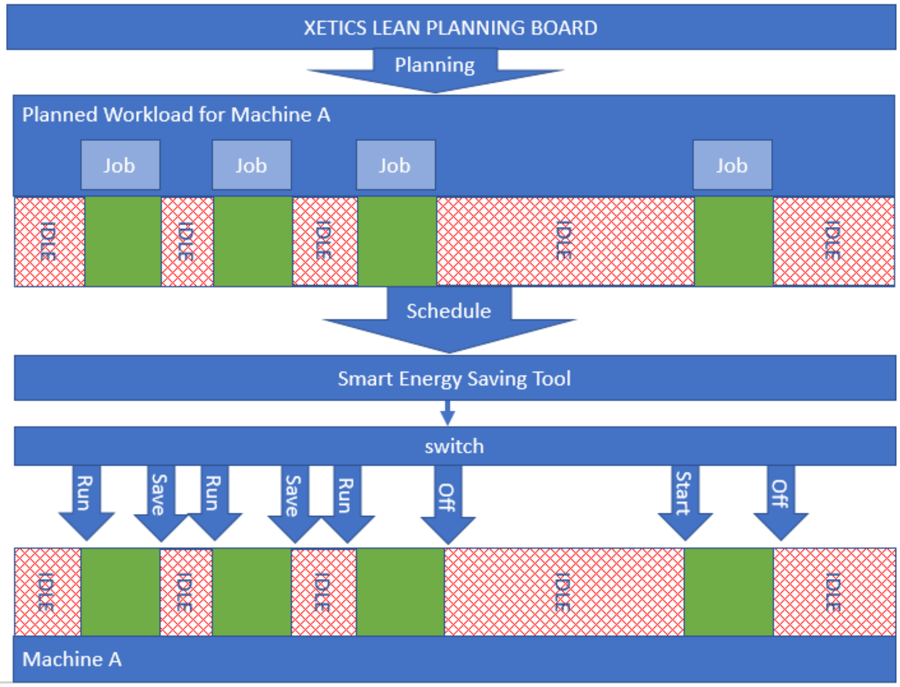


Figure 28: Trigger Equipment based on Schedule

The Schedule of the individual machines needs then to be transferred to a Smart Energy Saving Tool to let them activate the saving state by deciding if a reduced power consumption or a complete shutdown will be suitable.

The following states are to be indicated:

- Reduced power: Reduce the energy consumption if the machine is not currently required in the process. Just in time, the component is then reactivated and makes its functionality available again.
- Complete shutdown: Shutdown the machine and restart for the next upcoming job.

XETICS ELAN has a scheduling module "Planning Board" for determining operating times for stations. The planning board supports the production planner by optimizing the worker and machine allocation to orders and production steps. It allows to optimize machine utilization, throughput and adherence to delivery dates. The planned operating times of the individual machines are transferred to the Smart Energy Saving Tool

On the basis of the information stored there, the stations are placed in energy-saving states. At the start of the planned operation, these states are reset. The planned operating times of the individual machines are transferred to the Smart Energy Saving Tool.

The following states are to be indicated:

- Supply operation (not all required): Especially in flexible production, a machine provides more functionality than is required for each individual production step. For example, in the case of large plants, machine components will only be produced at a later stage in the production process or not at all required. Or to reduce their energy consumption if they are not currently required in the process. Just in time, the component is then reactivated and makes its functionality available again.

- Part-load operation: If no full power is required, e.g. by field weakening in drives, a reduced energy consumption can be achieved.
- Measurement and visualization: Measurement and visualization are combined with data collection and data processing. In addition to electrical variables, other information, such as coolant and compressed air consumption, can also be relevant for cost optimization. With its equipment integration, XETICS LEAN provides mechanisms to capture and visualize the current consumption

The integration of machines and equipment is the basis for fully automated production: MES data must span multiple production facilities when the product is manufactured. The objective is to make machine data available plant-wide and to give XETICS LEAN controlling access to machines and equipment.

- Online visualization of the machine states
- Error acquisition and error statistics
- Machine state statistics

Begin Date 10/13/16 07:00:0
End Date 10/13/16 08:00:0
☑ Distance1 (mm)
☑ Distance2 (mm)
☑ Pressure (bar)
● Step

Figure 29 Visualization of Sensordata

2.2.7.5.2. Development goal in PERFoRM

Functionalities of the dispatcher algorithm to be fulfilled for Planning Board Version

- Orders should be planned according to priority given in the planned order list
    - The next process step of an order should be planned at the earliest possible starting time, but without affecting the planned process times of higher priorized orders (that means a process step is just done before higher priorized order-steps if the higher priorized is not affected in processing time)
- For the planning of the next process step and the concrete time assignment, there has to be an idle working station with the relevant process capability and an worker with the relevant process qualification

- When an assignment of a worker and station is done to a process step, worker and station should be marked as occupied for the planned processing time (except the working station has a capacity bigger than one. Then, just one capacity has to be occupied)
- Consideration of optimization windows for each working station
- Consideration of resource availability of material availability (by date) and "Fertigungshilfsmittel" (mobile tools needed for processing)
- Consideration of worker availability only at the beginning and the end of the process step
- consideration of ramp up times
- manual moving and fixing of individual process steps

2.2.7.5.3. Current state

Opposing to scheduling, the planning is currently done manually. The planning board supports the production planner in terms of:

- dispatching all production steps on the available resource at the earliest time based on an ordered list
- allowing the production planner easily to change the order sequence
- marking orders as 'rocket-lot' to put into effect, that this order has to be done as fast as possible
- highlighting orders that are not completed on schedule
- offering means to change the resource assignments while assuring only legal changes are made
- synchronizing with the latest progress on the equipment
- planning direction - forwards (push) or backwards (pull)

User interface:

1. The planning panel showing resource allocation over time.

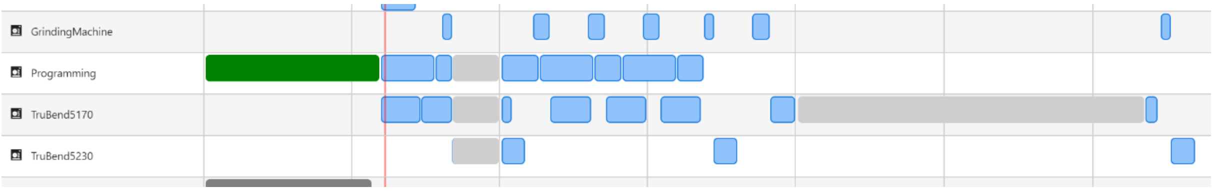


Figure 30: Planning Board

2. The planning list containing all orders that are planned and visible in the planning panel. The order of this list is the base for dispatching the steps on the machines.
3. An overview list containing all unplanned orders. Filters and sorting helping the planner to focus on the relevant entries

2.2.7.5.4. Test / Demonstration results

No test or demonstration results have been conducted until now.

2.2.8. Decision support

2.2.8.1. Value Stream Model in Excel

2.2.8.1.1. Purpose of the solution

The proposed solution has the objective to describe a specific model able to provide a responsive new manufacturing approach based on a holistic view of the factory and self-adaptive production capability enabling the ability to rapidly adjust the manufacturing system capacity to market demands and to quickly integrate new functions and process technologies into existing systems [14].

In particular, it is based on the identification of correlation between static factors that can have an influence on industrial plant and its dynamic parameters. The former are considered as those factors that are dependent on the factory business scenario such as, for example market demands, market variability or plant availability etc. and for this reason, they are called as Key Business Factors (KBFs). The latter considers the production parameters used to evaluate the production trend and the process behavior and for this reason, they are identified as Key Performance Indicator (KPIs).

2.2.8.1.2. Development goal in PERFoRM

This solution proposes a specific production system model able to point out a straightly relationship between KBFs and KPIs, leading to develop an innovative manufacturing system based on a new agile concept introducing the implementation of methods, methodologies and strategies for transforming existing production systems into plug-and-produce production ones based on Cyber-Physical Systems technologies, according to PERFoRM’s objectives [15].

In fact, by providing the relationship between static and dynamic parameters, the model can be exploited to replicate the factory performance, to conduct static simulations and to apply the sensitivity analysis needed to support the decision process.

These aspects can be facilitated from two points of view. From one hand, knowing the KBFs already selected, the model produces a static snapshot of the situation in the system at a specific time, highlighting the eventual problem of saturation, utilization rate and bottleneck as a potential element to be solved in order to have a full production improvement. On the other hand, a simulation activity based on what if analysis can be applied in order to evaluate the system performance. In fact, changing different KBFs, multiple scenarios can be evaluated aiming at choosing the best combination of KBFs able to guarantee the most advantageous performances. Therefore, it provides decision support systems based method, exploitable by different levels of decision maker: from process owner, through plant manager or logistic manager towards head of department or foreman.

In this way, it is possible to show this model can guarantee some capabilities enabled by Cyber Physical Systems. In fact, the identification of KBFs and KPIs relationship, the implementation of predictive model able to foresee how production process takes place given economic context and the ability to strategically manage the process variables and thus to control its operations, provide the integration of three different capabilities of CPS [16] as described in PERFoRM’s WP1:

- measurement and sensing capability enabling to capture what information can be sensed to measure the state of the factory;
- modeling and simulation enabling how well this information is used to model and simulate factory activities;

- optimization and control enabling how best to use it to continually optimize manufacturing production.

2.2.8.1.3. *Current state*

The structure of the model can be split in two different levels as depicted in the Figure 1 in order to correctly manage the inputs to be inserted and the consequent outputs.

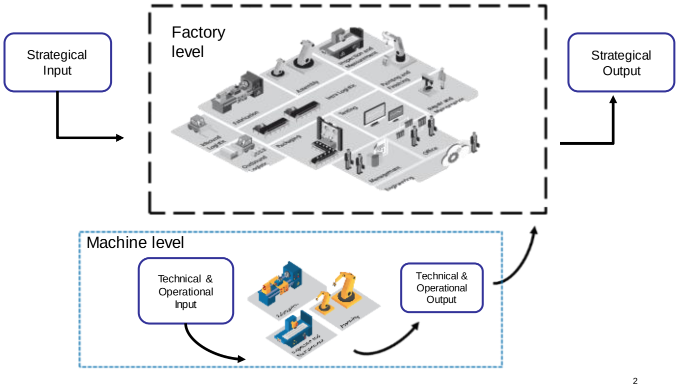


Figure 31 Production System Model: the structure

The ‘Factory Level’ represents overall production plant as it aims at depicting the impact of strategical decision, which can be taken for whole factory. The lower level represents the “Machine level” where different equations implemented to describe the mutual relationship among different workstations allow the overall production behavior description. For this reason, it needs to receive more specific, operational and more technical details. Obviously, the Machine level results lead to obtain the Factory level outputs. This implies different methodological phases needed to reach the desired results. They are listed below:

- Factory input definition: overall input that has been decided as a strategical choice by decision maker before to start production process are defined.
- Machine input definition: all data needed to describe the process of single workstation are specified. These data concern to the technical parameters, which characterize the theoretical behavior and the operational information which describe the effective behaviour of specific workstations when they are involved within the production context.
- Machine output calculation: the process parameters of different machine are collected and they are aggregated in the specific Key Performance Indicators (KPIs). Here the resulting parameters coming from two different inputs layers are elaborated and reformed into outputs.
- Factory Output identification: the results describing the overall Factory behaviour are obtained.

2.2.8.1.4. *Test / Demonstration results*

The industrial use case has been implemented and it concerns to home appliance production.

The production process is composed of a production line made by different machines and buffers and it is represented in Figure 32. In particular, the line ends with decoupling buffer that aims at provide different semi-finished products to four parallel lines.

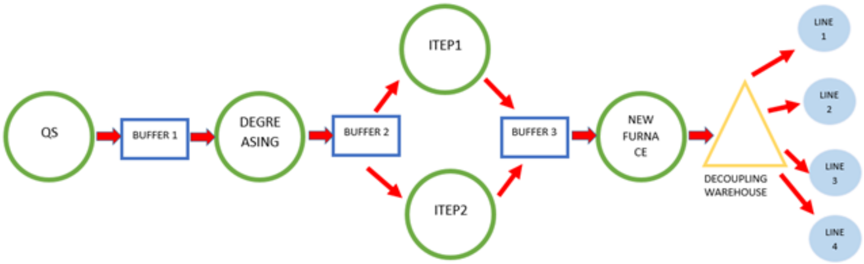


Figure 32 Use Case: production line

As these lines are identical in terms of process and in terms of market demand to be satisfied, it has been decided to model a unique total demands impacting directly on the decoupling buffer. For this reason, the overall Machine Level of this industrial use case has been modeled as shown in Figure 33.

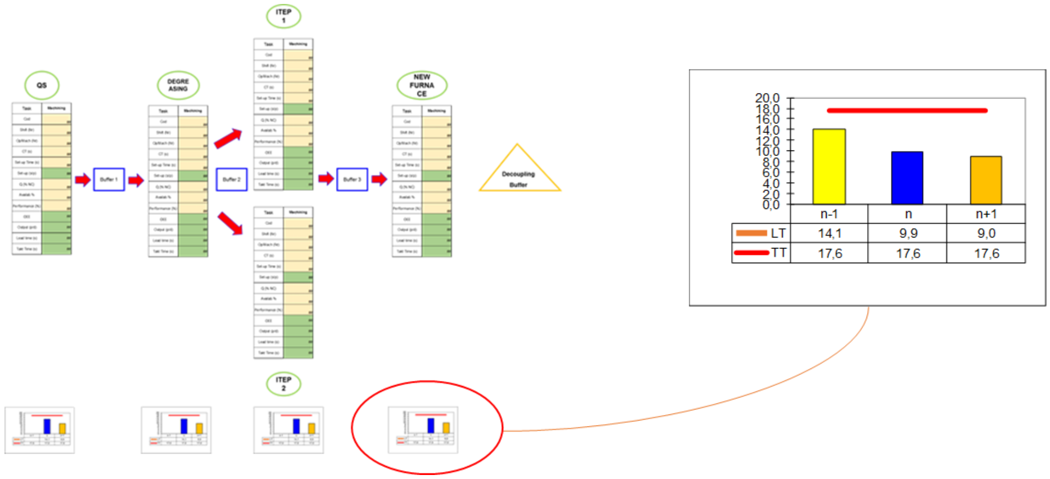


Figure 33 Use Case: Machine Level model

Figure 33 illustrates the relationship with consecutive machines in terms of Throughput, Lead-time and Takt Time. In particular, considering the Takt Time (TT) as the maximum threshold for product realization (red line in Figure 33, on the right) and knowing the specific Lead-time of upstream and downstream workstation, it is possible to manage the different KBFs in order to maximize the production process balance and the workstation utilization rate. Furthermore, the analysis represented in Figure 33 (on the right), identifies the bottleneck that determines the overall production pace, suggesting which is the workstation needs to be instantaneously improved.

The second result is depicted in Figure 34 where the Factory Output are compared (Warehouse Cost, Set-up Cost, Cost of Production Loss; exemplary values). The figure points out that the Cost of Production Loss (blue line) has the highest impact on the overall cost (yellow line). In particular, Warehouse Costs (orange line) have to be considered according to the right scale (which scores from 0 to 70000€). On the other hand, the other considered costs refer to the left scale (which scores from 150000€ to 2100000€). From these evidences, it can be state that the most convenient strategy for this use case is to maintain a big amount of items within decoupling buffer rather than to continuously adjust the batch size needed to meet market demand.

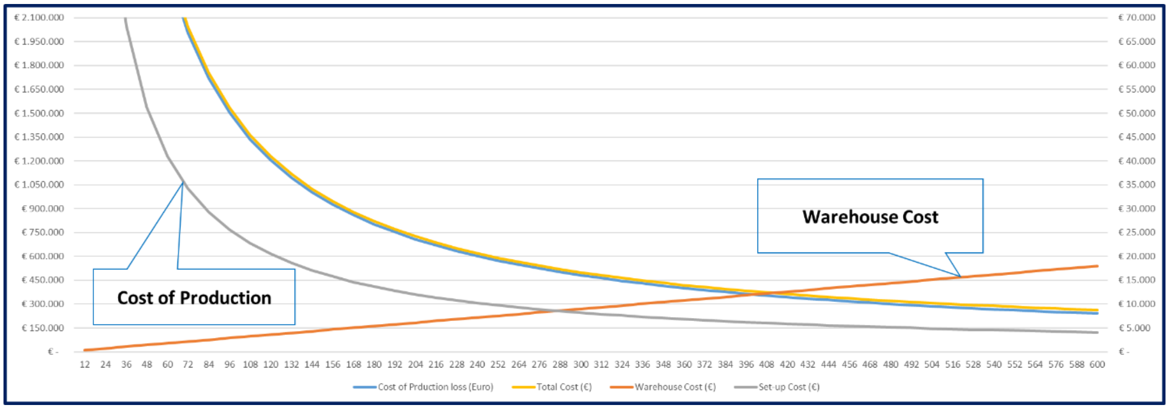


Figure 34: Overall Cost representation

2.2.8.2. KPI monitoring with what-if-game functionality

2.2.8.2.1. Purpose of the solution

The KPI monitoring with what-if-game functionality tool is a Web-based solution to support decision-making strategies, by monitoring Key Performance Indicators (KPIs), detecting trends and deviations, and performing what-if game based on the variation of Key Business Factors (KBFs) and generating the associated KPI implication.

2.2.8.2.2. Development goal in PERFoRM

The goal is the development of a tool for monitoring and visualization of existing production environments and corresponding KPIs to improve overall production efficiency. The solution is also used to perform what-if game based on the variation of Key Business Factors (KBFs) and generating the associated KPI implication (see Deliverable 4.3).

2.2.8.2.3. Current state

The development of the tool for the Whirlpool use case is concluded, being necessary the possible adaptations during the deployment phase.

2.2.8.2.4. Test / Demonstration results

The connectivity tests involving the tool, a middleware solution and a prototype SQL DB adapter were performed successfully, using dummy data, accordingly with the tests defined in deliverable 6.2. The calculation of KPIs based on the received KBFs was tested for correctness, also successfully. The what-if functionality is implemented and tested (in-house). The tool may require minor adjustments at deployment time on the industrial environment.

2.2.8.3. Min-Max Data Mining Toolbox

2.2.8.3.1. Purpose of the solution

The purpose of this research work is to develop a condition based predictive maintenance system in accordance with the overall PERFoRM line of thinking. The research work aims on the detection of abnormal behavior of production equipment. The proposed approach should detect defects and unexpected behavior down to component level (e.g. electrical drives, pumps or mechanical parts) of whole machine tools. All components of e.g. a machine tool are connected indirectly or directly to a central electricity supply. The combination of measuring the central electricity connection of a machine with a high data acquisition rate and perform an analysis of the resulting power signature can give important evidence to the actual machines component conditions.

2.2.8.3.1. Development goal in PERFoRM

The development goal consists of two different components. On the one hand there is the development and implementation of the data analytic framework. According to Figure 35 the design of the data analytic framework can be divided into two main steps. The first stage includes the automatic detection of different components of machine tools. This feature is realized by manual segmentation and labelling and the extraction of features within the time domain. In the second “Automatic and real time” stage, the extracted feature vector is used to make a live prediction of the condition of selected components within the machine tool. This is done by use of a signal comparison module. Deviation of selected features is tracked and the deviation is used to define the actual components condition.

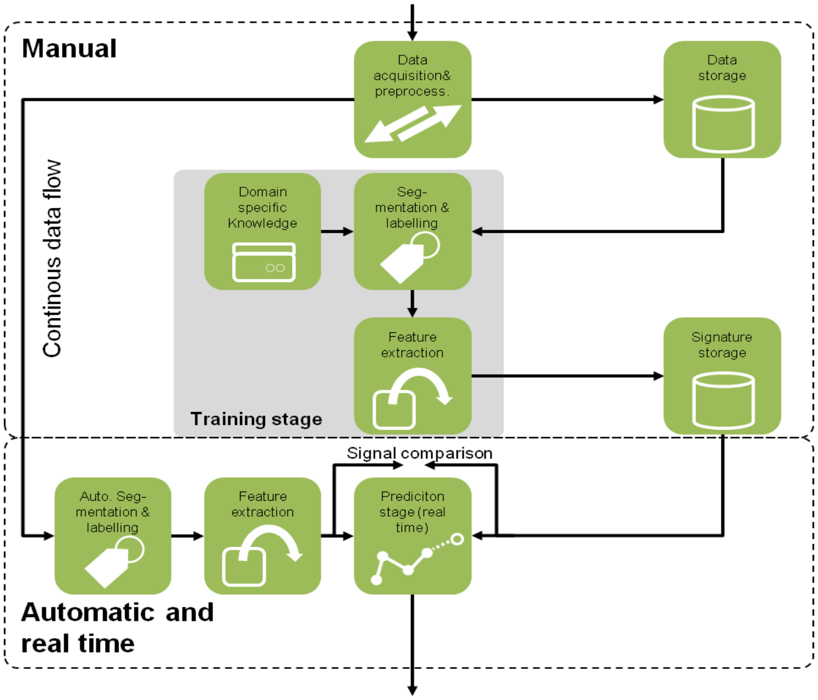


Figure 35 Design of the data analytic framework

On the other hand, the development and implementation of a hardware interface for communication, metering of electric current and as the embedded system for the data analytic framework. The hardware implementation is depicted in Figure 36.

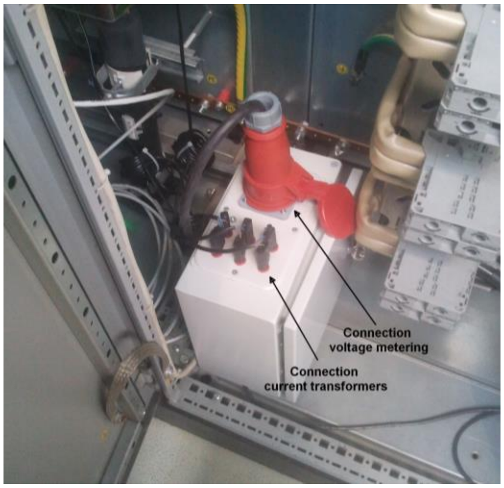


Figure 36 Hardware implementation of the "Min-Max data mining" solution

2.2.8.3.2. Current state

The hardware implementation is completed. Currently two boxes according Figure 36 are installed at the Siemens Turbocompressor Plant in Duisburg. Additional one training box for the mapping and allocation of machine tool components is installed. The development of the data mining framework according to Figure 35 is finished. Implementation of the intended solution is under construction.

2.2.8.3.3. Test / Demonstration results

First results show that an automatic segmentation and labelling of the acquired data is feasible. The detection of component defects down to the level of smallest functional units (e.g. bearings) is feasible with the suggested approach.

2.2.8.4. Data Mining

2.2.8.4.1. Purpose of the solution

Ensuring the availability and reliability of the equipment is one of the biggest challenges in the manufacturing industry. Equipment faults can cause long production line stoppages, high maintenance costs and low product quality. The industry is capturing machine condition data, maintenance reports and other data from different parts of the process chain. Currently, a significant portion of this data is not being optimally analysed or used for scheduling optimal maintenance task to avoid unexpected downtime of equipment. Typically data are captured during the manufacturing process by various solutions for specific purposes. These data are generally not suitable for performing data analytics for the purpose of predictive maintenance. Consequently a tool is required to merge data from multiple sources, pre-process the data and visualise for proper decision making. As part of the Task 4.3, a framework for data mining for predictive maintenance has been proposed (see Figure 37). A tool for pre-processing and visualisation of the data has been developed. The results from this tool will support the machine maintenance team to schedule appropriate maintenance well in advance to avoid any unexpected downtime. The Data Mining framework consists of a Backend system, offline data mining, visualisation and also web services to allow access to the functionalities provided by the backend system. The Backend system uses the alarms, maintenance data and the machine maintenance manual. The tool merges the data, pre-process the data and makes it available to the visualisation tool. The results generated by the *Data Mining* tool will support the creation of new maintenance tasks within the *Maintenance Task Editor*. The *Scheduling* tool will then access the maintenance tasks and propose schedules for production and maintenance tasks. After evaluation of the schedules are done by the *Simulation* tool, the most appropriate maintenance task will be transferred to the *SAP* system.

The Data Mining tool has incorporated a service-oriented architecture to support plug & produce and ease of integration with the other systems developed in WP2, WP3 and WP4.

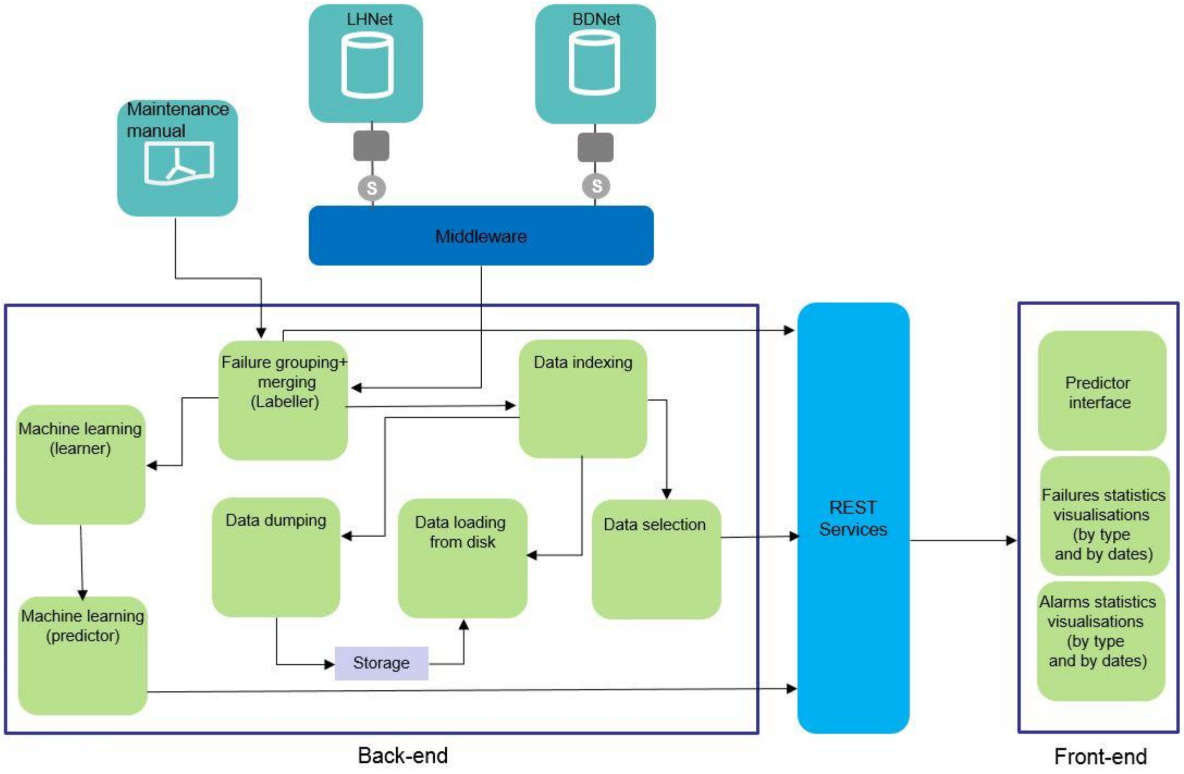


Figure 37: Architecture of the system data mining tool

2.2.8.4.2. Development goal in PERFoRM

The goal is to develop a fully service-oriented data mining tool that is able to collect data from the databases (through the middleware and right adapter), map and aggregate the failure and alarms data, pre-preprocess the data, group the failure records using information acquired from the machine maintenance manual, generate alarm patterns (using machine learning) for given type failure. The output of the data mining tool should be available to a visualisation tool as well as other solutions been developed in WP2, WP3, and WP4.

2.2.8.4.3. Current state

The prototype of most of the components have been developed and tested with offline CSV file (replicas of the data from the LHNet and the BDNet database). Work is currently being conducted to integrate the back-end system with other visualisation tool developed by project partners. Work is also being conducted to test the integrated solution involving the input of raw data to the Backend system directly from the databases via the PERFoRM middleware. Currently, the contents of the databases (LHNet and DBNet, Oracle and SQL servers respectively) are being replicated on local MySQL databases.

2.2.8.4.4. Test / Demonstration results

REST service for requesting preprocessed data for visualisation:

REST Services component wraps the modules developed within the Backend system into exported services as REST APIs. Thus allowing an interaction between the external modules and the Backend system. The results of the Backend system are serialised into JSON objects to be used visualisation tools. An illustration of an input and output for a REST API (getFailure) for getting a list of failures can be seen below:

- Inputs (in a text format):
    - machine_name,
    - start_date and
    - end_date
- Returns a list of failures recorded for the specific machine during the specified time frame. The format used for the list is JSON.

- Example of input service request:
    - http://localhost:8080/PERFoRMServices2/perfservices/perfservices/failures?begin=10/10/2015&end=10/10/2017&machine=Carnaghi AC16
- Example of the output:

```
[ {   "damageType": "mechanisch",   "eventStatus": "Erledigt",   "faultMessage": "Captohalter im Einsatz fallengelassen.Maschine zeigt nicht an,das kein Werkzeug drin ist.",   "invent": "Carnaghi AC16",   "inventNumber": <removed>,   "label": "9 ZUBEHÖRTEILE UND WERZEUGHALTER",   "notificationDate": 1447372800000,   "orderNumber": "<removed>",   "regiNumber": "<removed>",   "reporter": "<removed>",   "spAt": 1447372800000,   "standstill": true  },

{   "damageType": "mechanisch",   "eventStatus": "Erledigt",   "faultMessage": "Späneförder defekt",   "invent": "Carnaghi AC16",   "inventNumber": <removed>,   "label": "13 SPÄNEFÖRDERER",   "notificationDate": 1446422400000,   "orderNumber": "<removed>",   "regiNumber": "<removed>",   "reporter": "<removed>",   "spAt": 1446422400000,   "standstill": true  },…]
```

Visualisation tool:

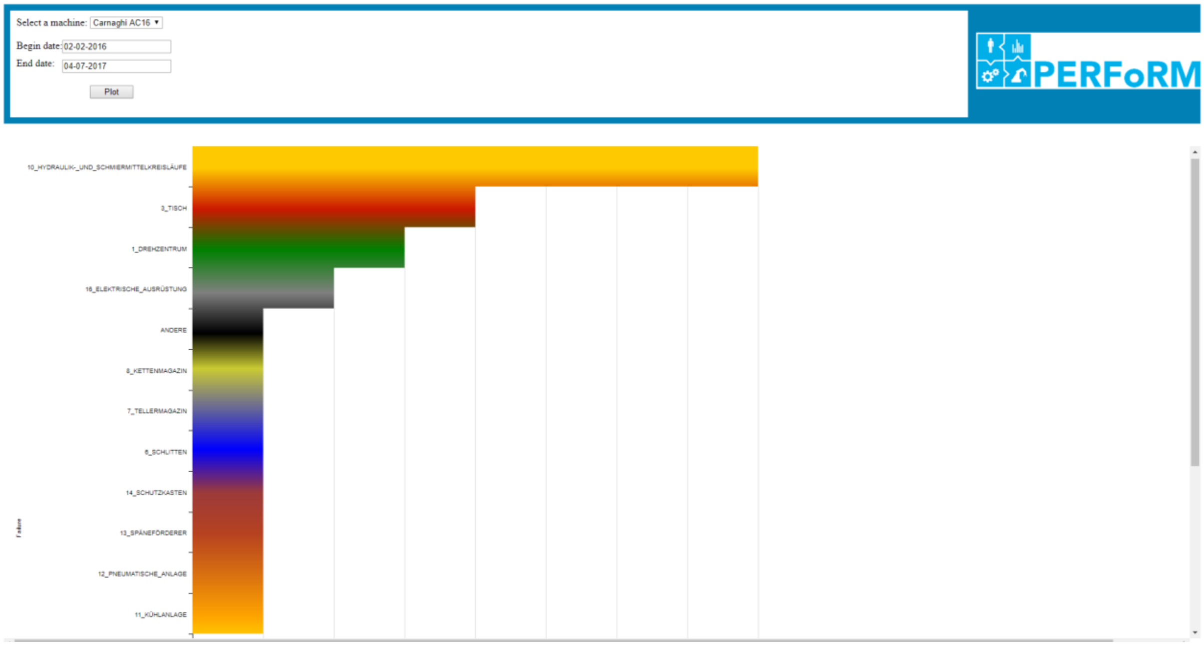
A web based visualisation tool has been developed to analyse and visualise the data analysed by the Backend system. The visualisation tool will use the REST API of the Backend system to request and get pre-processed data for visualisation.

Figure 38 shows an illustration of visualisation of the trends of alarms (for a set of key alarms) against the observed failures during a time frame.

Select an alarm: 700040, 510315, 14014, 700041
Select a machine: Carnaghi AC16
Begin date: 08-11-2016
End date: 05-07-2017

Figure 38: Visualisation of alarm trends

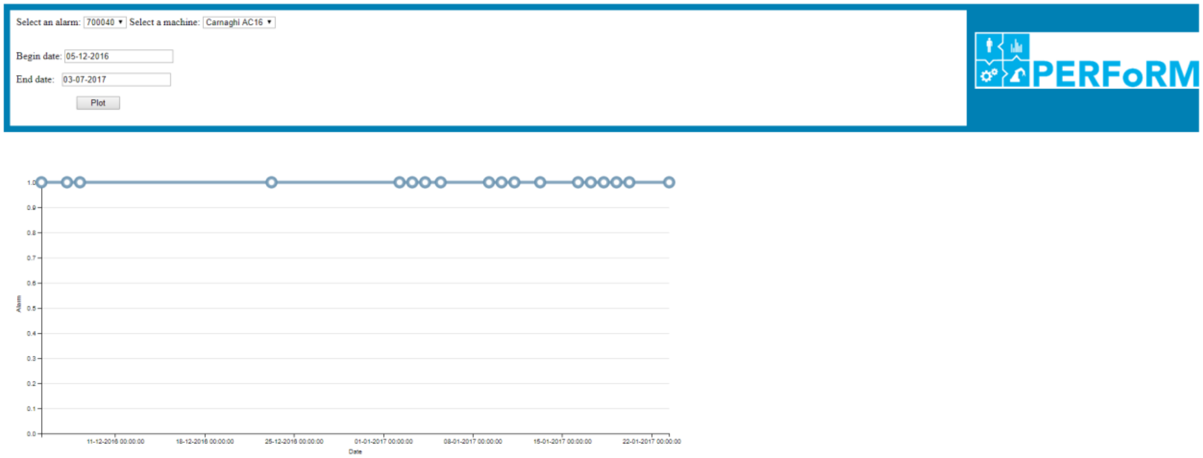
Figure 39 shows the percentage of each type of failures for a given machine for a given time frame.



Select a machine: Carnaghi AC16
Begin date: 02-02-2016
End date: 04-07-2017

Figure 39: Failures statistics

The Figure 40 visualizes a single alarm for a given machine in a time frame.



Select an alarm: 700040 ▾ Select a machine: Carnaghi AC16 ▾

Begin date: 05-12-2016

End date: 03-07-2017

Figure 40: Single alarm visualisation

Offline Datamining for generating the prediction rules:

Decision tree technique was applied on the data grouped by different failure categories and also on the complete dataset (without grouping them by types of failures). Further work needs to be conducted to validate these rules and potentially increase the accuracy of these rules. Table 1 illustrates a few rules extracted via the decision tree technique.

Table 1: Alarms pattern (rules) extracted from the decision tree model created using full ungrouped data

Rules / Alarm patterns
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-510308 = Yes AND Alarm-700540 = No THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-700543 = No AND Alarm-10208 = Yes AND Alarm-67051 = No AND Alarm-700754 = No: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-700543 = No AND Alarm-601012 = Yes AND Alarm-700636 = No: THEN failure = Yes
Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-700543 = No AND Alarm-700310 = Yes AND Alarm-700733 = Yes AND Alarm-16913 = Yes AND Alarm-67051 = No: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-10621 = No AND Alarm-700310 = No AND Alarm-4075 = No AND Alarm-510216 = No AND Alarm-700239 = No AND Alarm-600609 = Yes AND Alarm-700635 = Yes AND Alarm-600908 = Yes AND Alarm-600410 = Yes AND Alarm-16906 = Yes AND Alarm-67050 = No: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-10621 = No AND Alarm-700310 = Yes: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-510216 = No AND Alarm-700239 = No AND Alarm-600609 = Yes AND Alarm-700635 = Yes AND Alarm-600908 = Yes AND Alarm-600410 = Yes AND Alarm-700646 = Yes AND Alarm-510309 = Yes AND Alarm-10208 = Yes AND Alarm-6406 = No AND Alarm-700732 = Yes AND Alarm-601012 = Yes: THEN failure = Yes

2.2.8.5. Automatic Monitoring and visualization of KPIs

2.2.8.5.1. Purpose of the solution

Transparency of production system status and functionality through automated production system monitoring and visualization of real-time KPIs

2.2.8.5.2. Development goal in PERFoRM

Calculation of KPIs are implemented as a service which stores all calculated kpi gathering from events like movein/moveout/sensor/statechanges and storing in own table/db.

2.2.8.5.3. Current state

To provide the designer of the KPIs with a maximum of felxibility, XETICS LEAN allows to define the calculation based on system variables. The usable predefined variables are:

> Period, scrappedParts, goodParts, standByTime, productiveTime, maintenanceTime, rampUpTime, repairTime, nonScheduledTime, undefinedTime

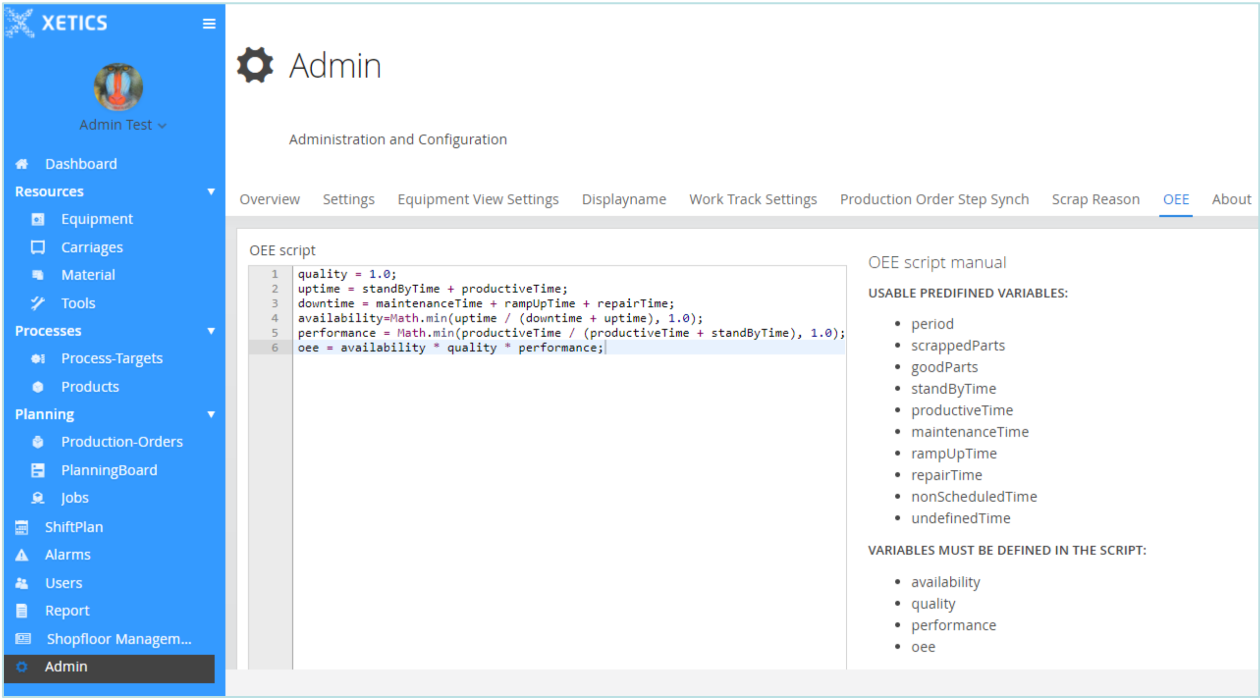
These variables are the base for calculation the following KPIs

KPI	Formula
Availability	standByTime / period;
Quality	scrappedParts / (goodParts + scrappedParts);
Performance	productiveTime / (standByTime + productiveTime);
OEE	availability * quality * performance * 100;

The formulas can be dynamically changed at any time.

2.2.8.5.4. *Test / Demonstration results*

A special field ‘OEE script’ contains editable code that calculates the KPI individually.



OEE script

```
quality = 1.0;
uptime = standByTime + productiveTime;
downtime = maintenanceTime + rampUpTime + repairTime;
availability=Math.min(uptime / (downtime + uptime), 1.0);
performance = Math.min(productiveTime / (productiveTime + standByTime), 1.0);
oee = availability * quality * performance;
```

OEE script manual

USABLE PREDIFINED VARIABLES:

- period
- scrappedParts
- goodParts
- standByTime
- productiveTime
- maintenanceTime
- rampUpTime
- repairTime
- nonScheduledTime
- undefinedTime

VARIABLES MUST BE DEFINED IN THE SCRIPT:

- availability
- quality
- performance
- oee

Figure 41 Settings in Admin/OEE Tab

The Result is transformes in thes graphical icons. They are used in context of the complete shopfloor or in context of a selected equipment (machine)

Figure 42 KPIs for an Equipment (OEE also for the shopfloor)

2.2.8.6. *Bayesian Diagnostics & Prognostics for Manufacturing Equipment*

2.2.8.6.1. *Purpose of the solution*

The aim of data-driven fault diagnosis is to identify the status of the machine and isolate the faulty element (subsystem) in case of fault or failure. In other words, the data-driven fault diagnosis can be performed in two steps, firstly; by extract the machine status, secondly; identify the faulty part in fault/failure case.

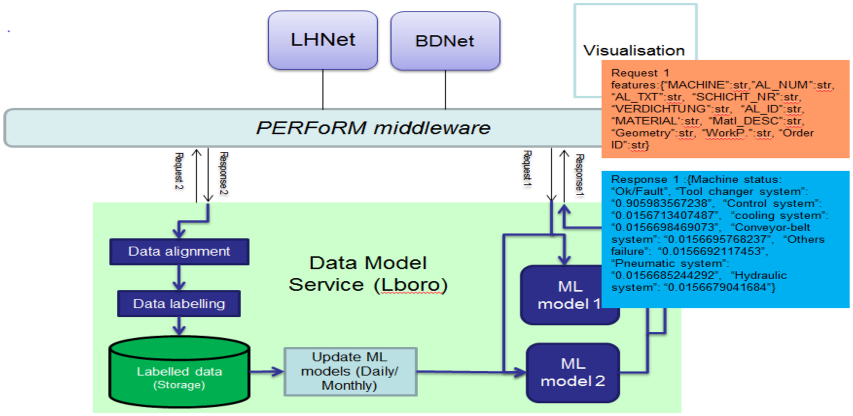
The basic idea is to identify the status of the machine and isolate the faulty element in failure case directly from data available about the machine (such as sensors, maintenance data, etc.), instead of the process model. Initially, a purely data-driven approach was introduced. Then due to performance limitation on the data-driven approaches, hybrid approaches that combine experts' knowledge with probabilistic machine learning approaches were introduced.

2.2.8.6.2. *Development goal in PERFoRM*

The main objective of data-driven fault diagnosis is to support maintenance decisions in manufacturing applications, through continues monitoring and visualization of machine status. The machinery data can be sensory data or based on human observation on the shop-floor. Hence the users will be able to make beneficial decisions regarding KPIs, to react properly to changes or disruptions in the machines/production lines.

2.2.8.6.3. *Current state*

In the previous stage, data-driven models were tested with a different set of features, and features that guarantee the most accurate performance were selected. Currently, the proposed solution packaged as a web server that can enquire data through PERFoRM middleware and update the data driven-models on regular basis (e.g. daily/weekly). Also, the data analytic service can be called from the visualization service and respond with the predicted machine status and the predicted root cause of the failure. Figure 43 illustrates the architecture of the proposed data-analytic services and how it interacts with the PERFoRM middleware and the visualization service.



```
Request 1
features:{"MACHINE":str,"AL_NUM":str,
"AL_TXT":str, "SCHICHT_NR":str,
"VERDICHTUNG":str, "AL_ID":str,
"MATERIAL':str, "Matl_DESC":str,
"Geometry":str, "WorkP.":str, "Order
ID":str}
```

```
Response 1 :{Machine status:
"Ok/Fault", "Tool changer system":
"0.905983567238", "Control system":
"0.0156713407487", "cooling system":
"0.0156698469073", "Conveyor-belt
system": "0.0156695768237", "Others
failure": "0.0156692117453",
"Pneumatic system":
"0.0156685244292", "Hydraulic
system": "0.0156679041684"}
```

Figure 43: Data-analytic Service with PML and visualisation

The data-driven based models were improved by including the alarms from the machine with the operational data. However, it is believed that data-driven diagnostic system can be improved by including human knowledge using the Bayesian network and Failure Mode and Effect Analysis (FMEA); as a structured representation of human-knowledge.

FMEA is a systematic approach that determines potential failure modes in a complex system, e.g. production line in a factory, caused by either design or manufacturing process defects. Moreover, it identifies process significant characteristics that are required to prevent or detect failure mode. In general, FMEA is a tool used to understand the mechanics of failure in order to prevent pit from occurring.

In the Siemens case study, the FMEA study was conducted by interviewing the operators and maintenance engineering, in order to identify the probable failures and their negative impacts on the overall process. In addition, the interdependence between different subsystems was extracted from the experts.

Based on the FMEA we noticed that the nature of the desired process (drilling and milling) and the workpiece properties (size, material) are directly affecting the failure modes. Moreover, crucial failures highlighted by experts rarely appeared in the maintenance data, which indicates the importance of histogram analysis and frequency features in the data-driven models. In short, experts can identify and deal quickly with frequent problems.

The process characters extracted from the FMEA and the data-driven approaches are the core of the Bayesian models, which use these characters as input while the outputs are the maintenance information with faulty systems probabilities. The implementation side of the Bayesian network will be continued in WP7.

2.2.8.6.4. Test / Demonstration results

The proposed solution was tested on Siemens case-study. In which, maintenance data, operational data, and alarms log datasets for three Carnaghi machines (AC16, AC32, and AC46) were provided. In this case-study, the machine learning techniques were used to find the correlations amongst different features in the given dataset with the failures and faulty subsystems in the three Carnaghi machines. The data-driven models were trained using different machine learning methods with different features. Then, models were tested using unseen testing data. Accordingly, models and features that have the best accuracy were adopted for the data-driven service. The evaluation results are available in the deliverable of D4.3 In addition, the web-server, that provides machine status and isolate faulty elements, were tested.

2.2.8.7. Universal web based KPI visualization

2.2.8.7.1. Purpose of the solution

The target of the “Universal web based KPI visualization” is web based platform for homogenous presentation and visualization of decision support tools with respect to flexible and reconfigurable production systems. The solution offers a common platform for different decision support solutions coming out of the PERFoRM project. The visualization platform should be capable to communicate with the PERFoRM Middleware. A live data exchange between the different solutions connected to the PERFoRM Middleware is intended.

2.2.8.7.2. *Development goal in PERFoRM*

The development goal is to set up the visualization interface that is capable to communicate with the PERFoRM Middleware within the demonstration case at the SmartfactoryKL and the Siemens use case in Duisburg. To cope with the diverse set of requirements during the project duration, the visualisation is based on a modular assembly principle. All elements within the user interface (UI) are defined as entities. Entities can take several different forms and adapt to the needs of the user. An entity can, for instance, be a machine, a process, a whole factory, an analytic method, a simulation or a whole simulation environment.

2.2.8.7.3. *Current state*

The visualization module was successfully developed and implemented within the test bed of the TU Braunschweig. The visualization for the demonstration case (PERFoRM demonstrator) at the SmartfactoryKL is developed and now under implementation. The visualisation for Siemens in respect to the scheduling tool from IPB, the data mining solution from MTC and the data driven diagnostics for manufacturing equipment from Lboro is under development.

2.2.8.7.4. *Test / Demonstration results*

First demonstration results at the test bed of the TU Braunschweig are promising that the solution can be expanded to the use cases. Live data with presentation of complex calculations (presentation of harmonic response analysis depicted in Figure 44) are successfully visualized on mobile devices (tablet). The rendering speed, as well with high data quantities, is adequate for the presentation of complex decision support solutions. A roll out to the demo / use cases is indicated.

Figure 44 Harmonic response analysis of DMU100 within the test bed of TU Braunschweig

3. Use Case: Siemens

3.1. As-Is-Situation in the use case

3.1.1. Improvements intended through PERFoRM

Compressors are manufactured as three major subassemblies, rotor, stator and housing. The central manufacturing processes are machining, heat treatment, balancing, assembly, and painting. These processes are required for all compressors – independent of their type. In pre-fabrication, where the parts for rotors and stators are manufactured, mainly automated machining processes are present. In contrast, in the main fabrication - assembly, measuring, cleaning and painting - value is added manually, as the necessary works for testing and delivery are executed as well. For the selection of a concrete use case scenario, the whole production process at the factory was evaluated, leading to the selection of machining processes in the in pre-fabrication as well suitable to fulfill the goals and requirements of the PERFoRM project. Narrowing down the selection of particular equipment for being investigated within the project, three similar turning machines were selected.

As it is described in Deliverable 7.1 [17], the Siemens Use Case aims on improving the availability and reliability of production machinery and thus, as a direct consequence, the production flexibility at the Duisburg compressor plant of the company. This shall be achieved by preventing unplanned downtimes of machinery through an improved monitoring of the actual condition of the production equipment, followed by better planning of maintenance activities necessary to prevent breakdowns and repair. Further, respective measures shall be planned and executed with respect to the overall manufacturing schedule, in order to find suitable slots with minimal negative impact on the manufacturing execution. So, the major goals to be met are

- Raised machine availability through improved condition knowledge and maintenance activities,
- Better planning and scheduling of necessary downtimes of machinery for maintenance in order to reduce negative impacts on production figures (e.g. completion dates, downtimes, schedule deviations).

3.1.2. Installed Base

3.1.2.1. Infrastructure

In the existing implementation, all relevant systems are connected through the Manufacturing-LAN, which is a local, separate network only accessible from within the factory. During detailing the use case, the relevant legacy system components were narrowed down to the Machine and Production Data Acquisition System (“BDE/MDE”), and the the Maintenance Ticketing System (“LHnet”), compare Figure 45.

The MDE/BDE System collects production data from all machines in the factory in a centralized database (Oracle). Collected data, for example, are production tasks executed, productive state of a machine or alarm messages from the controls. Additionally, the BDE/MDE database stores data on planned manufacturing tasks, fed from the Manufacturing Scheduling System (“SAP APO”) to the storage.

For maintenance management, a separate system called “LHnet” is in place. The system consists of ticketing terminals distributed throughout the factory on the shop floor, which allow the workers to open tickets to the maintenance department in case of machine failures or breakdowns. Tickets are stored in a centralized, independent database. Reactions to failure tickets are done manually through the maintenance department. Besides reporting failures and breakdowns, the system is also used for documenting repair measures taken.

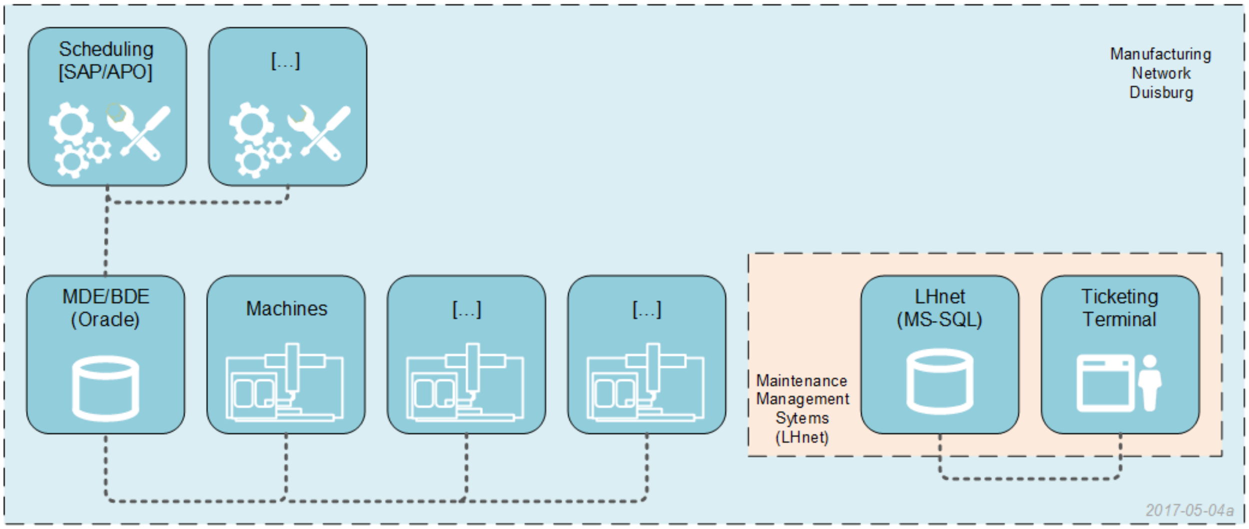


Figure 45: As-Is Architecture of the legacy System (visualization reduced on relevant system components).

3.1.2.2. Planning Processes

In general, scheduling of production tasks is done by the Planning department using the SAP APO system. Prioritization of manufacturing tasks is using the typical principles, like required ready dates, setup time reduction or economical figures / objectives when sequencing requested tasks.

In case of planned maintenance, suitable slots are adjusted manually among the departments involved, mainly planning and maintenance. Adjusted maintenance windows are fed to the scheduling system manually, so that scheduling can consider them iteratively (Figure 46). In case of unplanned maintenance / repair activities, the request for action is raised by the machine operator through the Maintenance Ticketing System, and is accepted by the Maintenance department followed by executing the necessary steps to repair the failing machine (Figure 47). Prioritization of repair measures is of course done based on severeness of the issue, external service or spare part availability, etc. The logical interaction of both cases is depicted in Figure 48, where the legacy systems are depicted in light blue.

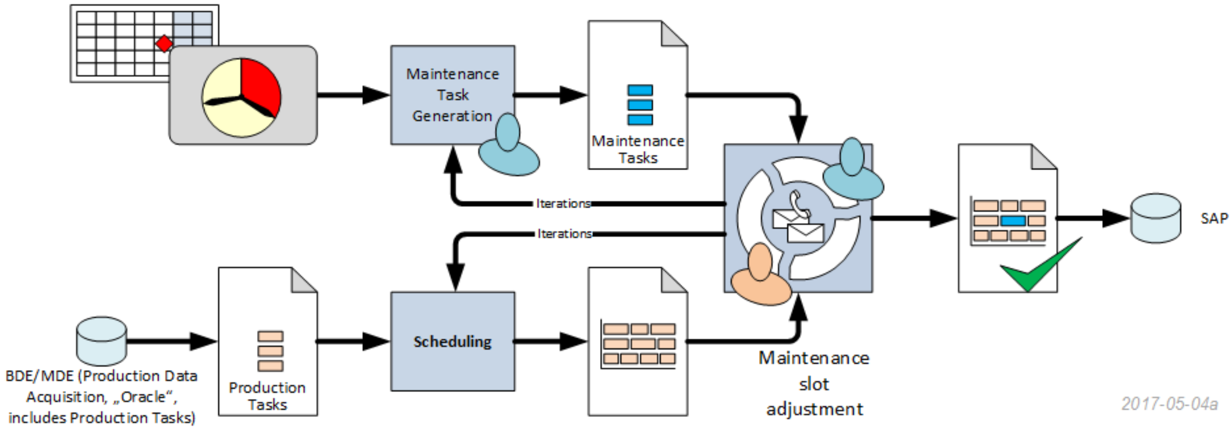


Figure 46: Process flow in case of planned maintenance.

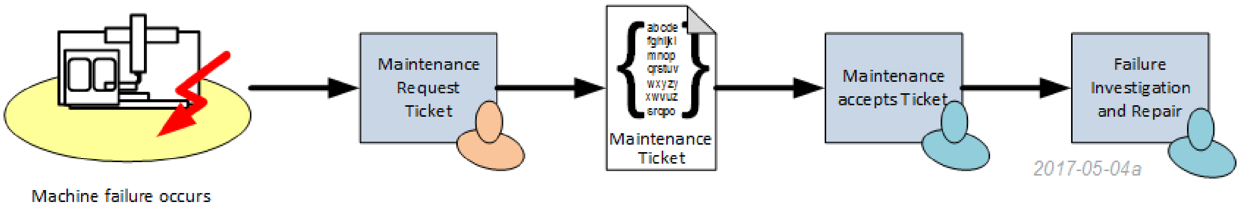


Figure 47: Process flow in case of Machine failures / breakdowns.

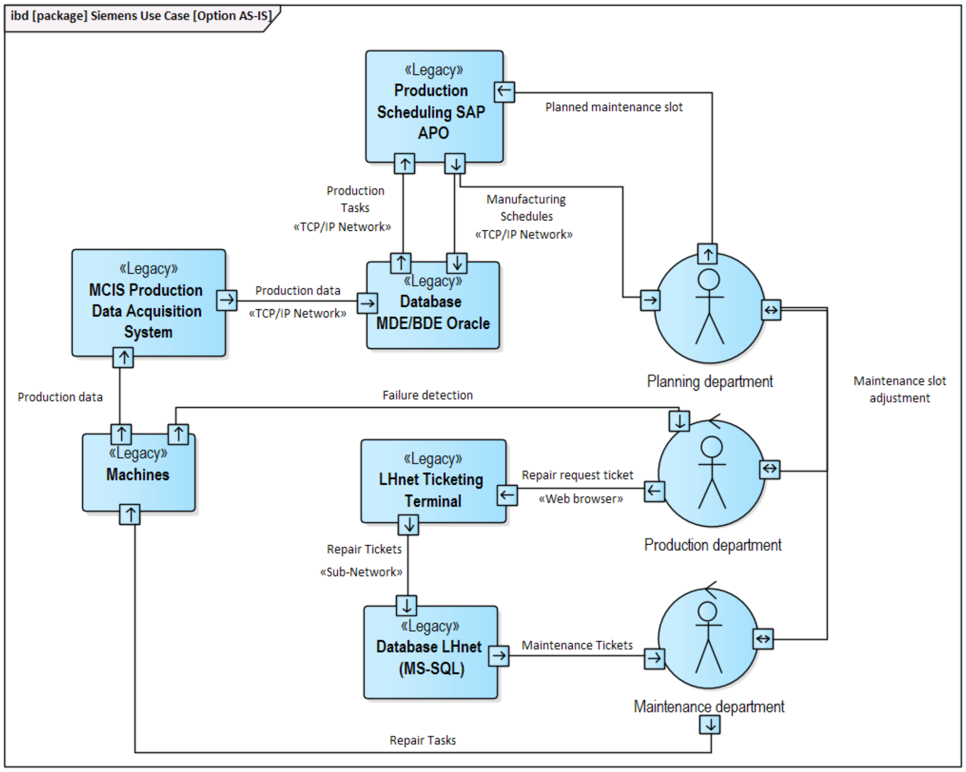


Figure 48: Logical interaction of the different roles involved in maintenance and repair.

3.2. Vision

In the long term, the Siemens Compressor Plant at Duisburg is going to holistically implement the Digital Factory Paradigms. Following the general definition, this means that all necessary steps for product realization – from early planning through development to manufacturing and use of products, as well as for supportive processes like procurement, sales, etc. – are closely interconnected using information and communication technologies. It is intended to achieve a strong, dedicated ability for self-adaptation, always based on the situational conditions found at a certain time.

In the context of the Use Case selected for the PERFoRM project, this integration and automatic adaption is fostered by the integration of the PERFoRM-middleware on the one hand, and, on the other hand, by starting the implementation of a system to prevent machine breakdowns by moving to a predictive instead of a reactive and time based maintenance approach. Once fully established, a complete integration of maintenance system in ERP/MES (SAP) is intended, providing automated maintenance task definition based on automated data analysis of machine health data. Therefore, although currently mainly addressing machine availability and better scheduling of maintenance tasks, the implementation of the PERFoRM Use Case represents a first step towards the long term goal of the digital factory.

3.3. Use Case Implementation

3.3.1. Intended Work flow

As introduced in section 3.1.1, the intention of the Siemens Use Case is to improve the flexibility of the overall production by raising the machine availability through better maintenance. At the same time, the maintenance tasks to be conducted shall be better scheduled with the actual manufacturing tasks in order to minimize negative effects of machines undergoing maintenance and not being available for production meanwhile.

With the system to be put in place as solution for the use case, several tools and services developed within PERFoRM are combined to form on the integrated solution supporting these targets. In the workflow created, as a first step data available through the legacy and newly installed systems – alarm messages from the machine controls, failure tickets from the maintenance ticketing system and sensor data from dedicated sensors applied to the machines – will be analyzed in order to gain knowledge on the actual machine condition. This analysis will be conducted in parallel to the manufacturing running, thus allowing to detect condition changes over time. As a second step, the results of the Data Analysis are visualized for maintenance personnel, who will then be able to define maintenance tasks required to maintain the machines availability for production.

As visualized in Figure 49, these first two steps are conducted for the three machining centers considered. Thus, the intermediate results are maintenance tasks defined with respect to the current condition of each machine. The tasks are not yet scheduled, but are defined as tasks comparable to the actual production tasks, which means in general they are defined as consuming a certain amount of time on a specific resource (Machine). Additional information is an earliest (e.g. for spare parts to be available) or a latest date (e.g. after which the breakdown probability becomes too high).

In the following third step, scheduling is conducted for the primary production tasks, together with the maintenance tasks. The scheduling is done for the whole manufacturing area in order to consider bottlenecks e.g. caused by delays through maintenance slots. At each time, several schedules are generated, differencing for example in weighting of defining parameters (e.g. do maintenance as early, as late as possible, highest accuracy of meeting delivery date, earliness, tardiness, etc.). For progressing, these schedules are fed to the production simulation step. In principle, the simulation step is used to consider unforeseen events in the overall production – delays, breakdowns of (other) machinery, etc. – and their influence on the suitability of each schedule. As result, for each schedule a set of indicators is calculated based on the simulation, which gives an indication to the planning as well as maintenance personnel who will select one schedule to be implemented as the last step of the PERFoRM Use Case System.

For the intended implementation within the PERFoRM Use Case, the maintenance task definition as well as the selection of the most suitable schedule will be done manually. The data analytics solutions are intended to analyze trends in the data available in order to indicate changing machine conditions. Possible later extensions of the system therefore include:

- Consideration of further (all) machines in the condition monitoring

- Enhancement of the trend analysis approach towards prognosis of future issues on the machines
- Suggestion of maintenance tasks based on prior problem solving measures
- Automatic selection of maintenance tasks
- Consideration of machine wear in the simulation
- Automatic Schedule selection
- Automatic feed back to the legacy system (Schedule to SAP)

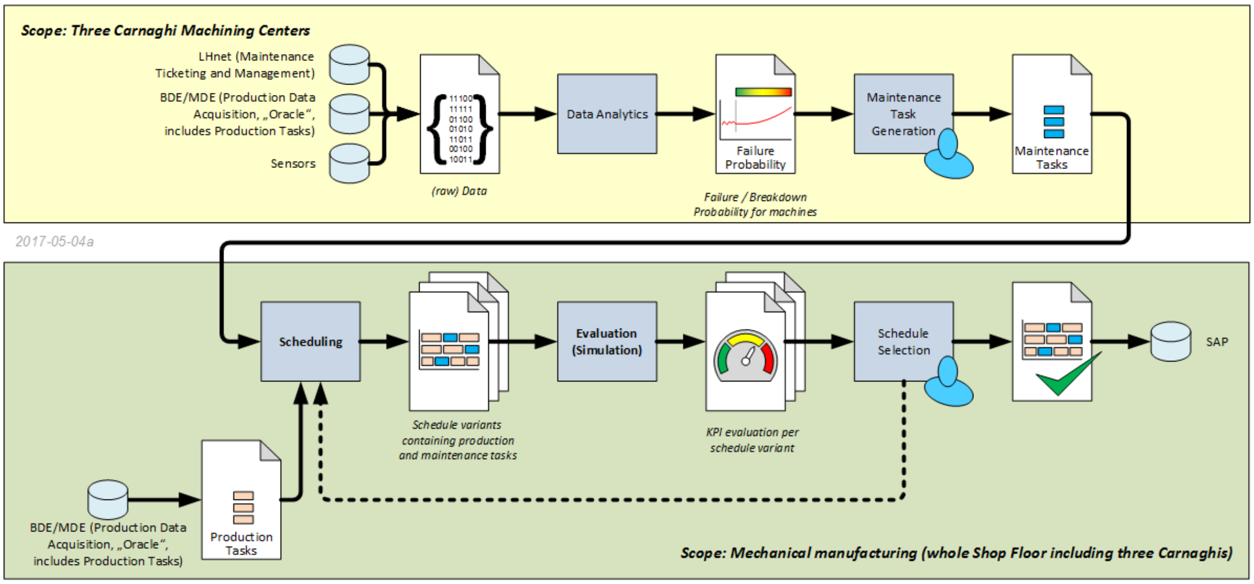


Figure 49: Designated Work Flow of the System implemented in the use case.

Figure 50 represents an overview of the logical interaction between legacy systems (in light blue color) and the new PERFoRM applications and adaptors (in light green). The human operator of Planning department is depicted in yellow since he will have to slightly change his skills to handle the new implemented systems. Also the legacy databases interfaces are colored in yellow because they will be used to extract specific data (described by connectors) for the new connected applications.

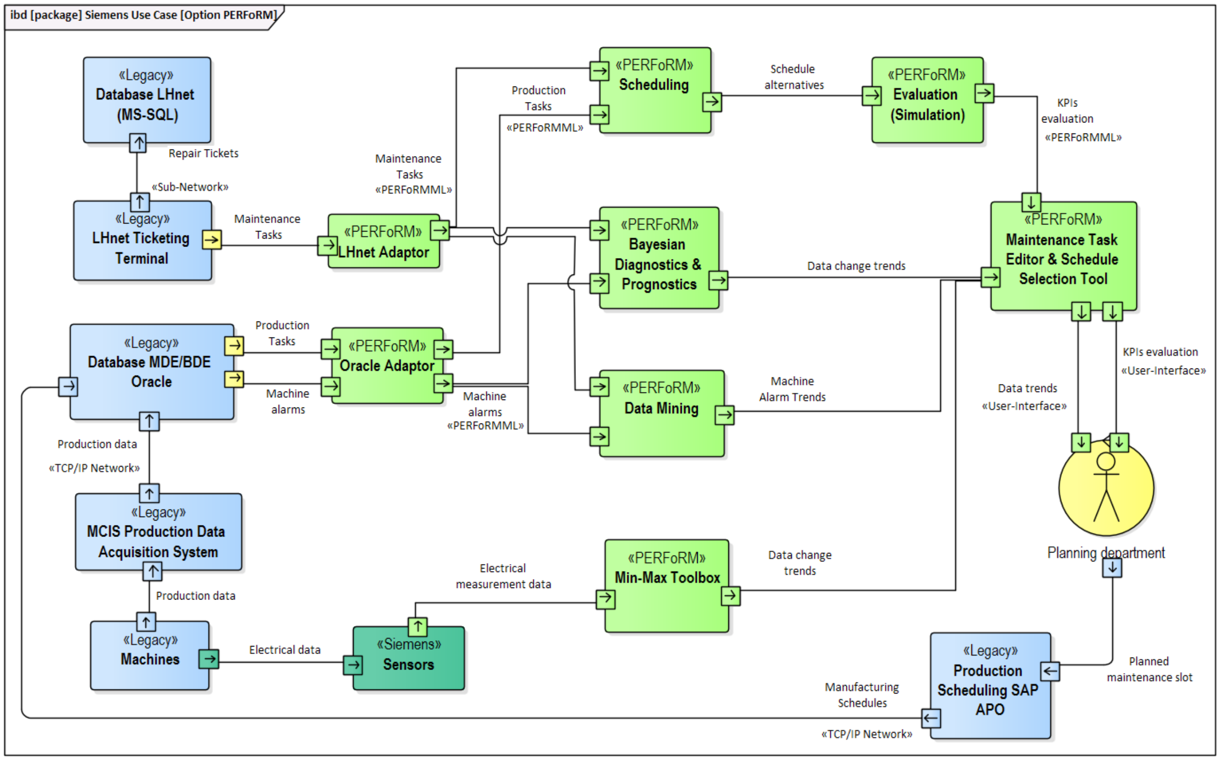


Figure 50: Logical interaction of the different roles involved in maintenance and repair as planned for the use case.

3.3.2. Implementation Architecture

For implementing the functionality to support the work flow as described in the previous section, a combination of several building blocks developed throughout the work packages 2 to 4 of the PERFoRM project will be set up at the Duisburg factory (see Table 2).

As a basis, the PERFoRM Middleware will be used to ensure the communication and interaction of the several solutions, which use the defined Data Model and Standard Interfaces to do so. The connection to the legacy systems is done via two Technology adapters, which are the Database Adaptors for Oracle and MS-SQL databases. Scheduling, Simulation as well as the interaction with a system user will be present once per category, three Data Analytics tools/services are going to be implemented.

Table 2: PERFoRM Solutions from WP2-WP4 applied in the Siemens Use Case

Area	Solution	Section in Document
Middleware	Industrial Middleware	2.2.1
Technology Adapters	Oracle Database Adapter	2.2.3.1
	MS-SQL Database Adapter	2.2.3.2
Data Exchange	Data Model and Standard Interfaces	2.2.2
Data Analytics	Data Mining	2.2.8.4

	Min-Max-Data-Mining-Toolbox	2.2.8.3
	Bayesian Diagnostics & Prognostics	2.2.8.6
Visualization	Maintenance Task Editor & Schedule selection Tool	0
Scheduling	Multiobjective Scheduling of production orders and maintenance tasks (bio inspired)	2.2.7.1
Simulation	Simulation Environment	2.2.6.1

For the realization of the use case, the PERFoRM Middleware and Services will be hosted on dedicated Hardware at the factory. This includes a standard PC which is going to host the middleware itself, as well as most of the services. Depending on the performance requirements of the several services, it is possible that during the implementation and testing this concept is enhanced to host middleware and services on different hardware, which are then connected through a standard network, which means no difference for the service implementations due to the middleware approach (services are always communicating through network/middleware). In addition to the PC-hosted tools/services, the “Min-Max-Toolbox” Data Analysis Service is implemented using dedicated hardware (Microcomputer Units, Sensors) which are mounted directly in the cabinets of the machines considered for the use case (compare Figure 36 in section 2.2.8.3.1).

The necessary connections to legacy systems will be realized through the manufacturing local area network. For the first implementation, the first transition step towards the overall vision though, connections to legacy systems will be read only for preventing unintended interference with the factory systems by accident. Necessary data flows from the new PERFoRM- to the legacy systems (“write connection”) will be done manually. In the realization, this means that schedule adaptations defined by the PERFoRM system will be manually transferred to the existing SAP APO Scheduling.

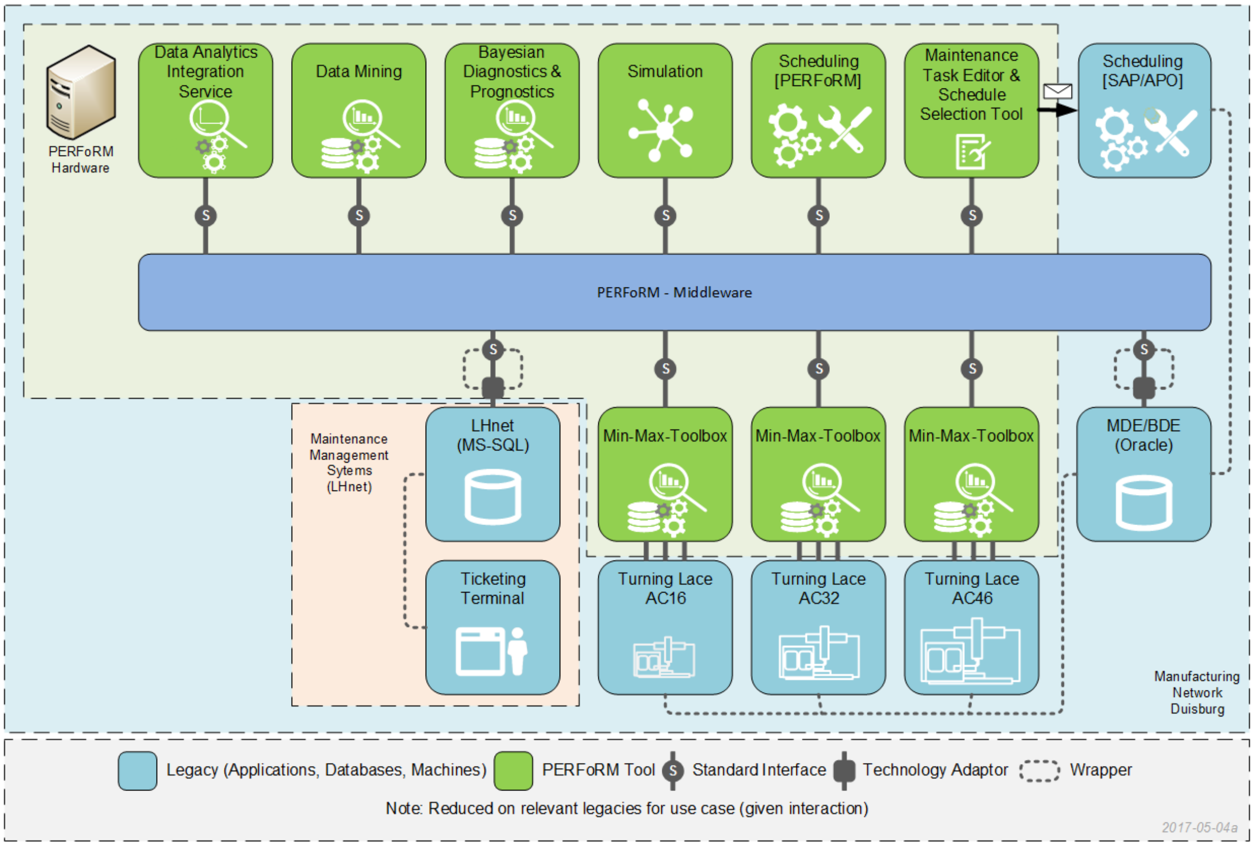


Figure 51: Planned Architecture for the Siemens Use Case Implementation.

3.3.3. Required Testing

Since in case of the Siemens Demonstrator, several different Tools and Services (see Table 2) are to be connected, integration is a major concern. Therefore, integration tests are to be conducted, meaning that in a test environment the interaction of all those Tools and Services will be tested.

At the time of the deliverable writing, preparations for this are ongoing in terms of setting up the middleware backbone as well as database replicas for emulation of the legacy systems. It is planned to start the integration tests at the end 2017 / beginning of 2018.

4. Use Case: e-District

4.1. As-Is-Situation in the use case

4.1.1. Improvements intended through PERFoRM

Through PERFoRM I-FEVS and Polimodel aim at making available low cost automated turnkey flexible assembly lines to rapidly start the manufacturing of safe, ergonomic, clean and efficient vehicles adapted to local needs. The proposed experimental assembly line has been originally conceived for 50 vehicles a day over two shifts.

4.1.2. Installed Base

Currently E-District uses a complete manual assembling line with no legacy systems to adapt to the new architecture that will be developed in PERFoRM.

4.2. Vision

The higher degree of automation achieved through PERFoRM is expected to improve the efficiency and reproducibility of the processes while enabling higher product qualities. The proposed approach is also expected to reduce re-work of sub-modules and in some cases part rejections incurring energy and material losses.

To mitigate the variability related to low volume specialized productions we also envision the adoption of plug and play hardware - software to implement the cloud-based CPS architecture with services aimed at predictive maintenance methods, supported by knowledge based concepts of self-learning and self-adaptation. At full capacity, the factory will have a CPS architecture that self-adapts and self-optimizes thanks to the feedbacks given by the operators, the quality engineer and the various sensors placed in the working areas.

E-district aims at the physical implementation (HW/SW) of a demonstrator representing a real working island of the assembling line.

This demonstrator platform will consist of a MES that will contribute to the following E-District's goals:

- Production scheduling
- Traceability of production
- Traceability of used materials
- Storage of data in cloud (no ERP system will be required)
- Control of the production flow
- Visualization of the KPIs

4.3. Use Case Implementation

4.3.1. "Work flow"

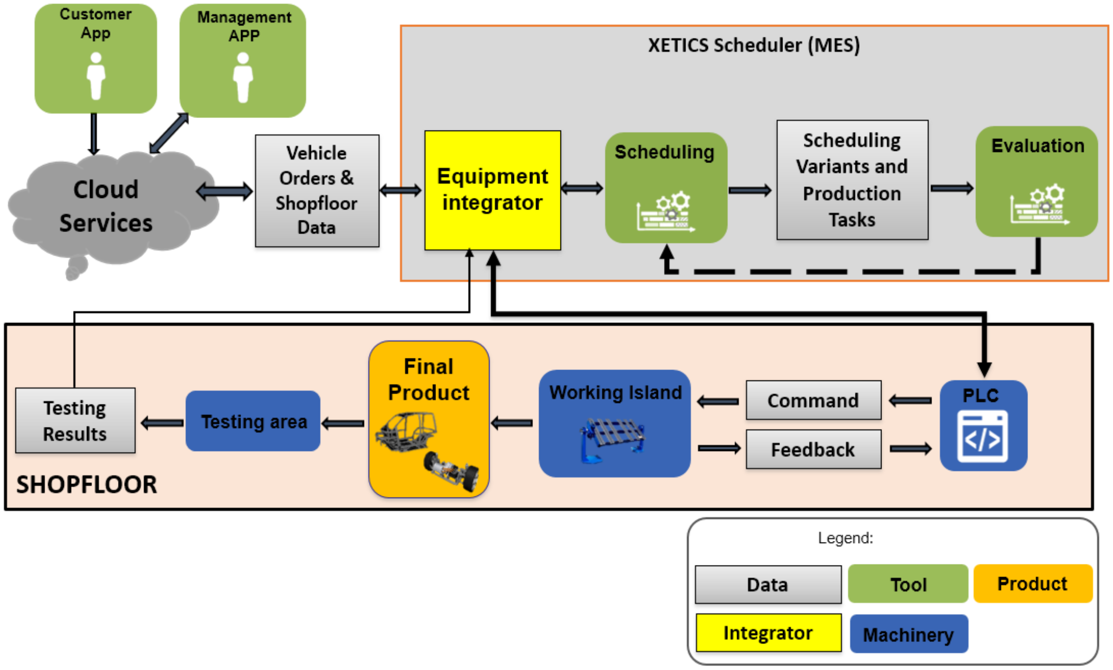


Figure 45: Targeted Workflow of the PERFoRM Demonstrator Implementation.

The architecture of the demonstrator is based on the following tools and machinery:

- MES:
    - Accepts orders inserted manually or from the cloud
    - Monitoring and visualization of the KPIs
- Scheduler:
    - Scheduling Service which generates suitable schedules following alternative goals
- Working island:
    - Sensorized working island including PLC and HMI
- Testing area:
    - The test results are stored within the MES.

4.3.2. Implementation Architecture

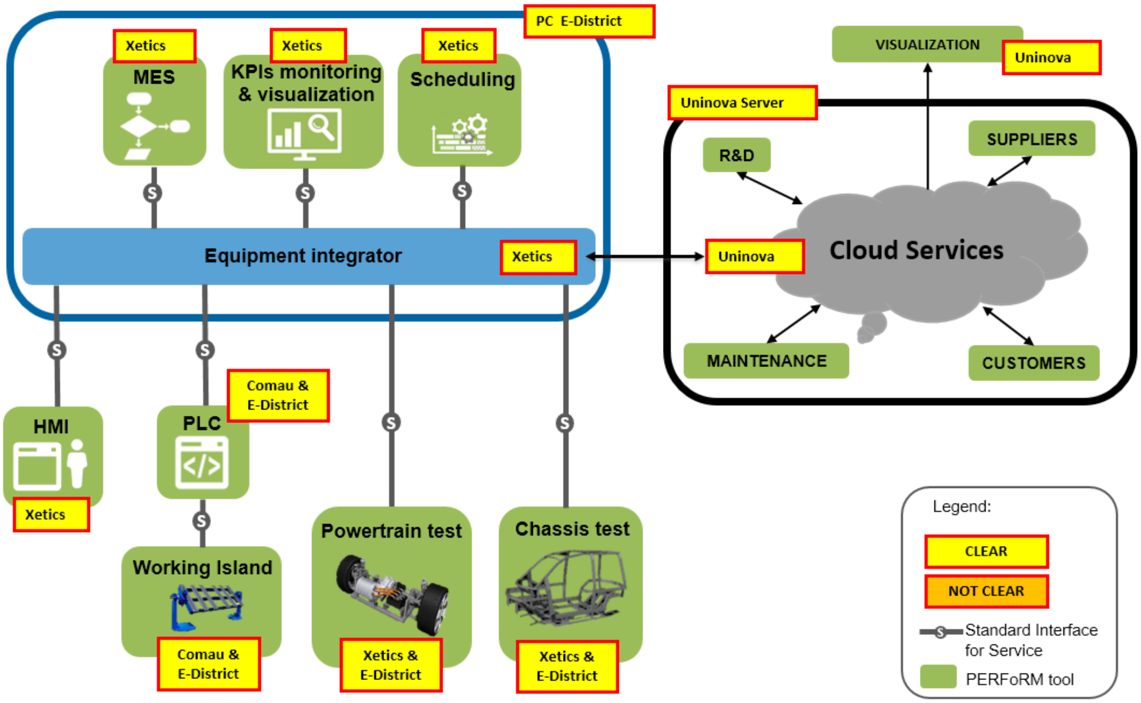


Figure 46: Targeted Architecture of the PERFoRM Demonstrator Implementation.

PERFoRM MES will run on a dedicated PC hosting also the scheduler and KPI monitoring and visualization developed by Xetics.

A complete working island will be implemented hosting part presence sensors and a plc.

The HMI developed by Xetics runs on an android based portable device as shown in picture 47.

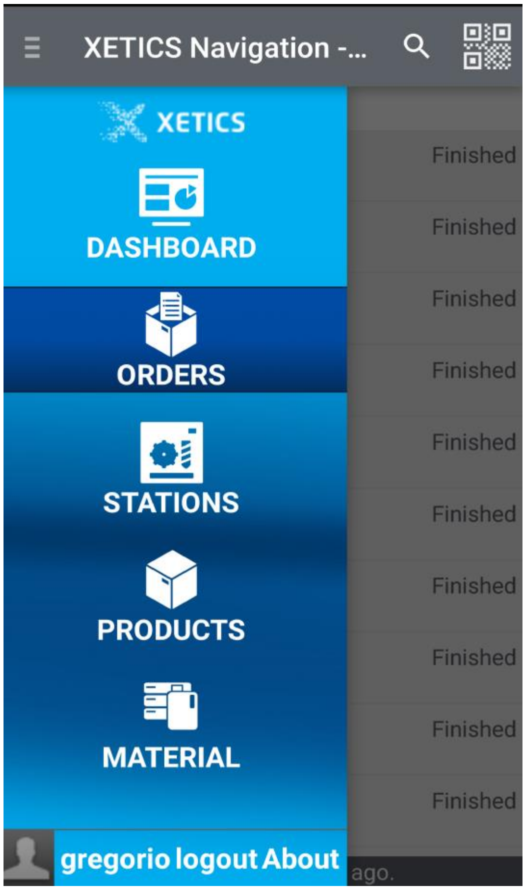


Figure 47: Android based HMI developed by Xetics.

The scenario to be tested is the following:

1. Order is activated
2. A workpiece is placed at the assembly workplace
3. Sensor 1 signals IN
4. User
    a. starts with the assembly
    b. starts the job on the app (Move IN)
    c. work instructions are displayed
    d. has finished with the assembly
    e. Quality parameters are queried
    f. Number of parts and scrap can be edited
    g. Finishes the job on the app (Move OUT)
5. Work piece leaves the assembly station bit position
6. Sensor 2 signals OUT

Xetics is currently testing the environment in its facility using the protocol IO-Link, and the IO-Link Sensor and IO-Link Master components with integrated OPC / UA server (more details in the deliverable 6.6). The goal is a seamless communication between sensor and XETICS LEAN for fast, flexible, agile and efficient production.

This solution will be used for the E-district's demonstrator because of its simplicity and adaptability.

The architecture is under completion together with XETICS aiming at its simplification and optimization.

The cloud environment (by Uninova), intends to bring the new concept of cloud manufacturing to the production of the electric cars. The idea is to develop a cloud environment, deployed into a dedicated server, and available on the internet, capable of storing relevant data extracted from the shop floor and share this data with external web based applications which can consult this information at any time and any place (see Figure 47).

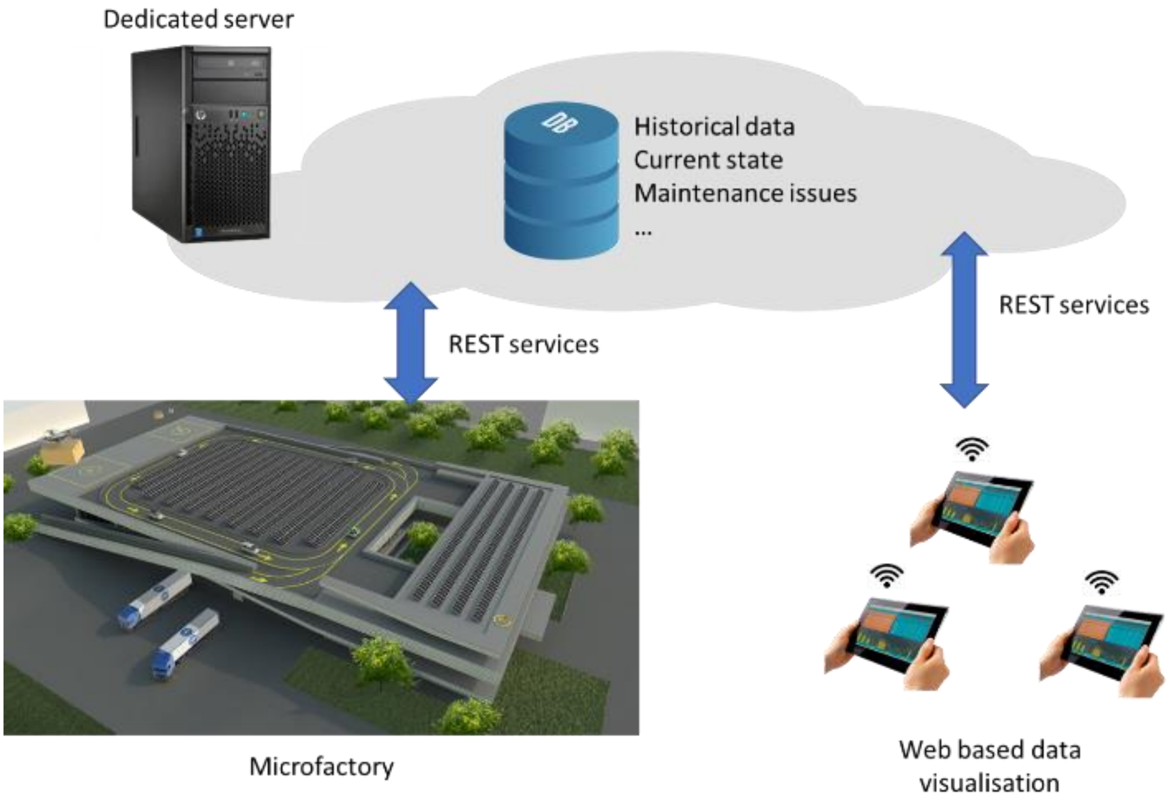


Figure 47 - Cloud environment for the E-district use case

The development of this cloud environment encompasses the implementation of three different software modules:

- SQL database: This database will be responsible for storing and organizing the information from all the available products, customers, suppliers among other information. It will be updated with information sent from the shop-floor (middleware) and from external entities (suppliers, costumers, engineers, etc.).

- REST services: The REST services work as interfaces to and from the exterior. Whenever information should be written into the database or retrieved from the database, one of the proposed services should handle that job.
- Web based data visualisation: The visualisation tool will be developed in a web based technology, this will allow any device to consult and interact with the cloud environment (Computer, Smartphone, Tablet, etc.).

For implementing the functionality to support the work flow as described in the previous section, a combination of several building blocks developed throughout the work package 2 of the PERFoRM project will be set up at the E-district facility.

Table 3: PERFoRM Solutions from WP2 applied in the E-district Use Case

Area	Solution	Section in Document
Visualization	Automatic Monitoring and visualization of KPIs	05
Scheduling	Xetics planning board	2.2.7.5.3

4.3.3. Required Testing

Xetics is currently testing the MES with the feature requested by E-District. In particular:

- Traceability of production
- Traceability of used materials

Xetics is also testing the connection between optical sensors and the MES.

In the Table 3 below, the different sources and destinations of communication activities are listed down respectively on the left and on the top side. The description of each connection is reported below with a focus on the IT-Level.

Table 3: Connection description

ID	Description
F-IF-01	Summarizing and retrieving data from Sensors, it is possible to aggregate them and to evaluate the overall production activity performance
F-IF-02	The awareness of production trend is reached collecting production material consumption parameters

F-IF-03	Product identification is obtained through RFID (see F-IF-01) and barcode that provide information about internal component production, their serial number, part-list and finished good
F-IF-04	The production is checked and controlled to optimize the moment to start production, to change product item or to switch batch realization
F-IF-05	Summarizing and retrieving data from Equipment, it is possible to aggregate them to evaluate the overall production activity performance
F-IF-06	The awareness of production trend is reached collecting production and material consumption parameters
F-IF-07	Production orders are listed with a temporal sequence and item priority
F-IF-08	The awareness of long term planning is obtained to provide a list of products to be realized and material to be requested.
F-IF-09	Production Plan details are provided to perform the scheduling activities
F-IF-10	Production orders including temporal sequence, product details in terms of serial number and BoM are provided
F-I-01	Display performance indicator to support decision maker activity
F-I-02	KPI results are provided to each operator
F-I-03	Principally finished and semi-finished good declarations are communicated.
F-I-04	Each data is collected in the cloud to speed up the information procurement.

5. Use Case: Whirlpool

5.1. As-Is-Situation in the use case

5.1.1. Improvements intended throughout PERFoRM

Whirlpool EMEA manufacturing footprint consist in more than 20 plants across Europe and South Africa. These plants are governed, from the aspects related to continuous improvement and for transformation, from a central department named Operation Excellence which dictates standards in terms of Production System (based on World Class Manufacturing) and Technology. Factories are typically dedicated to a specific family of products to increase specialization and optimize competencies and economy of scale and as much as possible they are standardized in some common aspect (value stream definition, basic principle of lean manufacturing etc.) Production plan in each factory is decided by a unique system (SAP APO) and, generally rescheduling is not an accepted practice. Batches, pitches and takt time are defined for each Value Stream and cannot be changed. In this perspective, the word "reconfiguration" is to be intended as a practice of re-design the factory or part of it (either a production assembly line, an intermediate warehouse, implementing a kan-ban, activate an additional shift) which typically occurs two to three times a year for each factory.

Of course each factory reconfiguration, usually the result of a WCM activity called Cost Deployment aimed at identifying improvement directly driven by a cost reduction (or a wate eleimination), is typically oriented at addressing a small part of the factory itself, and is measured by verifying some of the specific KPI of the factory scaled down.

The big challenge that WHR would PERFoRM to face is the current limitation of the reconfiguration process, which is lead by team of local and central engineers, and is carried out with very little support from ICT tools:

1) Impact of reconfiguration (expressed in change of some operating and descriptive paramenters called KBF, Key Business Factors) to KPI is otften only estimated
2) Reconfiguration is designed based on a data gathering which is often incomplete and resource intensive
3) Reconfiguration approach is based on trial and error approach: variants are analyzed using expert estimation and results are gathered after the execution of the reconfiguration.

This situation, in the complex situation of factories made up of different sub-system, very often resulting of brown filed progressive modification, lead to a sub optimized process of reconfiguration, characterized by long time of execution, intensive use of resources, limitated use of divergent process to increase innovation and out-of-the-box thinking.

PERFoRM can, with a proper utilization of available data currently gathered by SAP and MES tools available at factory level, improve the speed and accuracy of reconfiguration effort and enable a more pervasive continuous improvement process, conducted on a day-by-day approach through a easier distributed Data Visibility.

Through the implementation of the project, Whirlpool wants to:

- Minimize change-over effort; where change-over is to be intended as general transformation of part of the factory that could be implemented without significant physical investment. This is a typical situation occurring when a small change is required by the market, such as limited capacity increase, New Product Introduction, Value Stream Re-design.
- Allow fast integration of automation systems into wider company ICT infrastructure; the robot Reconfiguration demonstrator will allow future implementation of robotic system which are getting instruction directly from PLM: whenever the information about a product is available at design stage and can be used to program a robot (or in general an automatic machine) it will be directly used and thus achieving the elimination of intermediate stage of information processing, reprogramming and testing.
- Improve visibility of real-time production system status for decision makers. Data is the basis of Cost Deployment process, which is, as said before, the master process to identify improvement opportunities and that is run at least every year. But all the decisions, being there day-by-day or step improvement, are expected to be more accurate, faster and more robust when a strong correlation between source data and KPI. So visibility of data is not only improving typical process of continuous improvement methodology, but they are also enabling a more solid management of production at each level, from workstation, trough shop-floor, to upper management.

5.1.2. Installed Base

Whirlpool Manufacturing installed base and legacy system is described in detail in Deliverable D9.1 [18]. We report here below the picture and description of the system, extracted from the above-mentioned document.

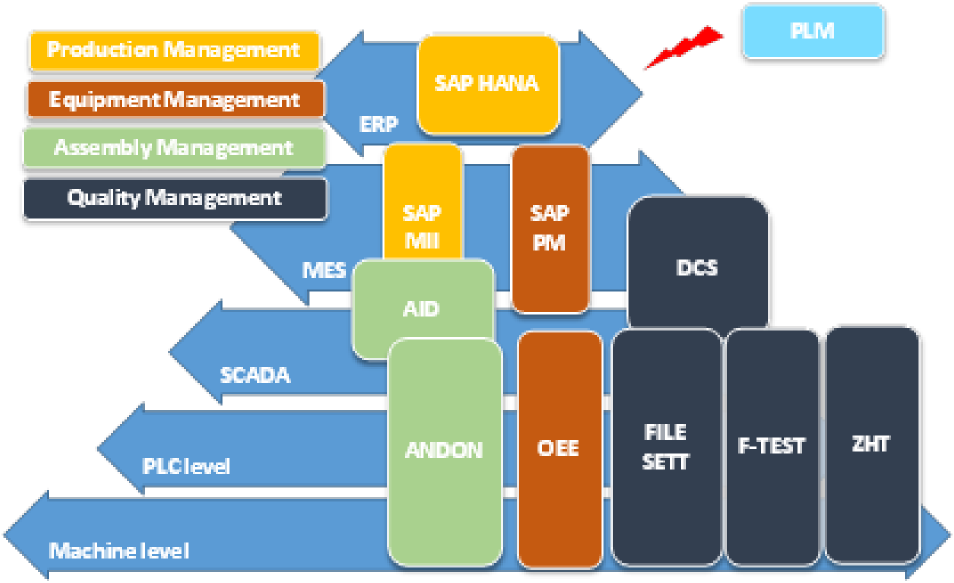


Figure 52: Current Automation architecture based on ISA-95 pyramid

Whirlpool Manufacturing Information system is currently the result of a combination of evolutions of separated components. The main architecture sees SAP Hana as main ERP system acting also as the main link with other company information system (finance, HR, Procurement etc.). At lower levels of the classic ISA-95 pyramid, there are several applications which have been logically grouped in four categories: Production management, Equipment Management, Assembly Management and Quality Management.

The current degree of vertical integration between subsystems is not at the required level to secure a migration towards Industrie 4.0 concept, and it has been mainly developed on the terms of need-to-serve basis, i.e. each application requiring a data share or a communication with other application is modified / equipped with a specific interface.

From the standardization point of view, there is a solid standard at PLC level: all factories are equipped with Siemens S7 PLC using both Profibus and Ethernet as a communication channel.

At upper levels the presence of Windows operating system and SQL Server DB is prevailing, even though there are some isolated systems using different approaches (e.g. Firebird database for ANDON and OEE tools)

5.2. Vision

The main goal of the integration is to flatten the present automation architecture with a middleware based approach and allow a novel simulation system to get connected with real data coming from field through a factory DB. This new approach should enable more robust and solid reconfiguration of factories KBFs.

5.3. Use Case Implementation

5.3.1. Factory reconfigurationWork flow

5.3.1.1. AS-IS Workflow

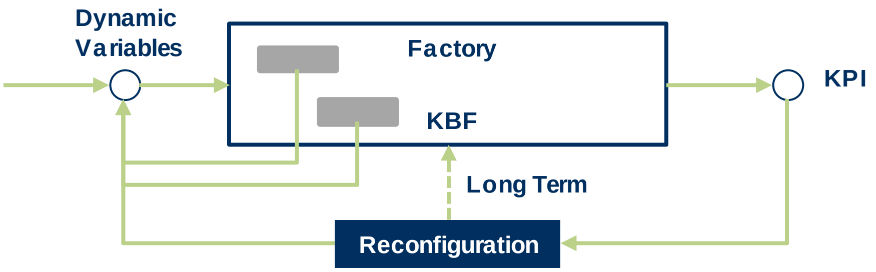


Figure 53. AS-IS workflow

Factory KPIs are the sum of all the outputs coming from the legacy systems.

The reconfiguration activity is carried out by industrial engineers, who are in charge of reconfiguring the factory's set-up in response to specific events such as factory masterplan, profit plan, new product introduction, etc.

It is important to note that the visibility of the potential correlation between internal KPIs and the factory's configuration is mediated by time factors (i.e. the visibility is not real time), and that there is no direct correlation among the KPIs.

5.3.1.2. TO-BE Workflow

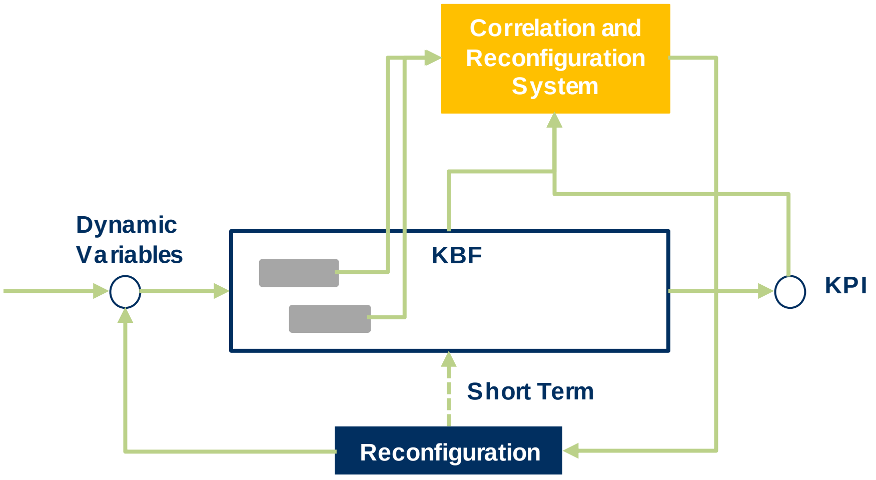


Figure 54: TO-BE workflow

In contrast to the AS-IS situation, in the TO-BE scenario there will be direct and real-time visibility on the potential correlation between internal KPIs and the factory's configuration. The employees in charge of the factory reconfiguration have direct visibility on the KPIs, and have available a system able to perform simulations, with a greater amount of data, and able to generate projections.

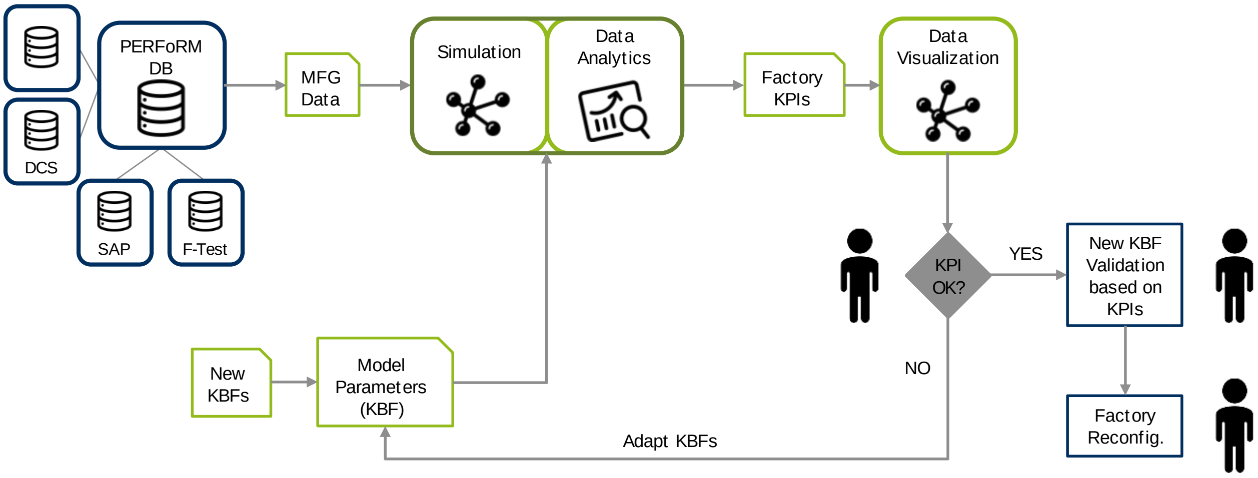


Figure 55. TO-BE user story

Manufacturing data coming from the shop-floor are stored in the PERFoRM database, and feed software applications for simulation and data analytics. The elaborations of these applications giveus an output the calculation of factory KPIs, which are provided to the reconfiguration operator in a visual manner, thanks to a data visualization application. The operator verifies the KPIs consistency with expectations:

- In positive case, a new validation of KBFs is performed and the factory re-configuration is subsequently executed;
- In negative case, KBFs are adapted and/or modified and another process iteration is executed.

5.3.2. Implementation Architecture

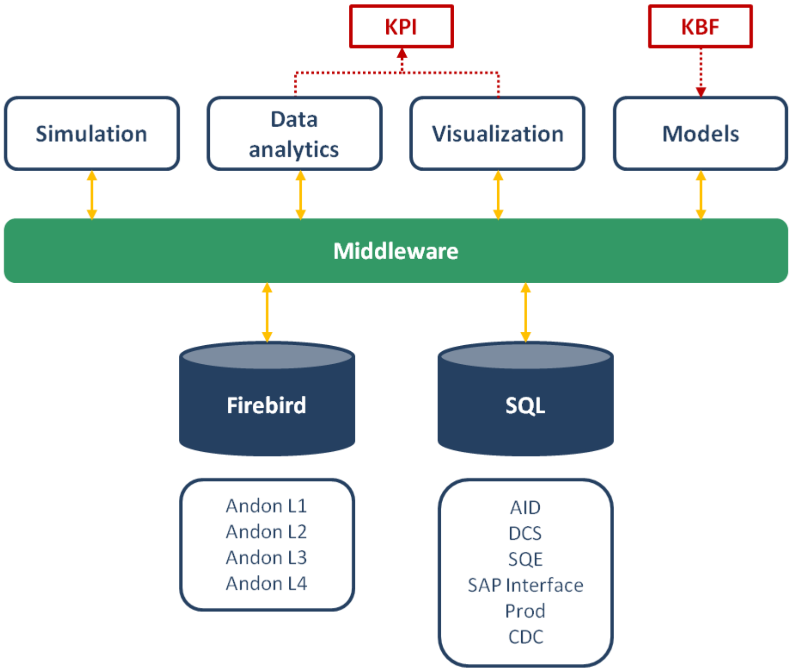


Figure 56. Implementation architecture

The implementation architecture’s central element is a Middleware, which acts as a connection between the three developed application tools (i.e. Simulation, Visualization and Data Analytics), the factory model and the SQL and Firebird Factory DBs.

5.3.3. Robot Reconfiguration Workflow

5.3.3.1. As-is reconfiguration workflow

Ijn the As-Is situation the programming of the robot to test the leakage is a complete manually operation done at factory level and involving in some cases external third parties. The information flow is unstructured and is based on e-mail communication and/or physical meeting in order to let the robot programmer understand the new requirements for leakage testing. The robot reconfiguration is then performed and tested offline, usually during a production stop.

5.3.3.2. To-Be reconfiguration workflow

In the PERFoRM workflow the information generated at PLM level is automatically transferred to robot instruction

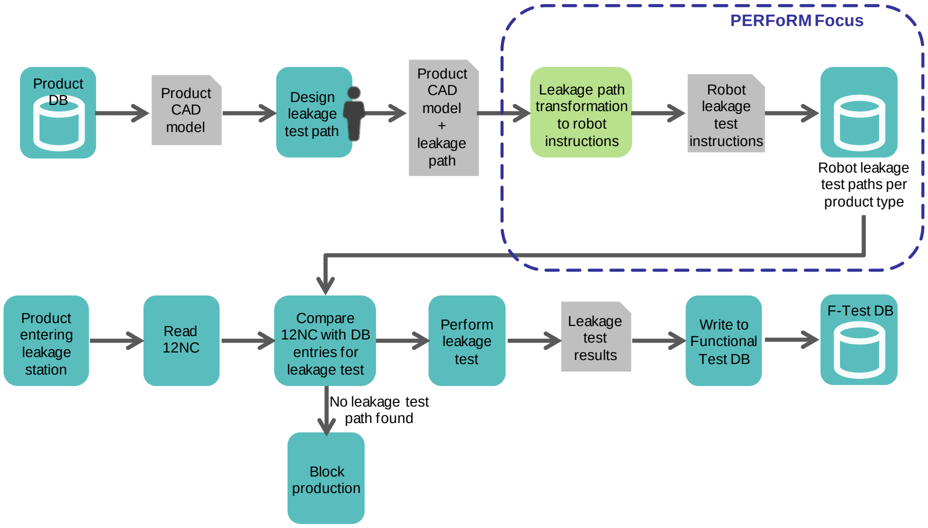
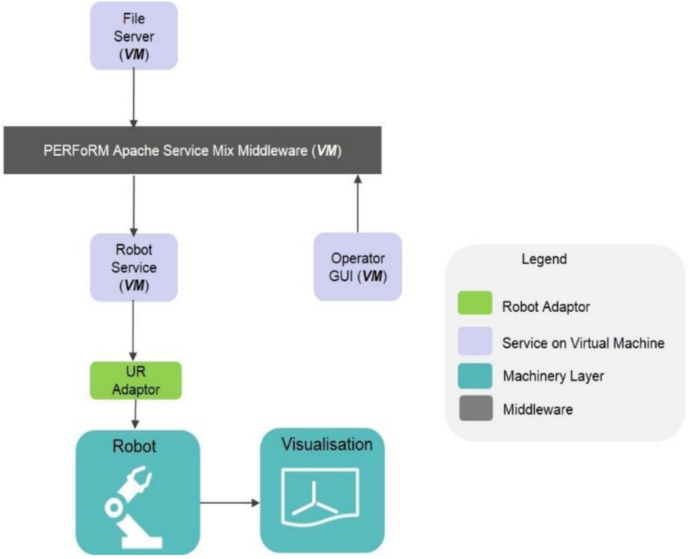


Figure 57: Robot reconfiguration workflow

In the migration optic the Robot service will receive fundamental information (such as CAD files) in native mode from Whirlpool PLM system.



5.3.4. Technical Solution implemented:

Technology Adapters	MS-SQL Database Adapter	2.2.3.2
	Robot Adaptor	2.2.3.3
Simulation	Excel Based Simulation	2.2.6.2
Decision Support	Value Stream Model Excel	2.2.8.1
	KPI monitoring with What-if	2.2.8.2

5.3.5. Required Testing

Test Scenario ID	Test Scenario	Priority
TS-WHR-F-1	Correct population of DBs with useful data to test activities	(only for validation process)
TS-WHR-F-2	Simulation activity on process production behavior with KPI evaluation	High
TS-WHR-F-3	Data aggregation using Data Analytics in order to identify correct information about production process	High
TS-WHR-F-4	Visualization activity needed to point out the evaluation of data coming from both simulation and field activities	High
TS-WHR-F-5	Data population of PERFoRM DB using PLCs and OPC-UA interface	Low
TS-WHR-F-6.1	Decision maker activity: a new set of KBFs are defined to optimize the process and to remake the simulation	High
TS-WHR-F-6.2	Decision Maker activity: Field configuration	High

6. Use Case: GKN

6.1. As-Is-Situation in the use case

6.1.1. Improvements intended through PERFoRM

GKN Engine Systems manufactures engine components to almost all new large aircraft jet engines worldwide in new engine programs The specialization areas are structural parts like fan cases, fan hub frames, intermediate cases and rear frames. The focus on light weight solutions and the strategy to use fabrication (i.e. welding smaller parts into sub-assemblies and final components) and Additive Manufacturing is one driver for automation and flexibility to manage a growing number of parts and processes.

From studies of operation lists for different product types and product families it is clear that many components need operations that can be automated. There are also sequences of relatively short and simple operations for e.g. surface improvement, deburring cleaning, inspections, NDT in various combinations. The production system idea as a starting point for the project, was to develop a modular production cell concept that can be reconfigured with different automated or semi-automated processes. The cell and process modules should be easily and quickly changed depending on current production demands aligned with the ideas of plug-and-produce for cyber-physical systems.

The goal with GKN participation and the use case in the PERFoRM project is to develop and demonstrate such a production cell system that be a contribution to the industrial system solutions needed, see Deliverable 10.1 [19] for detailed description. The business oriented criteria are to reduce lead times and increase the level of automation as well as equipment utilization. For this purpose, mechanisms for the seamless reconfiguration of the production process should be addressed by plugging-in/-out modular processes in robotic stations. From the analyses and conclusions of the current system, future needs and opportunities, there is a clear interest and potential in the Micro-Flow cell concept from different functions and stakeholders in the organisation.

Several technical and organizational challenges need to be addressed, namely interfaces for process modules that allow simple and short change over time, methods for production cell planning and scheduling to maximize throughput, decision support to identify when to reconfigure the cell to have the best impact on cost and production lead time, and procedures to coordinate the human intervention in the reconfiguration phase.

Some benefits will come from the modular approach for the Cell concept, to reconfigure the cells, and the technologies that it is based on. The system architecture, data models and communications are other very important part of the solution to be successful that also opens up possibilities for planning, scheduling and better use of data etc. However, not to forget that a somewhat new mindset about production and process planning etc. will be needed to utilise the advantages of flexibility, reconfigurability and the improved data communication.

6.1.2. Installed Base

The current workshops are often organised in different component shop areas, or value streams that produce one or several product families. However, due to the high product variety and to the different operations needed, some in shared resources, the production process is planned to operate according to job shop logic and mainly a functional layout. Therefore, products need to be transported and stored several times to the different workstations, creating a complex and not very visual and controllable product flow with many information interfaces. This is illustrated in Figure 58 below.

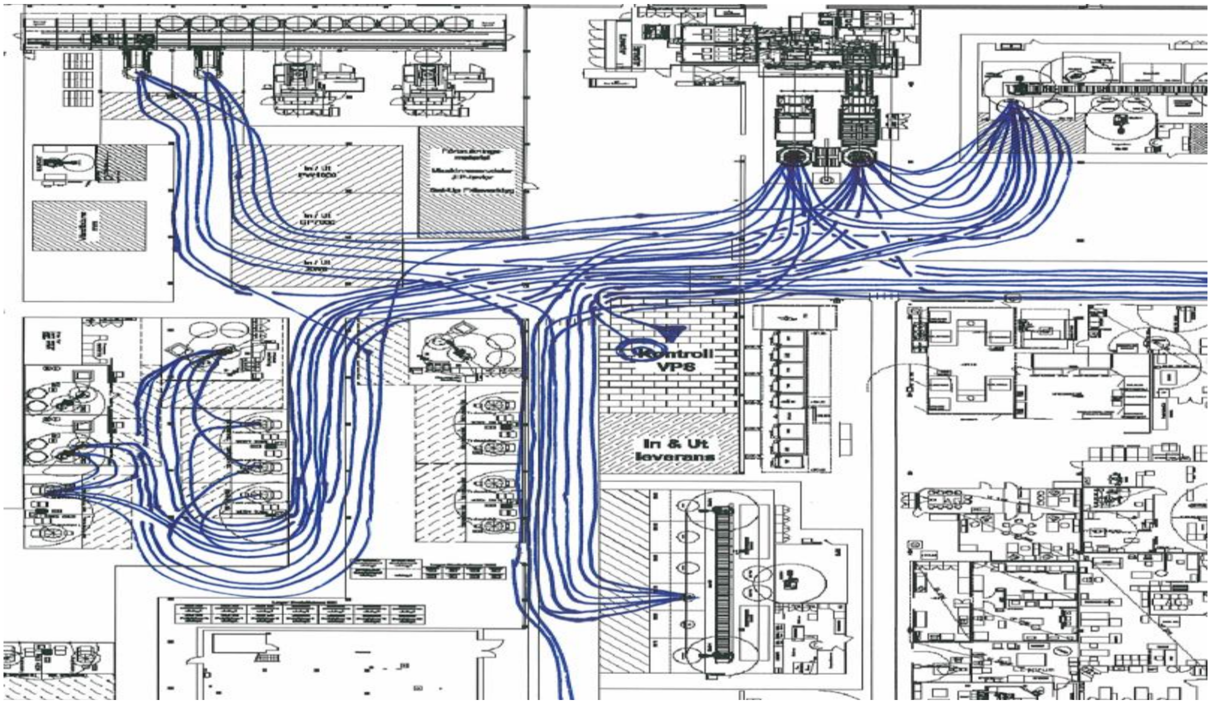


Figure 58: Example of shop floor routing ("spaghetti") for a component that requires transportation between each separate WorkCentre.

Besides the typical machining centres, a number of different of automated or semi-automated processes are used. See Figure 59. However, the level of automation is sometimes low and based on separate process automation cells with low level of process flow integration. The drivers behind this development are often a combination of ergonomics, environment and quality requirements.

Figure 59: Examples of automated processes.

The production system and its equipment's has a certain degree of flexibility and potential for reconfigurability, however it is often difficult and requires much effort from manufacturing engineering to utilise those possibilities.

The current shop floor IT systems and network is designed and configured mainly for straight communication between different IT systems to CNC-machines, robots and other equipment. The information flow in the typical day-to-day process is to down load programs to CNC machines, robots and other equipment that need specific control code, and upload data such as measurement data from CMM or CNC machines. Also some condition or process monitoring as well as stops are logged in the OEE system. There is no typical SCADA or MES system and not much communication between different machines or cells, other than in a well define work centre or cell, typically robot to PLC. All data (master data) is stored and managed through the SAP ERP system, which provides the functions for production planning, scheduling and MRP and it also collects different kind of documentation from the processing and inspection in order to guarantee traceability. The near term production planning and scheduling is done by shop floor planners based on ERP data for long term order scheduling and customer demands. As illustrated in Figure 60, the current architecture has mainly a „vertical“ data flow and very little of a „horizontal“ integration and control.

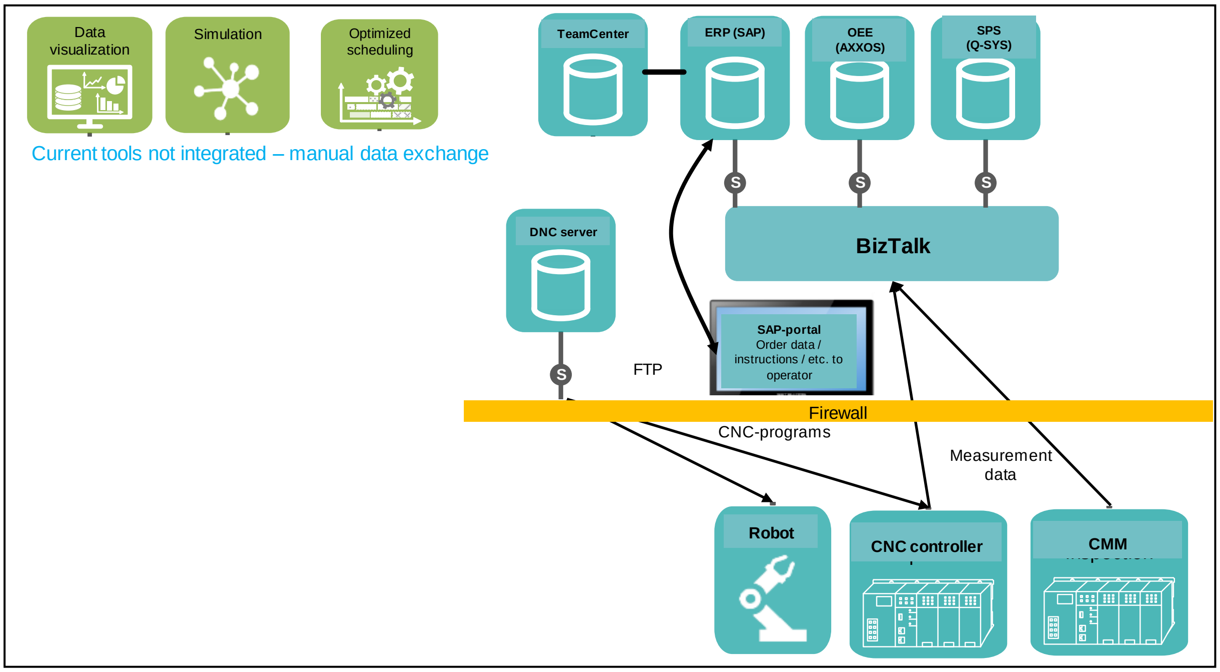


Figure 60: Illustration of the current information and data flow structure from business systems to shop floor nodes.

6.2. Vision

The long term vision and goal is to build an industrial solution for higher degree of flexibility and reconfigurability of automated or semi-automated discrete manufacturing cells. The cell system should be a platform solution that can have different configuration of production processes. The processes can work individually or in an operation sequence – the "micro-flow cell" concept defined in Deliverable 10.1. The cell scheduling/operation sequencing and control, integration and reconfiguration in a heterogeneous system is the key functions/tools ("Cell middleware"). For both long term planning and near real time planning and scheduling the business planning systems need better integration than today. Thus, the vertical integration with the business level systems to automate data/information management for the production execution and reporting results ("Factory Middleware") is essential in the long term perspective.

A holistic view of the information flow and activities for planning, development and operation is outlined if Figure 61 and explained in the following text. The long term planning and development of required solutions will be just as important as the short term perspective on operations and execution to get the best possible effect of the advantages of the PERFoRM concepts and solutions.

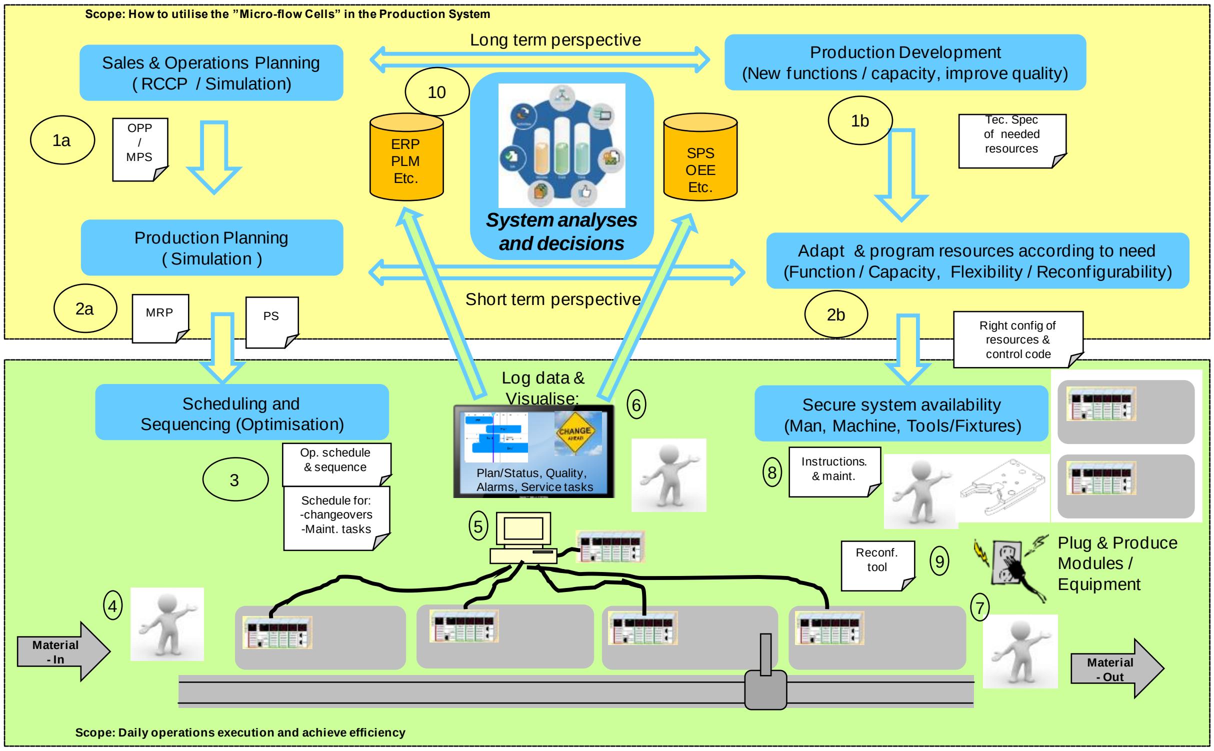


Figure 61: Outline of the future information and data flow structure from business systems to shop floor nodes.

The upper part of Figure 61, describes how the long term planning and development of production planning and the industrial structure and system solutions need to be done in an integrated way and plan for the use of the new cell concept. The long term analysis of sales/demands gives answers to the need of and different scenarios can be simulated and analyzed, which results is an "Operation & Production Plan" (OPP) or "Master Production Schedule" (MPS). With input from the Sales & Operations Planning, there may be needs to adapt the functions and/or capability or capacity of the production system and its resources. This also include the need of and advantages from having flexible and reconfigurable equipment/resources. The decision can be to acquire more resources, i.e. some long term technology development and investments, etc. as well as competence and skills of the work force. It can also be an option of using/specifying need of flexibility and re-use/reconfigure equipment or parts of the production system. The result is the technical requirements/specifications of the required production resources and plans to implement.

The lower part of Figure 61, , describes the short term perspective for production operations in which the cell concept is used. The MPS and definition of available resources is input to make a more refined and detailed plan. This can be supported by simulation or other planning tools, and it results is an MRP for material supply and Production Scheduling (PS) for the value streams. The production resources need to be prepared and set up to do the right things and can, depending on level of flexibility etc., be adapted and prepared to be used according to the Production Plan. The result is right configuration of HW equipment together with instructions, programs, and etc. To enable the “Plug and Produce”, a “Reconfiguration Tool” in the cell control system manage the changeover process.

The daily scheduling and sequencing of jobs at ”cell-level” should be as optimal as possible – i.e. using the PERFoRM tool to be developed. The result is an optimized schedule and sequence for the next shift. This plan also can create the slots and plan for when to make the changeovers, i.e. when to change from one process module to another, as well as take into account the need for maintenance and create slots for that on the schedule.

The jobs are triggered by the system/schedule and the operators load/feed required material into the cell, and start the different orders. The cell controller executes the automated activities of the production schedule, when the jobs are released by the operator, which is supported by supervision and visualisation from the system, to show status of jobs/orders, equipment, etc. and display on Monitor or HMI. The operator un-load parts from the cell, and after any required inspection is done and confirmed, the operator finalize each job to report the completion. The data made available from the production cell/system provides the source to visualisation of KPIs, to be used in different ”tools”, e.g. the PERFoRM ”tools” as well as analyses and decision support for improvements and the planning, scheduling etc.

The industrial structure need to be developed and constructed step by step over a long period of time, and following the needs for replacement of old equipment and upgrading technology and/or production capacity. The information system at the top level should be implemented together with the first applications of the automated cells, which then will facilitate and shorten the lead time for introduction of new“micro-flow” cells. As illustrated in Figure 62, that production cell platform can be expanded and reconfigured for different applications and used to design parts of the production flow.

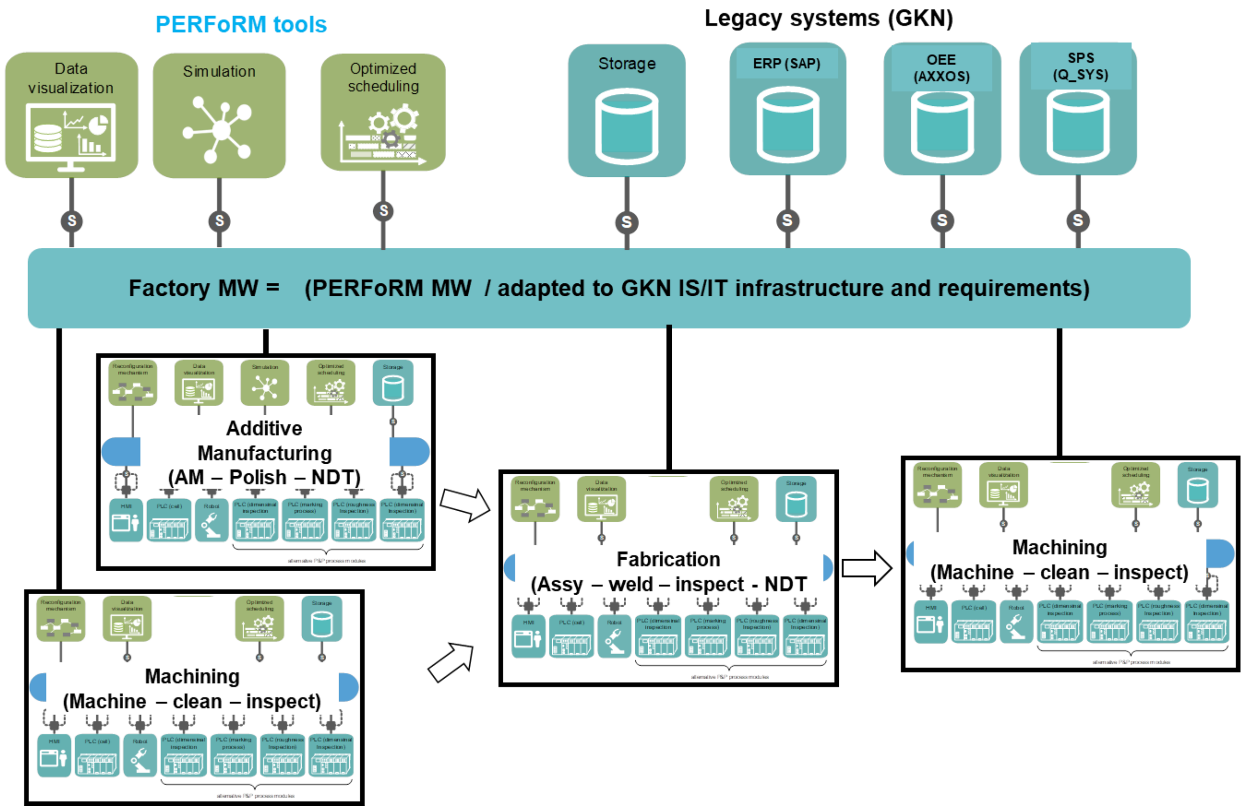


Figure 62: Illustration of a potential future modular and scalable architecture.

6.3. Use Case Implementation

6.3.1. "Work flow" for flexibility and reconfigurability

For the GKN use case the main goal is to develop and demonstrate the necessary solutions for simple and quick changeovers in the modular system, i.e. an application of the principles for Cyber Physical Production Systems. A set of production orders will also be scheduled and executed to test and verify the PERFoRM adapters and tools. (See Section 2 and table in Section 0).

To test and demonstrate flexibility and reconfigurability, four process modules are planned (deburring, marking, inspection of surfaces and dimensional measurement), and the sequence for *shut down – plug out – replace module – plug in – start-up* will be tested and evaluated. The definitions and functions specified for the "micro-flow cell" and the GKN use case is documented in Deliverable 10.2 [20].

In Figure 63, the role of humans, activities and information flow for operating the cell is is illustrated and the "work flow" summarised in the following text. In Figure 63 the "work flow" for exchanging process modules in the cell is illustrated and explained in the following text. A more extensive description is provided in Deliverable 10.2. [20].

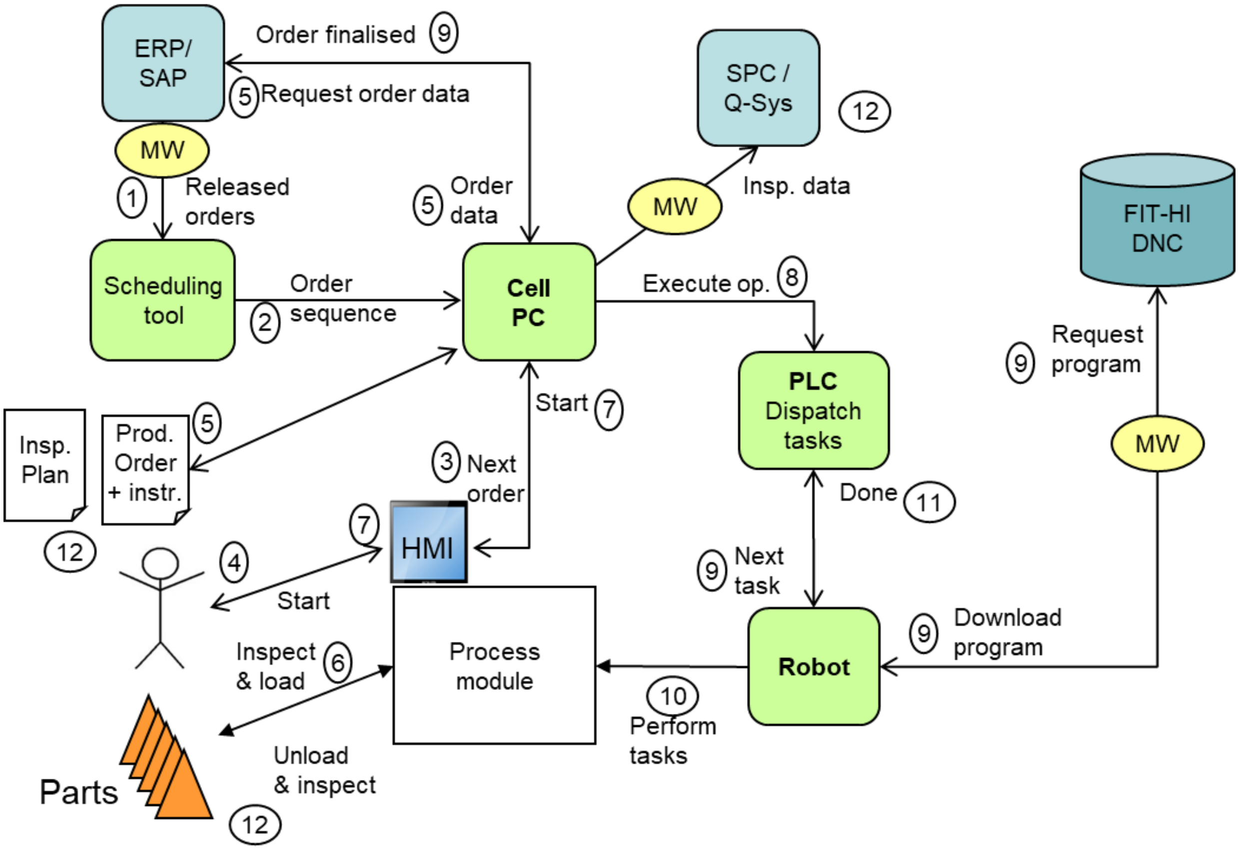


Figure 63: The "work flow" for operating the cell to plan and execute production orders.

The generic operation sequence to use any of the processes in the cell and to execute a production order follows the main steps as in this example:

- From the ERP system, available orders are scheduled (1) and an optimized sequence is generated (2), managed by the Cell PC and posted on HMI(3).
- The operator reads the available production orders and related information/instructions (4, 5) and prepare for each operation to make sure conditions are correct and all material and tools at hand.
- Operator load "part" to magazine (6) and starts the job from the HMI (7).
- The system (cell PC) checks that all requirements for work are fulfilled and send tasks to be executed to the PLC (8).
- Robot is requested to start. The part ID is read/confirmed with the order number and correct robot program is down loaded (9).
- The robot starts the operation sequence, getting the right gripper and load part in the fixture or tools etc. (10).
- When the processing is finished, the robot return the part to the magazine and the order status is set to done (11).
- Operator is informed that the order is processed (from the HMI), unload parts, check / inspect as needed and finalize the order and report in SAP (12).

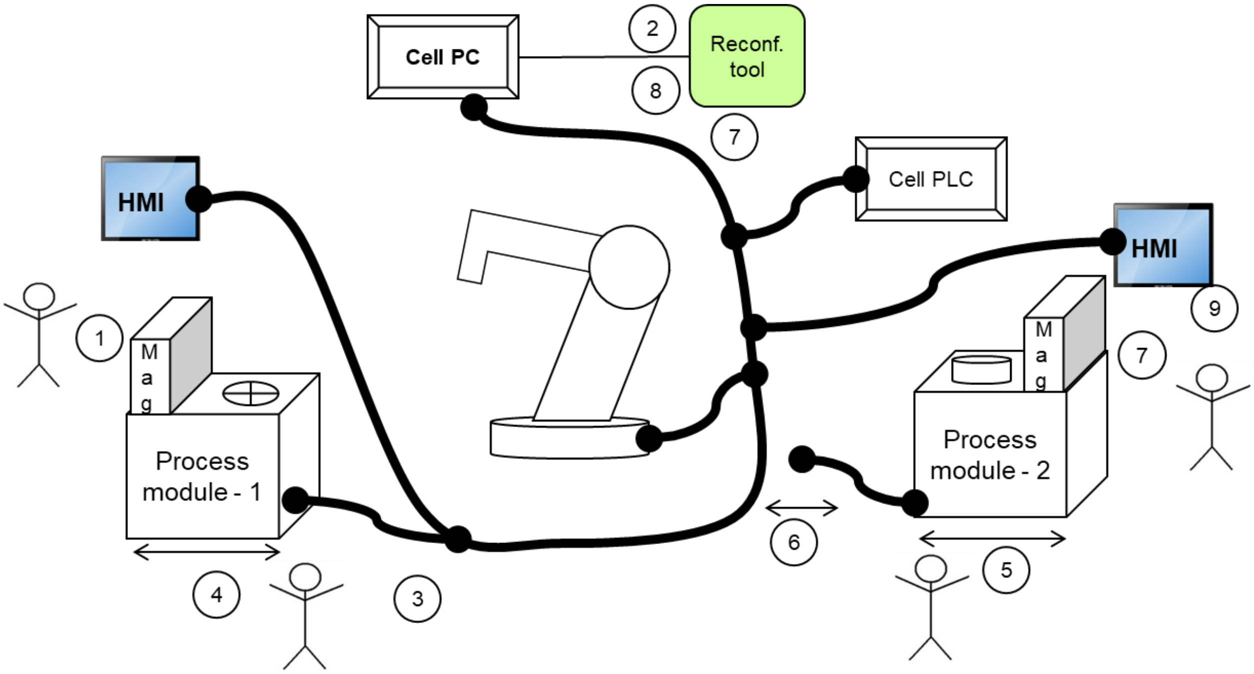


Figure 64: The “work flow” for planning and changeovers demonstrating the production flexibility and reconfiguration of the cell.

The cell system is designed and set-up to handle different process modules. A generic procedure to replace any of the modules in the cell should follow the steps as in this example:

- The operator plans for a production stop and start a sequence to prepare for change over is initiated on the HMI (1).
- The module is shut down and ”check-out” from the cell – managed by the ”Reconfiguration tool” – communication through the HMI (2).
- The process module is (physically) disconnected – signal cables, media … - and prepared to be moved (3). The process module is released and transported out of the cell (4).
- The other process module is transported into the cell, and positioned and locked, using clamping / positioning devices (5).
- The process module is prepared for production (any transport mode securing's removed) and (physically) connected – signal cables, media, etc. (6)
- The process module is turned on, and perform a self-test , to be ready to connect to the cell system (7).
- The identification of the process module is recognized by the “Reconfiguration tool”, which starts the initiation and changes the settings in the cell controller (8).
- Confirmation is received on the HMI that the reconfiguration is done and the module is ready to be used (9).

6.3.2. Implementation Architecture

To accomplish the functionality needed in the work flow as described abowe, a combination of several solutions developed throughout the work packages 2 to 4 of the PERFoRM project will be integrated in the GKN use case demonstrator cell (see Table 4).

The PERFoRM Middleware will be used to ensure the communication and integration of the solutions, which use the defined Data Model and Standard Interfaces. The connection to the legacy systems is done via a gateway or adapter (. Scheduling of work orders and changeovers in the cell, togheter with KPI monitoring will support the "Human in the loop" and "Human in the mesh" to make decisions and operate the cell to maximize utilization and throughput. (This can be tested and simulated).

Table 4: PERFoRM Solutions from WP2-WP4 applied in the GKN Use Case

Area	Solution	Section in Document
Middleware	Industrial Middleware	2.2.1
Data Exchange	Data Model and Standard Interfaces	2.2.2
Technology Adapter	Sensor Adapter (Surface measurement tool)	2.2.3.4
Scheduling	Scheduling of production orders and changeovers	2.2.7.2
Flexibility	Agent based reconfiguration tool	2.2.7.4
Decision support	KPI monitoring	2.2.8.2

For the realization of the use case, an architecture for the demonstrator cell will be developed and implemented with the functionality as illustrated in Figure 65.

The PERFoRM Middleware and tools will be hosted on dedicated hardware/Cell PC in the demonstrator cell, and the focus is om the cell functions to test and evaluate the flexibility and reconfigurability performance of "plug-and-play". The cell integration (cell Middleware) will be implemented using OPC-UA for communication between all nodes in the cell. The "Factory Middleware" i.e. connections to legacy systems, will be realized through a local database or a "sand box" system. (A full implementation and integration to factory systems is not realistic at this stage due to cost and risk). Some data flows to/from the legacy systems and the demonstrator cell may be done manually.

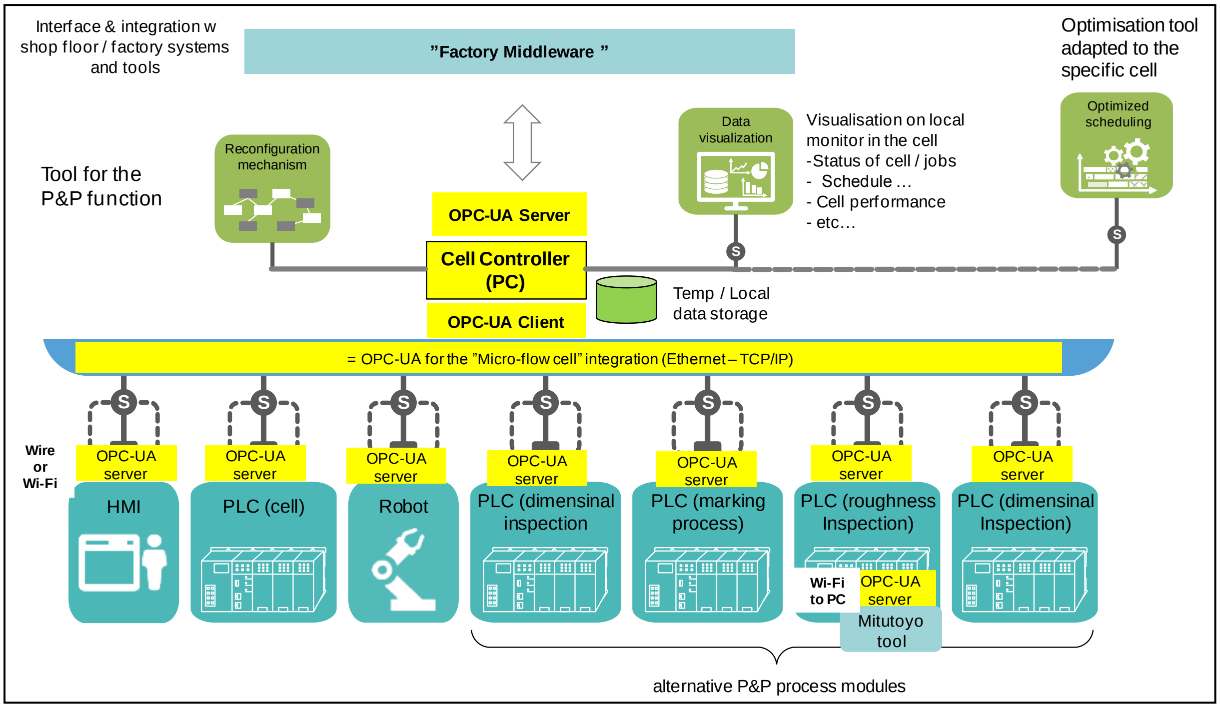


Figure 65: Illustration of the architecture to be implemented in the use case demonstration.

The physical cell concept and its process modules (colored boxes) is illustrated in Figure h. The process modules will have its own PLC/Controller and their current position in the cell is managed by the reconfiguration tool. The operator can access three of the module station areas, to load/un-load parts, replace tools etc. Eight of the modules can be moved/replaced while the cell is still operating. This is accomplished by a sophisticated safety system to prevent accidents.

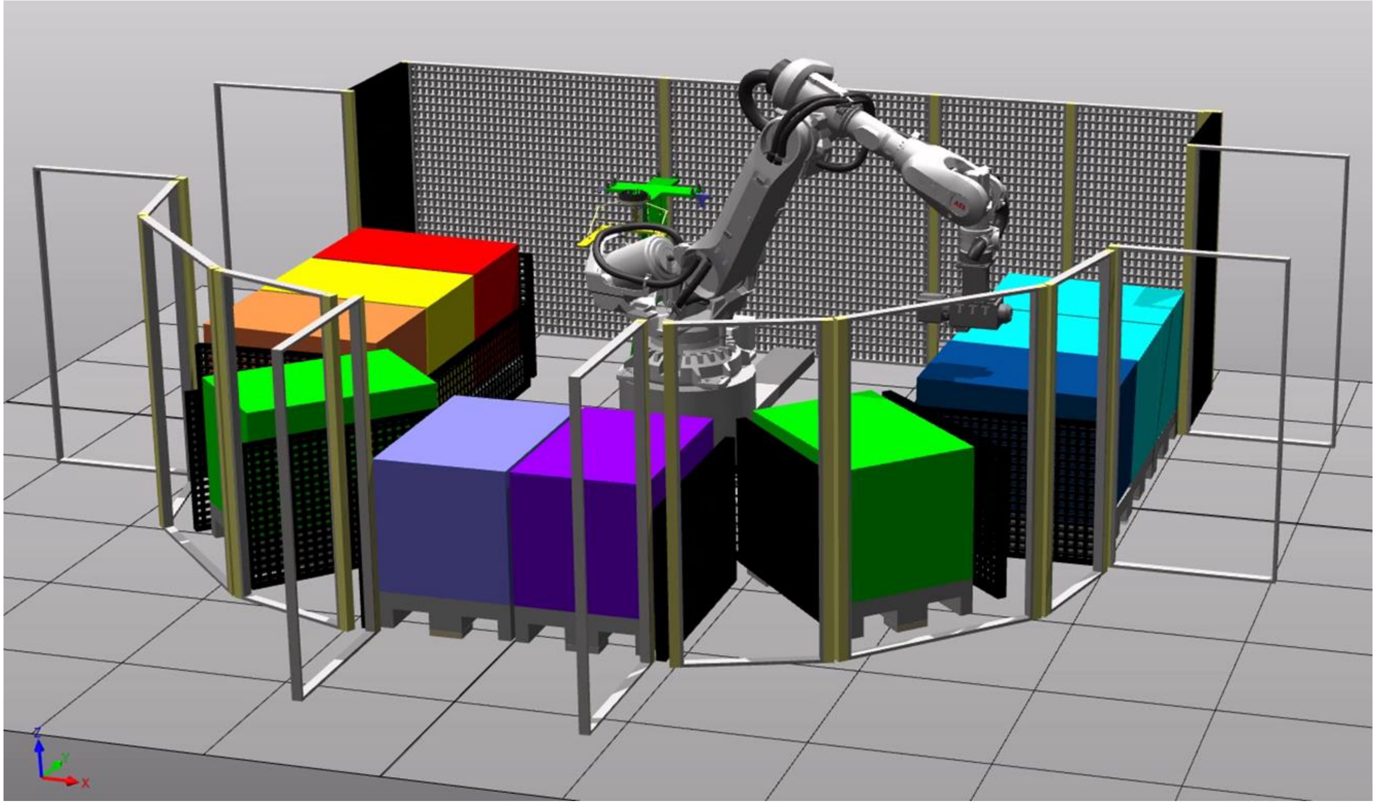


Figure 66: Illustration of the demonstrator cell with movable process modules.

6.3.3. Required Testing

The use case will be based on a number of different hardware and software solutions. Some are designed and implemented directly into the cell in which the use case will be demonstrated. Others will be developed and tested as separate functions, the results from WP 2, 3, and 4, at partner facilities, and as subsystem solutions in WP6 at SmartFactory, and The MTC, before they are implemented for the use case system. The required testing of main functions and tools for the use case are listed below:

- The hardware solutions will be designed, built and tested as part of the build and commissioning of the demonstrator cell at the research and test facility GKN is using. That will include the solutions for the modular process units and the docking system for quick changeovers, with appropriate geometry assurance management.
- Safety system solutions include both hardware, software for monitoring and logics to manage the flexibility and reconfigurability of the system, while it is up and running. (It is a relatively advanced and sophisticated system to minimize downtime without compromising safety). This will be tested and verified at GKN test facility.
- The complete cell system architecture and process (workflow) for the flexibility and reconfigurability, as described in Section 6.3.2 will be built, tested and demonstrated at the end of the project.

Tests of individual functions and tools have been done in finalized tasks and/or are planned as part of WP6 activities, See Table 5, Table 6, Table 7. For more details about the test scenarios, see Deliverable 6.1, 6.2 and 6,3 for the definition of test plans. The results from the test done so far are reported in Deliverable 6.4 (Deliverable 6.5 and 6.6 is in progress [21].).

Table 5: GKN - Test scenario description (Deliverable 6.1) for the self-adaptive machine demonstrator.

Test Scenario ID	Test Scenario	Priority
TS-GKN-F-1	Robot receives commands and performs correct operations	High
TS-GKN-F-1.1	Robot is able to receive instructions from middleware	High
TS-GKN-F-1.2	Robot performs operations according to instructions received	High
TS-GKN-F-2	Cell HMI displays correct information on screen and captures input from user correctly	low
TS-GKN-F-2.1	HMI displays correct information on screen	low
TS-GKN-F-2.2	HMI captures input from user and transmits correctly to middleware	low
TS-GKN-F-3	Cell transmits safety information to the middleware (through the PLC)	High
TS-GKN-F-3.1	Cell security sensors are tested one at a time to check that correct information is transmitted to middleware	High
TS-GKN-F-4	PLC of each process is able to communicate with the middleware	High
TS-GKN-F-5	The devices in each process are able to communicate with the process PLC via wired or wireless adaptor	High

Table 6: GKN - Test scenario description (Deliverable 6.2) for the self-adaptive flexible and modular cell.

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

Table 7: GKN - Test scenario description (Deliverable 6.3) for self-adaptive large system.

Test Scenario ID	Test Scenario	Priority
TS-GKN-F-1	Verify that the program from HI-FIT will be able to be transferred to the robot through OPC-UA.	High
TS-GKN-F-2	Verify that the reconfiguration tool is able to detect which process has been plugged in to the right side and which to the left.	High
TS-GKN-F-2.1	The reconfiguration tool can detect a new module/ process, after this will be plugged in.	High
TS-GKN-F-2.2	New module receives the confirmation from reconfiguration tool and start	High

	sending data to the OPC-UA.	
TS-GKN-F-3	The reconfiguration tool, after detecting the processes, can send the data to the scheduling system.	Medium
TS-GKN-F-3.1	The scheduling system receives the data and uses them to optimise the production schedule.	Medium
TS-GKN-F-3.2	The updated schedule will be sent back to the control execution through the OPC-UA.	Medium
TS-GKN-F-4	The specifications of the plugged “Process modules” are available to the HMI.	High
TS-GKN-F-4.1	The current status of the cell is tracked and visualised by software tools.	High
TS-GKN-F-4.2	During the communication with visualisation software, unexpected disconnection happens.	Low

7. Harmonized PERFoRM approach - Gaps, Risks, Obstacles

The PERFoRM project aims at the conceptual transformation of traditional production systems towards flexible and reconfigurable production systems. This deliverable collects all the technology solutions developed during the project to support this transformation for the four industrial Use Cases and shows how these technologies can be harmonized together in a unique solution architecture depending on the specific needs of each use case. Even if the technology harmonization enables the effective cooperative work between new and legacy software applications and hardware systems, some other aspects need to taken into account. The deployment in industry of the PERFoRM industrial middleware, standard interfaces, and tools, is also linked to some risks and obstacles that can keep manufacturers from migrating to the next generation industrial automation systems.

The relevant risks and obstacles regarding the industrial adoption of the PERFoRM solution have been derived from the Use Cases through questionnaires [22] and workshops. In particular, the identified risks consider performance, implementation, maintenance, and organizational aspects.

First of all, industries require very high reliability and availability of the new architecture, in order to avoid production and delivery delays lying on the risk to stop the production during its operation. A key aspect from the technological point of view is the lack of maturity of the selected technologies, i.e. adaptors and plug-and-play functions, and especially the Middleware, in terms of stability, latency and response time, also for real-time applications. A possible solution to avoid production risks is to further test and verify the new technologies in the real environment by allowing the PERFoRM Middleware and tools to gather production data only from an intermediary database that collects data coming from the shop floor through legacy databases.

Another important issue is related to the extendibility of the middleware to new equipment and, consequently, the scalability of the solutions in order to support the management of large-scale systems. This aspect is strictly related to the compatibility of new technologies with legacy systems and the risk that data adapters and possible gateway to factory and business systems could not work properly. In addition, internal security policies need to be taken into account to meet special requirements related to the integration of new applications with legacy systems. Indeed, it has been analyzed in one of the use cases how, for example, it is possible to connect different legacy complex modules, having their own legacy middlewares, with the PERFoRM middleware by using standard interfaces and technology adaptors between legacy and PERFoRM middlewares.

Moreover, the lack of support from middleware and technology providers on the market, as well as the skills available, has been pointed out as a possible obstacle for the continuity of use and user satisfaction. This aspect is also linked to the required good usability of system in terms of good user interfaces and low complexity of information and evaluable results. Also, required licenses for new software and systems can be an obstacle from an economical perspective.

The human acceptance of the new system solution at global level from IT organization has also a big importance in the successfulness of CPS industrial adoption. For this reason, new resources as well as new skills and competences are mandatory to use and maintain CPS systems and the automation architecture.

Especially, new roles need to be considered in the organization structure, e.g., system of systems expert able to keep an holistic system view and new figures that have the responsibility to fix and maintain the new distributed architecture. For this reason, adequate trainings and instructions need to be provided to the operators to cope with the new implemented systems.

In conclusion, a smooth migration strategy from existing traditional centralized systems towards flexible and reconfigurable production systems is necessary to mitigate these risks and support the industrial adoption of the Industry 4.0 paradigm. To this end, the outcomes of this deliverable are used in Task 5.2 and 5.3 as a starting point to develop an approach that supports the migration of the Use Cases from their "As-Is" situation to their long-term vision by deploying the defined PERFoRM architecture and technology solutions.

References

[1] Deliverable 2.4, “Industrial Manufacturing Middleware: Specification, prototype implementation and validation”, PERFoRM project, 2017.

[2] Angione G., Barbosa J., Gosewehr F., Leitão P, Massa D., Matos J, Peres R.S., Rocha A.D., Wermann J., “Integration and Deployment of a Distributed and Pluggable Industrial Architecture for the PERFoRM Project”, in 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy.

[3] Chakravorti N., Dimanidou E., Angione G., Wermann J., Gosewehr F., “Validation of PERFoRM reference architecture demonstrating an automatic robot reconfiguration application”, in 15th International Conference on Industrial Informatics, INDIN2017, 24-26 July 2017, Emden, Germany.

[4] Liberopoulos, G. (2013, May). Production Release Control and the Push/Pull and Make-to-Order/Make-to-Stock Distinctions. In H. Tempelmeier, H. Kuhn and K. Furmans, eds., Proceedings of the 9 th conference on Stochastic Models of Manufacturing and Service Operations, Kloster-Seeon, Germany (pp. 113-120).

[5] Nance, R. E. (1996). A history of discrete event simulation programming languages (pp. 369-427). ACM

[6] Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. The knowledge engineering review, 10(02), 115-152.

[7] F. Boschi, G. Tavola, M. Taisch, “A description and analysis method for reconfigurable production systems based on Finite State Automaton environments,” in Service Orientation in Holonic and Multi-Agent Manufacturing, 2016, pp. 349–358.

[8] Simulink, Matlab, and M. A. Natick. "The mathworks." (1993).

[9] W. Shen and D. H. Norrie, “Agent-based systems for intelligent manufacturing: A state-of-the-art survey”, Knowledge and Information Systems, vol. 1, no. 2, pp. 129– 156, 1999.

[10] M. J. Wooldridge, An introduction to multiagent systems, Reprint. Chichester: Wiley, 2008.

[11] M. Schönemann et al., “Simulation of matrix-structured manufacturing systems”, Journal of Manufacturing Systems, no. 37, pp. 104–112, 2015

[12] P.Vrba, “Simulation in agent-based control systems: Mast case study”, International Journal of Manufacturing Technology and Management, vol. 8, no. 1/2/3, p. 175, 2006

[13] Büth et al., “Introducing agent-based simulation of manufacturing systems to industrial discrete-event simulation tools”, IEEE 15th International Conference on Industrial Informatics (INDIN). 2017.

[14] Anna De Carolis, Marco Taisch, Giacomo Tavola, “sCorPiuS- D3.3 Final Roadmap-Future trends and Research Priorities for CPS in Manufacturing ” 2017.

[15] G.Tavola,F.Boschi,M.Taisch, “From key business factors to KPIs within a reconfigurable and flexible Cyber-Physical System,” IEEE-ICE, 2017.

[16] Lee, B. Bagheri, and J. Chao, “Introduction to cyber manufacturing,” Manufacturing Letters 8, pp. 11–15, 2016; National Institute of Standards and Technology (NIST), Smart Manufacturing and Construction Control Systems, EL Program Report, 2016.

[17] Deliverable 7.1, “Siemens description and requirements of architectures for retrofitting production equipment”, PERFoRM project, 2016.

[18] Deliverable D9.1, “Whirlpool description of Requirements and Architecture Design. PERFoRM project, 2016.

[19] Deliverable D10.1, “GKN Use Case goals, requirements and KPIs - Specification of applications, functions and requirements for the “Micro-Flow cell”, PERFoRM project, 2016.

[20] Deliverable D10.2, “GKN use case Concept Development of the “Micro-Flow Cell””, PERFoRM project, 2017.

[21] Deliverable D6.1, “Report on Self-Adaprive Machines Demonstrator Desing and Set-up”, PERFoRM project, 2016.

[22] Deliverable D5.2, “The PERFoRM Migration Strategy for A Generic Migration Scenario and for Additional Show Cases within the Testbeds in WP6 1st Release”, PERFoRM project, 2017.