Production harmonizEd Reconfiguration of Flexible Robots and Machinery

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

Deliverable 6.1

Report on Self-Adaptive Machines Demonstrator Design and Set-up

Lead Author: MTC

Version: 1.0
Date: 27/09/2016
Status: Final
Dissemination level: PUBLIC

Version history

Version	Date	notes and comments
0.1	25/04/2016	Deliverable structure, Table of Contents with effort allocation
0.2	28/05/2016	Data Flow diagrams for all use cases
0.3	20/07/2016	Revised Data Flow diagrams based on suggestions from use cases Contribution to Chapter 5.2 (Siemens) Contribution to Chapter 5.3 (IFEVS) Contribution to Chapter 5.4 (Whirlpool) Contribution to Chapter 5.5 (GKN)
0.4	25/07/2016	Consolidation of document with efforts from partners Contribution to Chapter 3, 4.4 (Polimi) Contribution to Chapter 3.1, 4.6 (Siemens) Contribution to Chapter 3.2 (IFEVS) Contribution to Chapter 3.3 (Whirlpool) Contribution to Chapter 3.4 (GKN) Contribution to Chapter 4.1 (HSEL) Contribution to Chapter 4.2 (UNINOVA) Contribution to Chapter 4.3 (Loccioni) Contribution to Chapter 4.5 (TUBS)
0.5	08/08/2016	Test scenarios for use cases Contribution to Chapter 6 (SmartFactory, IFEVS, Whirlpool, GKN)
0.6	09/08/2016	Version sent out for feedback
0.7	01/09/2016	Revised version, based on partners feedback
0.8	01/09/2016	Version sent for internal review
0.9	16/09/2016	Revised version, sent for second internal review Contribution to Chapter 3 (Polimi) Contribution to Chapter 4.1 (HSEL) Contribution to Chapter 4.2 (Siemens) Contribution to chapter 4.5 (TUBS) Contribution to chapter 5.2 (Siemens) Contribution to Chapter 5.3 (IFEVS)
1.0	27/09/2016	Final version

Author List:

Nandini Chakravorti (MTC)
Soghra M. Gargari (MTC)
Evangelia Dimanidou (MTC)
Sergio Davies (MTC)
Andre Hennecke (SmartFactory)
Filippo Boschi (Polimi)
Jeffrey Wermann (HSEL)
Ricardo Peres (Uninova)
Giacomo Angione (Loccioni)
Lennart Buth, Benjamin Neef (TUBS)
Matthias Foehr, Birgit Obst (Siemens)
Johan Vallhagen (GKN)
Pierluigi Petrali (WHR)
Gregorio Iuzzolino (IEFVS)

Abstract

WP6 "Validation and demonstration of reconfigurable and self-adaptive systems" is responsible for the verification of the technologies developed within the PERFoRM project prior to deploying them to the industrial environments. Task 6.1: "S*elf-adaptive and reconfigurable machines and robots*" in particular is responsible for the validation of the PERFoRM concept for Self-adaptive and reconfigurable machines and robots. The aim of this deliverable is to identify the test scenarios for the different use cases. An initial assessment of all test beds (MTC and SmartFactory) and pre-test beds (e.g. Loccioni, TUBS and IPB) has been detailed to provide the reader a flavour of the diverse production resource available for testing the concepts developed as a part of this project. This document also provides details of the architecture of the use cases.

1 Index

2 List of Figures

3 List of Tables

1. Introduction

The current manufacturing domain is under enormous demand to have the ability to produce more customised, cheaper and higher quality products. These demands have motivated the introduction of new paradigms such as modularity and re-configurability into the manufacturing domain. The objective of this project is to introduce a next generation of agile manufacturing systems that are dynamically reconfigurable to enable self-organisation and adaptation along the system life-cycle. These systems are targeted to be based on modular plug-and-produce components within the manufacturing system life cycle. The overall aim of the PERFoRM project is to develop a common reference architecture for Agile Manufacturing Control system for true plug-and-produce devices, robots and machines.

The structure of the PERFoRM project and the aim of each work package can be seen **Fehler! Ungültiger Eigenverweis auf Textmarke.**.Work package (WP) 6: "*Validation and demonstration of reconfigurable and self-adaptive systems*" is responsible for: (1) verification of the technologies developed, (2) validation of user requirements compliance and (3) de-risking the technology solutions developed within this project prior to deploying them in the industrial environments. The Technology Readiness Level (TRL) and the Manufacturing Readiness Level (MRL), used to estimate the maturity of technologies, are categorised into: (1) the research technologies (levels 1-3), (2) technologies under development (levels 4-6) and (3) technologies ready to be used in industrial environments (levels 7-9). This project aims to develop technologies in the TRL range 6-7, as they will be demonstrated in operational environments (H2020, 2014). The technology validation process will be conducted in an incremental manner and the validation effort has been decomposed in three different tasks: (1) Task 6.1: "S*elf-adaptive and reconfigurable machines and robots*", (2) Task 6.2: "*Self-adaptive and reconfigurable production modules*" and (3) Task 6.3: "*Self-adaptive and Reconfigurable Large Scale Systems*". Task 6.1 in particular is responsible for demonstrating and de-risking the technology solutions provided within the consortium via the use of standard off-the-shelf machine tools and robots present in the current infrastructure offered by MTC and SmartFactory (and also other test beds present at partner sites).

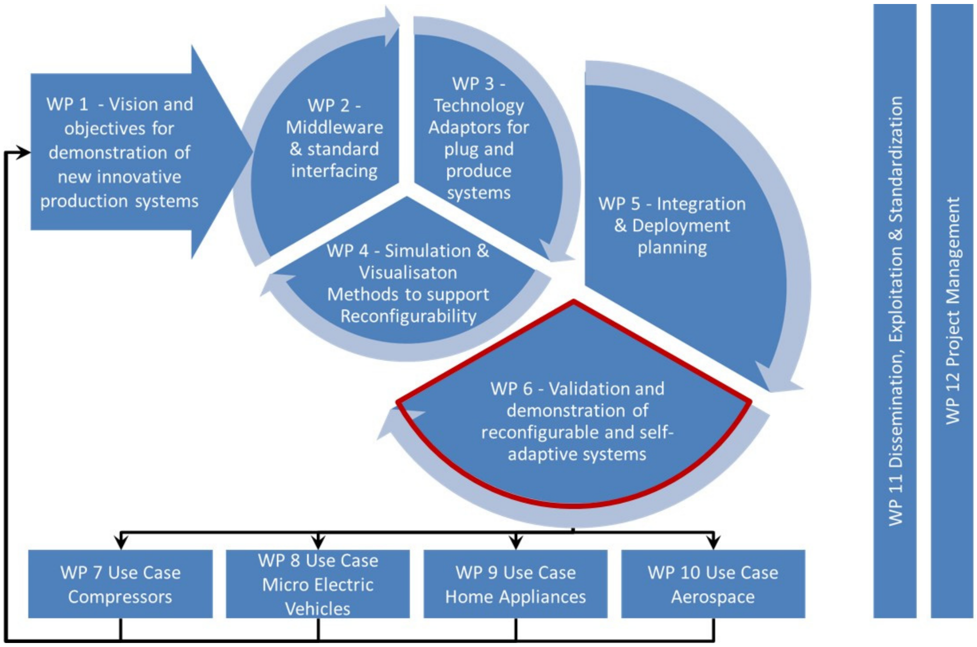

WP 1 - Vision and objectives for demonstration of new innovative production systems
WP 2 - Middleware & standard interfacing
WP 3 - Technology Adaptors for plug and produce systems
WP 4 - Simulation & Visualisaton Methods to support Reconfigurability
WP 5 - Integration & Deployment planning
WP 6 - Validation and demonstration of reconfigurable and self-adaptive systems
WP 7 Use Case Compressors
WP 8 Use Case Micro Electric Vehicles
WP 9 Use Case Home Appliances
WP 10 Use Case Aerospace
WP 11 Dissemination, Exploitation & Standardization
WP 12 Project Management

Figure 1. PERFoRM Project structure

The scope of the three tasks in WP6 (Task 6.1, Task 6.2 and Task 6.3) can be explained using the PERFoRM project framework (see Figure 2). This framework has been proposed in Deliverable 1.1: "*Report on decentralised control & Distributed Manufacturing Operation Systems for Flexible and Reconfigurable production environments*" (Siemens, 2016) and consists of three different views: *asset view*, *architectural view* and the *process view*. An explanation of the three views can be seen below:

- The *asset view* involves the physical equipment of a flexible manufacturing systems, including the human resources along with the workstations. The assets considered within this framework include: casting and molding, forming, machining, joining, additive manufacturing, inspections, robots, material handling, storage, humans, coating and laminating and changing of material properties (see Figure 2). Task 6.1 is concerned with the validation of the PERFoRM concept for self-adaptive and reconfigurable machines and robots, consequently the scope of this task falls within the realm of the *asset view* within the PERFoRM project framework.
- The *architectural view* represents the software aspects of a flexible manufacturing system such as the information and communication technologies (ICT). Several IT systems such as PLC (Programmable Logic Controller), SCADA (System Control and Data Acquisition) and DCS (Distributed Control Systems) systems are utilised for monitoring and controlling industrial applications within the lower levels of the automation pyramid. Whilst the higher levels of the automation pyramid involve applications such as Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP), the PERFoRM architecture promotes the interoperability of diverse production resources via the use of open communication and standardised data exchange models. The integration of legacy resources is allowed via the use of dedicated technology adaptors thus enabling the interface between the manufacturing service bus (middleware) and the proprietary control systems of the devices. The addition and removal of resources are managed via the middleware. Task 6.2 is involved with the deployment of self-adaptive and reconfigurable production modules and consequently the scope of this task falls within the realm of the *architectural view* within the PERFoRM project framework.
- The *process view* considers all typical processes within the shop floor and the office space, as seen in Figure 2. The inbound logistic is responsible for delivering the raw material to the manufacturing machinery at the factory. After the assembly workstation finishes the job, the production parts are inspected. Once inspection has been completed, either the painting or finishing workstation is used to finish the part or repairs are conducted at the repair and maintenance process area. The last step prior to sending the parts to the packaging station is testing. Task 6.3 involves the validation of the self-adaptive and reconfigurable characteristics within the relevant industrial environment, thus involving typical processes in the production floor (e.g. assembly, painting and finishing in the case of the Whirlpool use case). Consequently the scope of this task falls within the realm of the *process view* within the PERFoRM project framework.

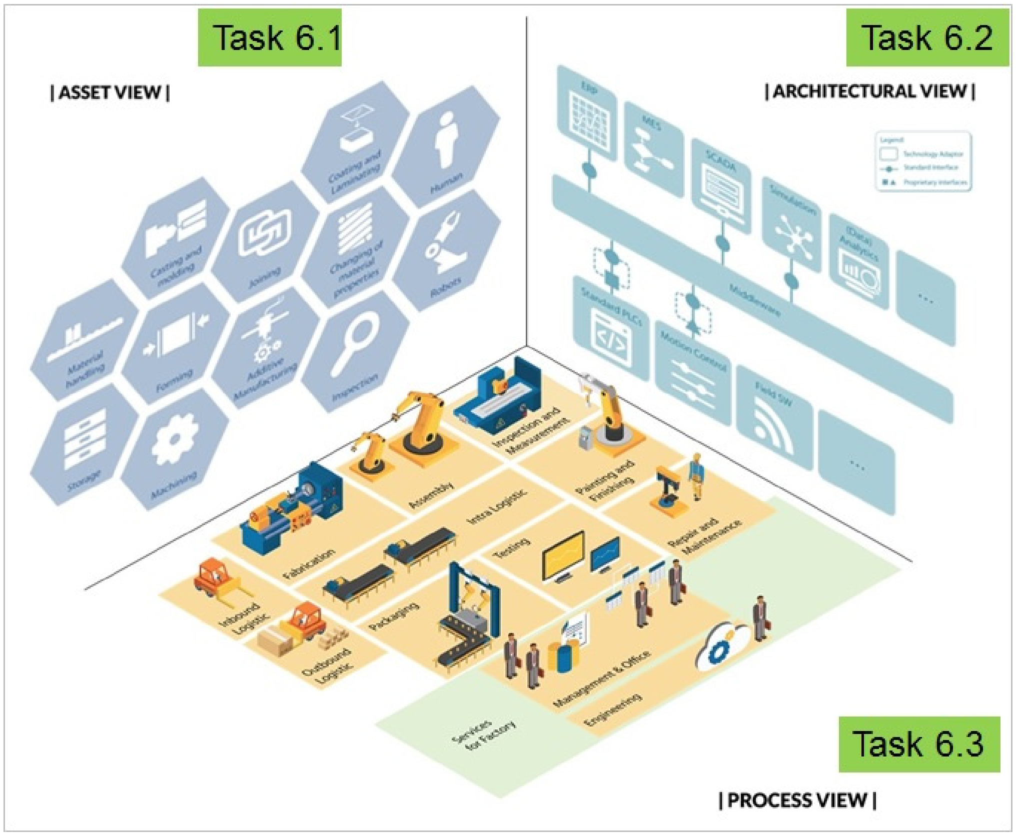


Figure 2. Scope of WP6 Tasks

1.1. Objective of the document

The objective of Task 6.1:"*Validation of the PERFoRM concept for Self-adaptive and reconfigurable machines and robots*" is to demonstrate and de-risk the technology solutions provided within the consortium via the use of standard off-the-shelf machine tools and robots present in the current infrastructure offered by MTC and SmartFactory (and also other test beds present at partner sites). This task consists of two deliverables: (1) D6.1: "*Report on Self-Adaptive Machines Demonstrator Design and Set-up*" and (2) D6.4: "*Self-Adaptive Machines Demonstrator Documentation and Results*".

The scope of the two deliverables can be seen in Figure 3. This deliverable (D6.1 in Figure 3) will present: (1) detail information about the flow of information within the to-be architecture of the use cases and (2) test scenarios suitable for Task 6.1. The next deliverable of Task 6.1 (D6.4: "*Self-Adaptive Machines Demonstrator Documentation and Results*") will present the details of the demonstrator, test cases and the results observed (as seen in Figure 3).

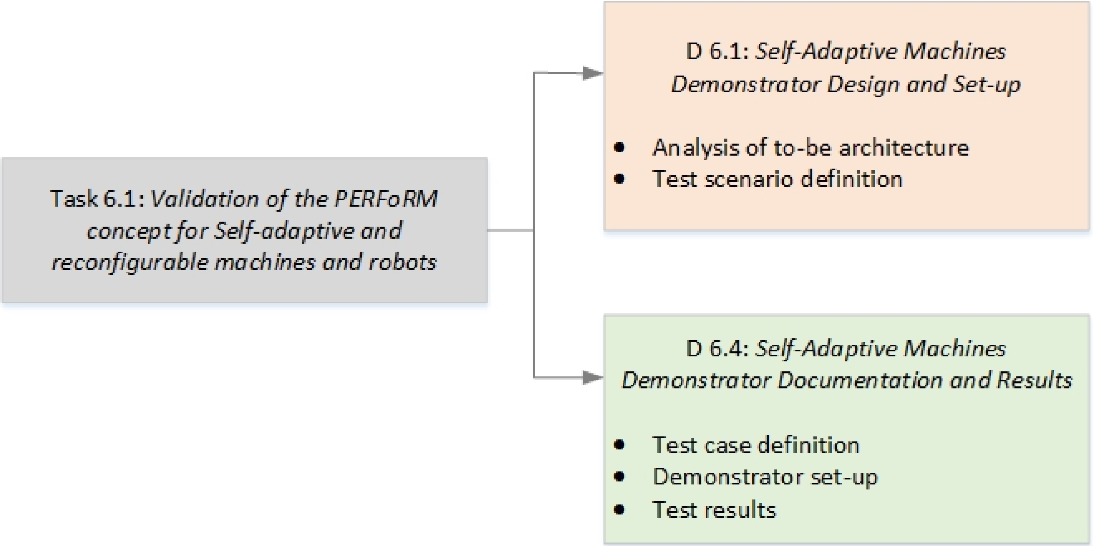

Task 6.1: *Validation of the PERFoRM concept for Self-adaptive and reconfigurable machines and robots*
D 6.1: *Self-Adaptive Machines Demonstrator Design and Set-up*
Analysis of to-be architecture
Test scenario definition
D 6.4: *Self-Adaptive Machines Demonstrator Documentation and Results*
Test case definition
Demonstrator set-up
Test results

Figure 3. Scope of Task 6.1 deliverables

The relationship of the Task 6.1 with the other Tasks within the PERFoRM project can be seen in

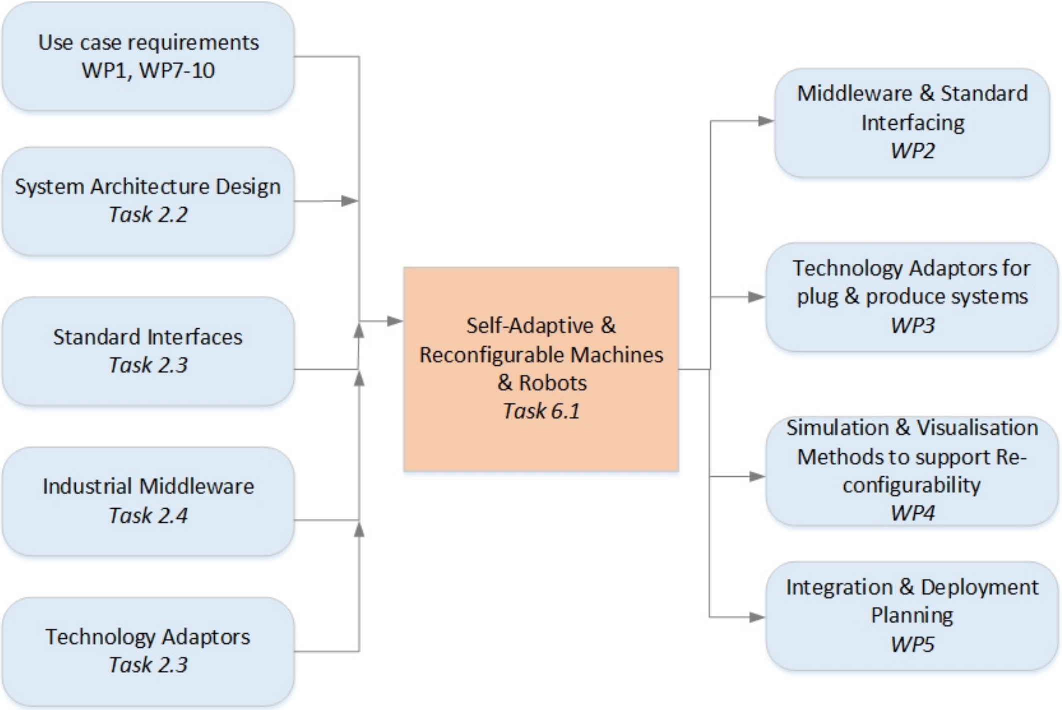
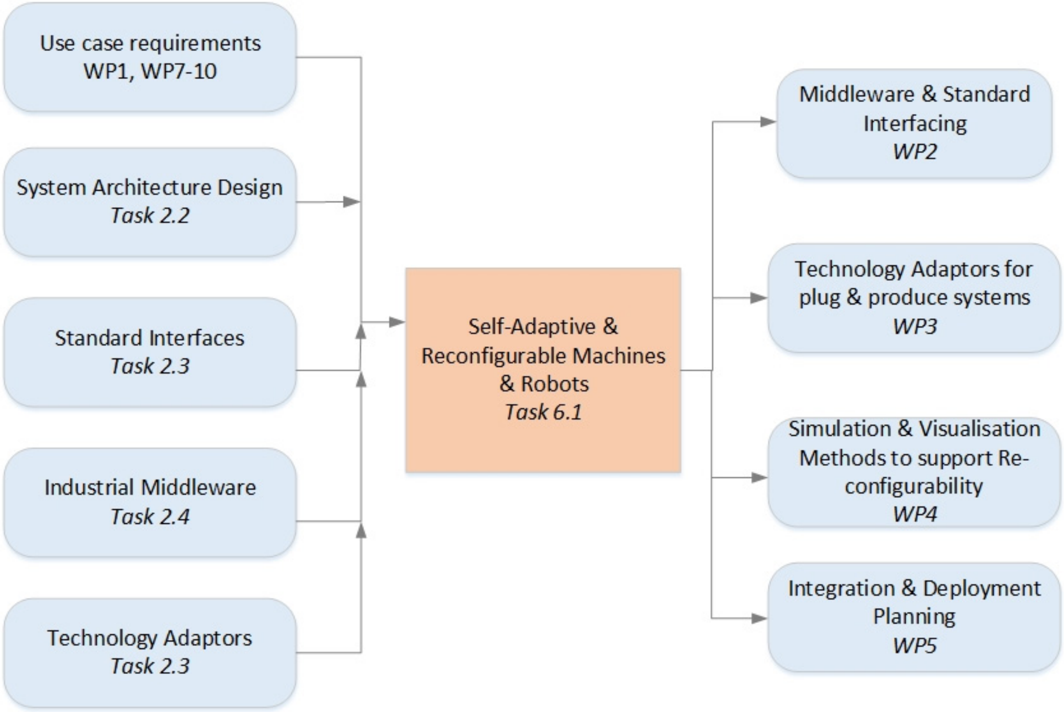
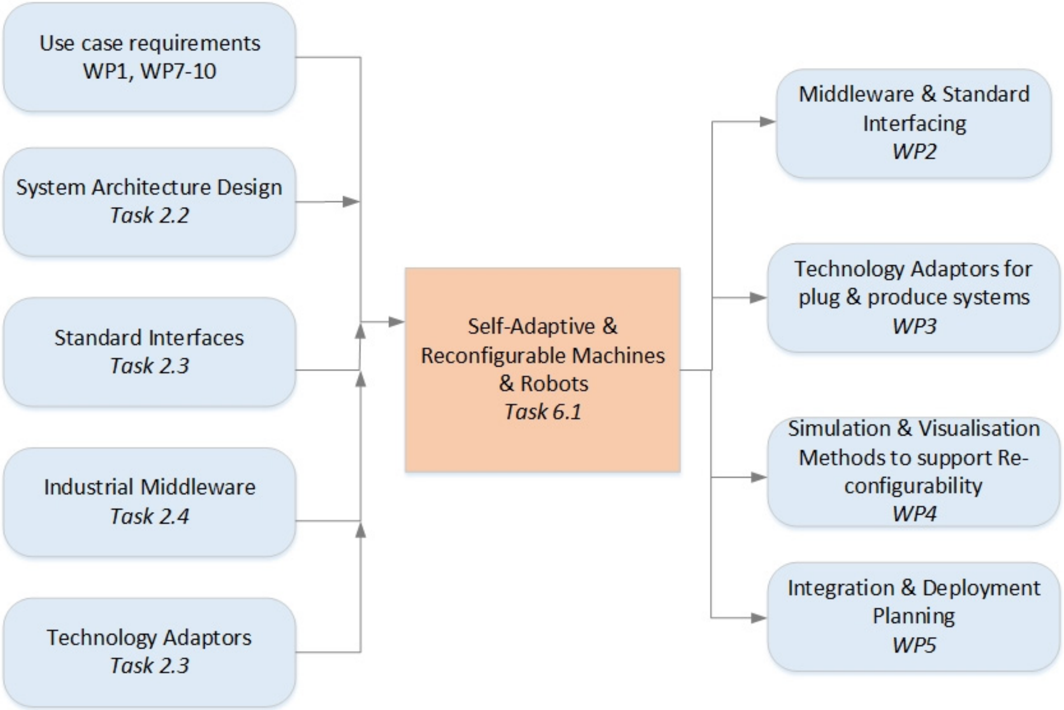

Use case requirements
WP1, WP7-10
System Architecture Design
Task 2.2
Standard Interfaces
Task 2.3
Industrial Middleware
Task 2.4
Technology Adaptors
Task 2.3
Self-Adaptive & Reconfigurable Machines & Robots
Task 6.1
Middleware & Standard Interfacing
WP2
Technology Adaptors for plug & produce systems
WP3
Simulation & Visualisation Methods to support Re-configurability
WP4
Integration & Deployment Planning
WP5

Figure 4. The outcomes from different deliverables (from other partners) used as inputs to Task 6.1 can be seen at the left-hand side of Figure 4 and are described below:


Use case requirements
WP1, WP7-10
System Architecture Design
Task 2.2
Standard Interfaces
Task 2.3
Industrial Middleware
Task 2.4
Technology Adaptors
Task 2.3
Self-Adaptive & Reconfigurable Machines & Robots
Task 6.1
Middleware & Standard Interfacing
WP2
Technology Adaptors for plug & produce systems
WP3
Simulation & Visualisation Methods to support Re-configurability
WP4
Integration & Deployment Planning
WP5

- The requirements of the four use cases have been collated by WP1 and presented in deliverables D1.2 "*Requirements for Innovative Production System Functional requirement analysis and definition of strategic objectives and KPIs*" (POLIMI, 2016) and D1.3 "*Requirements review, evaluation and selection of best available Technologies and Tools*" (FhG-IPA, 2016). The description and the requirements of each of the use cases have been presented in D7.1: "*Siemens description and requirements of architectures for retrofitting production equipment*" (Siemens, 2016), D8.1: "*Micro-Electric Vehicles description and requirements of architectures in view of flexible manufacturing*" (IFEVS, 2016), D9.1: "*Description of requirements and Architecture Design*" (Whirlpool, 2016) and D10.1: "*Use Case goals/KPIs and requirements defined*" (GKN, 2016) for the Siemens, IEFVS, Whirlpool and GKN use cases respectively.
- The system architecture design has been presented in deliverable D2.2: "*Definition of the System architecture*" (IPB, 2016).
- The standard interfaces and the industrial middleware are being designed within Task 2.3: "*Design of standard interfaces for machinery, control systems and data backbone*" and Task

2.4: "*Industrial manufacturing middleware*" as seen in


Use case requirements
WP1, WP7-10
System Architecture Design
Task 2.2
Standard Interfaces
Task 2.3
Industrial Middleware
Task 2.4
Technology Adaptors
Task 2.3
Self-Adaptive & Reconfigurable Machines & Robots
Task 6.1
Middleware & Standard Interfacing
WP2
Technology Adaptors for plug & produce systems
WP3
Simulation & Visualisation Methods to support Re-configurability
WP4
Integration & Deployment Planning
WP5

- Figure 4. The deliverables for Task 2.3 and Task 2.4 are due in M16 (January 2017) and M18 (March 2017) respectively.
- The technology adaptors are being implemented within Task 3.1: "*Open and compliant robot/resource adaptors*" and corresponding deliverable is due in M18 (Match 2017).

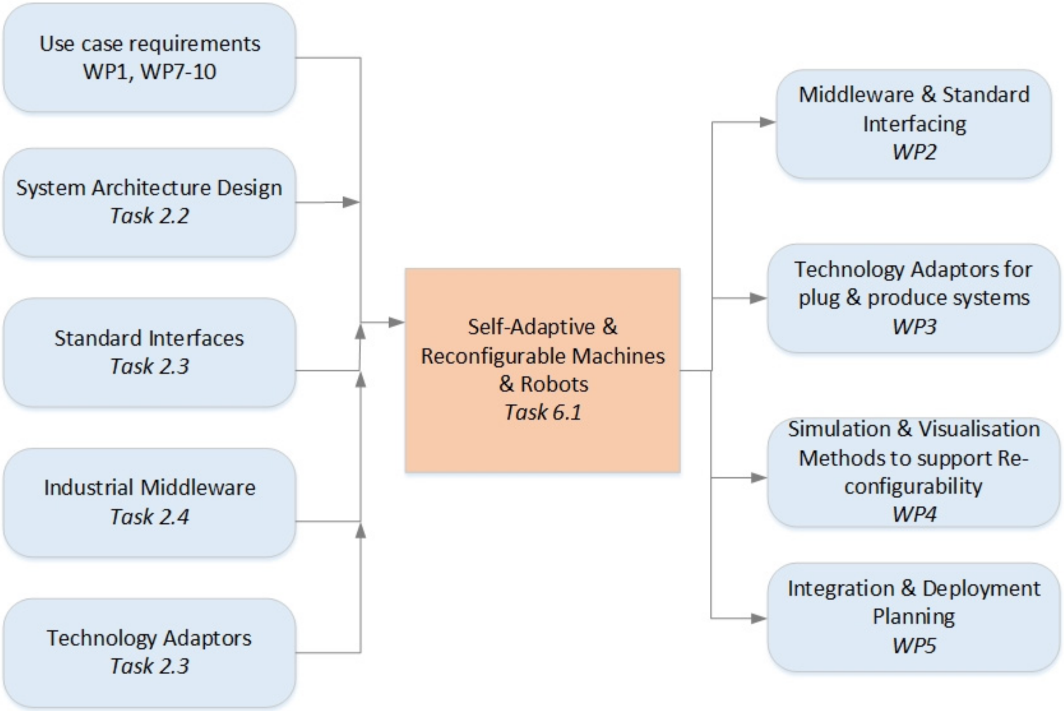

Use case requirements
WP1, WP7-10
System Architecture Design
Task 2.2
Standard Interfaces
Task 2.3
Industrial Middleware
Task 2.4
Technology Adaptors
Task 2.3
Self-Adaptive & Reconfigurable Machines & Robots
Task 6.1
Middleware & Standard Interfacing
WP2
Technology Adaptors for plug & produce systems
WP3
Simulation & Visualisation Methods to support Re-configurability
WP4
Integration & Deployment Planning
WP5

Figure 4. Interaction of Task 6.1 with other work packages

The outcomes of Task 6.1 will be utilised in different work packages such as WP2: "*Middleware and Interfacing"*, WP3: "*Technology Adaptors for plug & produce systems*", WP4: "*Simulation & Visualisation Methods to support Re-configurability*" and WP5:"*Integration & Deployment Planning*" to refine the technologies developed within the project.

1.2. Document Outline

A structured methodology has been used to define the work for Task 6.1, as detailed in Section 2. A summary of the requirements for each use case is presented in Section 3. Details about the foundations of the PERFoRM architecture such as the Middleware, Standard interfaces, Technology adaptors, Human-machine interfaces, Analytics and Simulations, are presented in Section 4.

As the architectures of each use case are complex and diverse, detailed flow of information within the sub-systems for each use case are explored within Section 5. This Section also presents a list of assets available at the test lab(s) (MTC and SmartFactory) and also pre-test beds (such as infrastructure present at IPB's and Loccioni's facilities) to give the reader a flavour of the diverse assets that may be available to test the readiness of the technologies developed within this project. Test scenarios relevant for Task 6.1 are presented in Section 6. Finally, the conclusions and future work are presented in Section 7.

2. Methodology for Task 6.1

A testing methodology (see Figure 5) for Task 6.1 was proposed and utilised to get a better understanding of the tasks to be performed. This methodology includes the core tasks that need to be performed in order to satisfy the requirements of WP6. It is to be noted that the scope of this deliverable does not cover Steps 4b, 6, 7, 8, 9 and 10 (indicated in italics and in light blue rectangles in Figure 5). These steps will be addressed in future deliverables of this work package. Figure 5 also illustrates the section numbers of the document that corresponds to the different steps within the methodology.

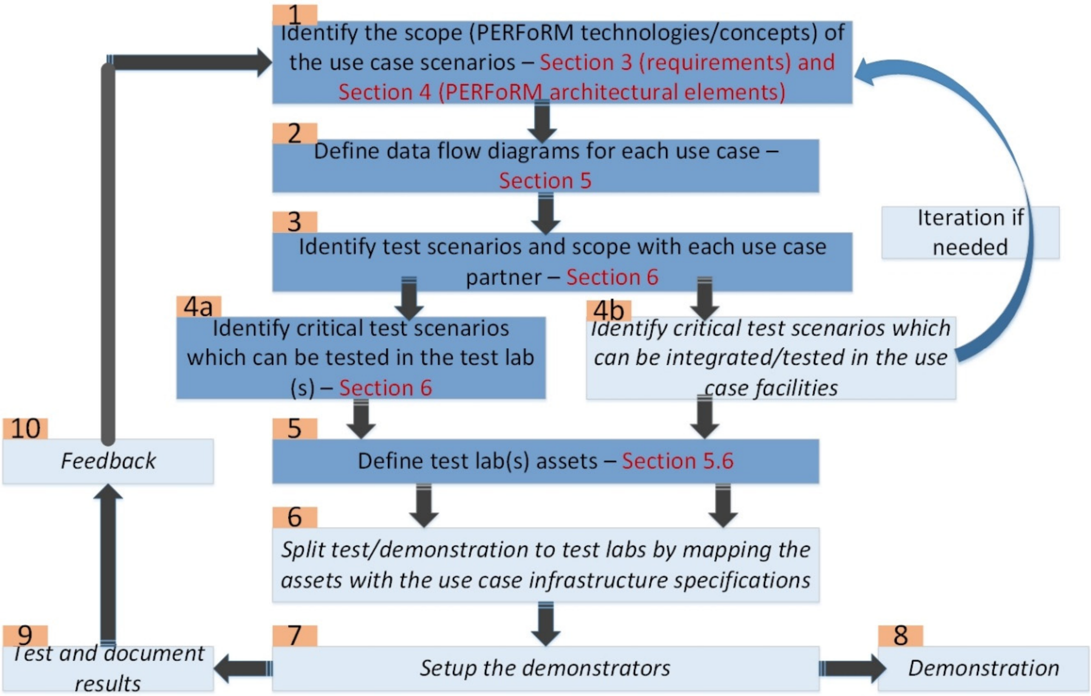

1
Identify the scope (PERFoRM technologies/concepts) of the use case scenarios – Section 3 (requirements) and Section 4 (PERFoRM architectural elements)
2
Define data flow diagrams for each use case – Section 5
3
Identify test scenarios and scope with each use case partner – Section 6
Iteration if needed
4a
Identify critical test scenarios which can be tested in the test lab (s) – Section 6
4b
Identify critical test scenarios which can be integrated/tested in the use case facilities
10
Feedback
5
Define test lab(s) assets – Section 5.6
6
Split test/demonstration to test labs by mapping the assets with the use case infrastructure specifications
9
Test and document results
7
Setup the demonstrators
8
Demonstration

Figure 5. Testing methodology for Task 6.1 – text in italics indicate items not in scope of this deliverable. Section numbers indicated

Step 1: includes identification of the scope and requirements of the use case scenarios realising the concepts and technologies of the PERFoRM project. The requirements of individual use cases are presented in detail in D1.2. This deliverable provides a snapshot of the requirements of the use cases within Section 3.

Step 2: includes understanding the flow of information within the ecosystem via the use of detailed data flow diagrams (DFD). These diagrams (Yonglei, 1991) are an easy, visual way of representing flow of data and is widely used to specify large complex systems. The DFDs have been verified by the use cases.

Test case generation is one of the key activities for validating that the technologies developed within this project conforms to the system requirements. Test scenarios are typically used to derive test cases (Xu, 2011) and will be designed within WP6. Step 3: involves the identification of test scenarios within the scope of Task 6.1. As seen in Figure 2, Task 6.1 is concerned with the asset view and therefore involves the validation of the PERFoRM ecosystem concepts within the realm of machines and robots. It is to be noted that the tasks performed within Steps 1 and 2 is also applicable to Tasks 6.2 and 6.3.

Amongst the test scenarios identified in the previous step, some of them can be validated at the test lab(s) (such as MTC and SmartFactory) (Step 4a) and some can only be validated within the facilities at the use cases (Step 4b) due to different needs e.g. a complete production line. This deliverable is responsible for identifying test scenarios that can be tested within Task 6.1, i.e. tests that can be conducted only at the machine and robot levels.

Step5: involves the definition of the assets available at the test lab(s). It is very important that the assets available within the test lab(s), as well as other within test beds available at other partner’s sites, are identified early on. Pre-test beds (i.e. infrastructure present at facilities of other partners) may be available for testing technologies developed within WP2 (such as the industrial middleware and standard interfaces) and WP3 (such as technology adaptors).

Step 6: shows that the effort for demonstration will be split between the test lab(s) based on the availability of assets and infrastructure. Step 7: involves setting up of the demonstrator (s), whilst Step 8 involves using the demonstrator to execute the test scenarios and document the results (Step 9). Step 10: involves sending the test feedback to the partners. Based on the availability of assets and infrastructure, the effort for demonstration will be split between the test lab(s), as seen in Step 6.

3. Use Case Requirements

WP1 has been instrumental in acquiring and collating the requirements of all the use cases. These requirements have been presented in Deliverable D1.2: "*Requirements for Innovative Production System Functional requirement analysis and definition of strategic objectives and KPIs*" (POLIMI, 2016) and a summary of the requirements identified for each use case is presented in this section. The primary aim of the requirements analysis phase was to identify the needs of the manufacturing industry to progress from the traditional control approaches towards an intelligent and dynamic manufacturing control systems, based on plug and produce production systems and self-adjusting devices implementation. An iterative methodology (derived from formal Requirements Engineering (RE)) was followed. The requirements and details of this approach are presented in Deliverable D1.2. This methodology includes four phases (i.e. Elicitation, Analysis, Specification and Validation). The first iteration was used to identify the requirements, analysing objectives, context and constraints, whereas the second iteration was utilised to validate the requirements.

The proposed methodology has been adopted for all the use cases, in order to identify which requirements and KPIs are to be implemented and benchmarked in a flexible and reconfigurable system to realise a next generation agile manufacturing paradigm, based on Cyber-Physical Production Systems (CPPS). The overall results are summarised in Table 1, which show the *General Requirements* and the *Others Requirements*. The *General Requirements* include only the requirements regarding flexibility and re-configurability that are required across all four use cases, whilst the *Other Requirements* column lists requirements that are specific to certain use cases and lead to attain those aspects of flexibility and re-configurability grouped in the previous section (*General Requirements*).

Table 1. General requirements: flexibility and re-configurability overview

General Requirements		**Others Requirements**
Flexibility	**Re-configurability**	**Necessary to flexibility and re-configurability**
• To change raw material • To change processes • To obtain process interactions • To create an agile production • To facilitate mobility,	• To obtain feedback from production to design • To obtain final test feedback to Robot system configuration • To obtain feedback to the process, based on failure control	100% Traceability and identification of single products up to the supply chain
		Ability to enable Simulation, Model and prototype in the CPS

including comparison among different units e.g. OEE, micro-flow-cells) • To reduce cycle time • To reduce cycle and cost	• To save cost in reconfiguration • To obtain new part reprogramming/setup through CAD critical paths • To have a self-configuring system, which can define the root-cause based on pattern recognition. • To reduce set-up time	environment (i.e. process parameters interaction, global factory behaviour, predictive failure)
		Increase the amount of data collected and data availability
		Automatic (semi-automatic) data gathering of machine condition
		Full integration and quick communication among different departments and functions (i.e. scheduling system and maintenance system integration, machine condition and maintenance tasks, production and process planning, etc.)

The proposed approach has been applied to the four use cases which are specifically described in the following sub-sections.

3.1. Use Case SIEMENS

The Siemens use case focusses on the production of compressors in a highly customised environment. The use case will look at retrofitting the existing production equipment. The PERFoRM architecture will enable product innovations and new product versions within a very short period thus increasing the flexibility of manufacturing these products. In order to ensure the flexibility, the productivity of the production line is a vital KPI for the factory. As detailed in D7.1: "*Siemens description and requirements of architectures for retrofitting production equipment*" (Siemens, 2016) the productivity depends on many factors such as ramp-up, set-up, maintenance, changeover and throughput capabilities. A top down-approach has been applied during the RE activities to identify the requirements (see

Figure 6).

Figure 6 also highlights the principal Key Performance Indicators (KPIs) needed to benchmark the requirements.

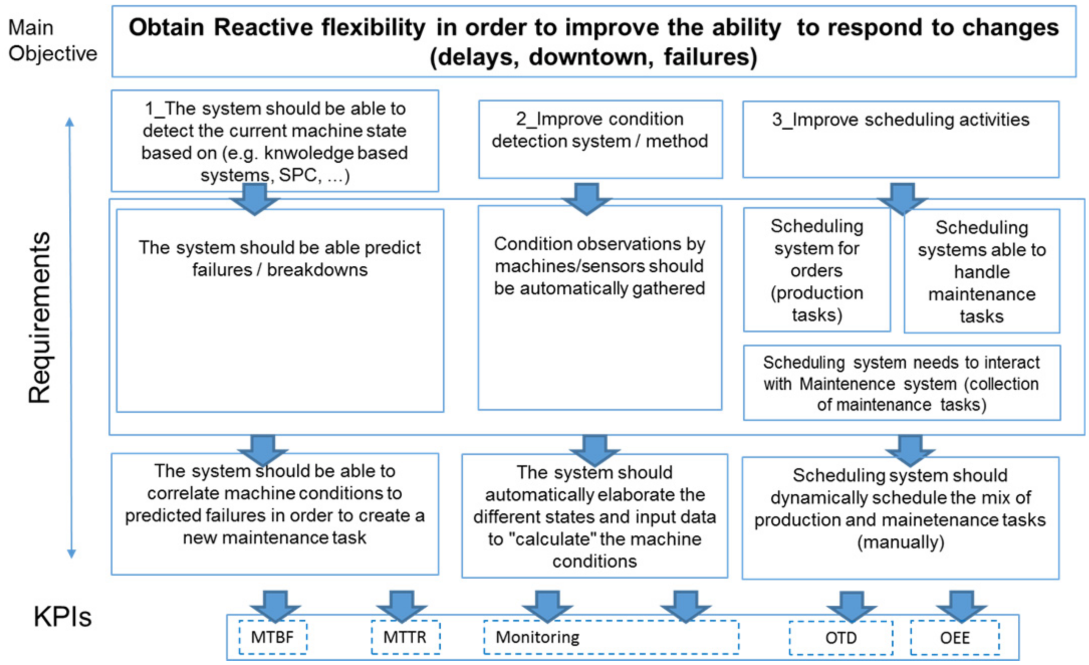

Main Objective
Obtain Reactive flexibility in order to improve the ability to respond to changes (delays, downtown, failures)
Requirements
1_The system should be able to detect the current machine state based on (e.g. knwoledge based systems, SPC, ...)
2_Improve condition detection system / method
3_Improve scheduling activities
The system should be able predict failures / breakdowns
Condition observations by machines/sensors should be automatically gathered
Scheduling system for orders (production tasks)
Scheduling systems able to handle maintenance tasks
Scheduling system needs to interact with Maintenence system (collection of maintenance tasks)
The system should be able to correlate machine conditions to predicted failures in order to create a new maintenance task
The system should automatically elaborate the different states and input data to "calculate" the machine conditions
Scheduling system should dynamically schedule the mix of production and mainetenance tasks (manually)
KPIs
MTBF
MTTR
Monitoring
OTD
OEE

Figure 6. Siemens: Framework for identifying requirements and KPIs

3.2. Use Case IFEVS

The IFEVS use case deals with the assembly of low cost full electric vehicles with high variants and high quality on low budget assembly lines. The principal objectives are to: (1) achieve a high degree of automation systems, (2) improve the efficiency and reproducibility of processes and enable highest product quality, (3) reduce re-work of sub-modules and part rejection and (4) minimise the variability of manual operations. A top-down approach has also been applied to this use case to identify the relevant requirements and their relation to the KPIs (see Figure 7).

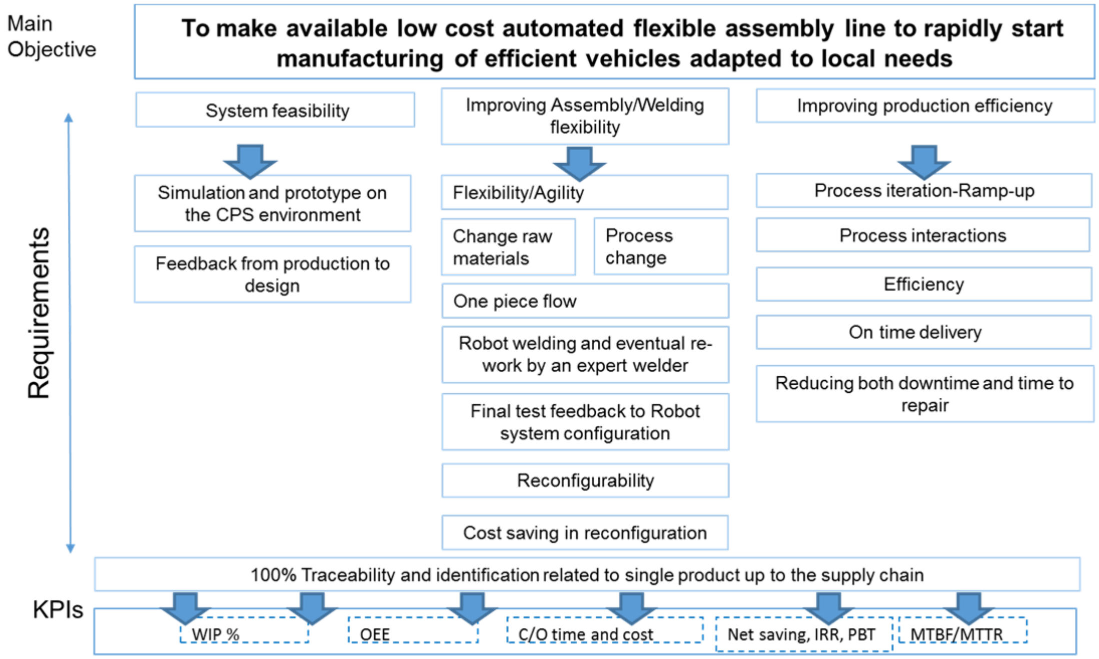

Main Objective
To make available low cost automated flexible assembly line to rapidly start manufacturing of efficient vehicles adapted to local needs
Requirements
System feasibility
Improving Assembly/Welding flexibility
Improving production efficiency
Simulation and prototype on the CPS environment
Feedback from production to design
Flexibility/Agility
Change raw materials
Process change
One piece flow
Robot welding and eventual re-work by an expert welder
Final test feedback to Robot system configuration
Reconfigurability
Cost saving in reconfiguration
Process iteration-Ramp-up
Process interactions
Efficiency
On time delivery
Reducing both downtime and time to repair
100% Traceability and identification related to single product up to the supply chain
KPIs
WIP %
OEE
C/O time and cost
Net saving, IRR, PBT
MTBF/MTTR

Figure 7. IFEVs: Framework for identifying requirements and KPIs

3.3. Use Case Whirlpool

The main objectives of the Whirlpool use case are to implement: (1) a real-time monitoring system that is able to correlate dynamic behaviour of the factory to the specific Key Performance Indicators (KPIs) and static indicators such as Key Business Factors (KBF) and (2) a reconfiguration support system that derives appropriate actions for reconfiguring processes based on factory KPIs. Following the same methodology, the principal requirements needed to reach these objectives and the specific KPIs used to monitor the requirements was identified (Figure 8).

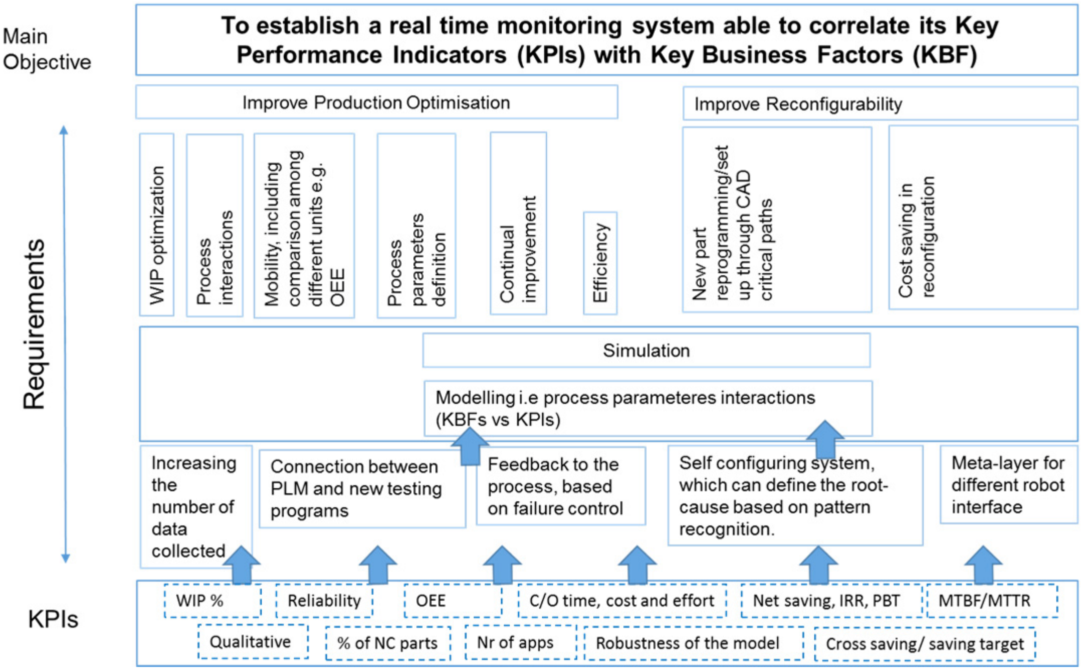

Main Objective
To establish a real time monitoring system able to correlate its Key Performance Indicators (KPIs) with Key Business Factors (KBF)
Requirements
Improve Production Optimisation
Improve Reconfigurability
WIP optimization
Process interactions
Mobility, including comparison among different units e.g. OEE
Process parameters definition
Continual improvement
Efficiency
New part reprogramming/set up through CAD critical paths
Cost saving in reconfiguration
Simulation
Modelling i.e process parameteres interactions (KBFs vs KPIs)
Increasing the number of data collected
Connection between PLM and new testing programs
Feedback to the process, based on failure control
Self configuring system, which can define the root-cause based on pattern recognition.
Meta-layer for different robot interface
KPIs
WIP %
Reliability
OEE
C/O time, cost and effort
Net saving, IRR, PBT
MTBF/MTTR
Qualitative
% of NC parts
Nr of apps
Robustness of the model
Cross saving/ saving target

Figure 8. Whirlpool: Framework for identifying requirements and KPIs

3.4. Use Case GKN

The main objective of the GKN use case is to: (1) implement a Micro-Production Flow Cell that is able to reduce change over time and to realise different products and (2) be able to give a flexible and reconfigurable aspect to the whole industrial plant. The top down methodology has been used to identify the principal requirements and the specific KPIs used to monitor the requirements behaviour during PERFoRM project development (Figure 9).

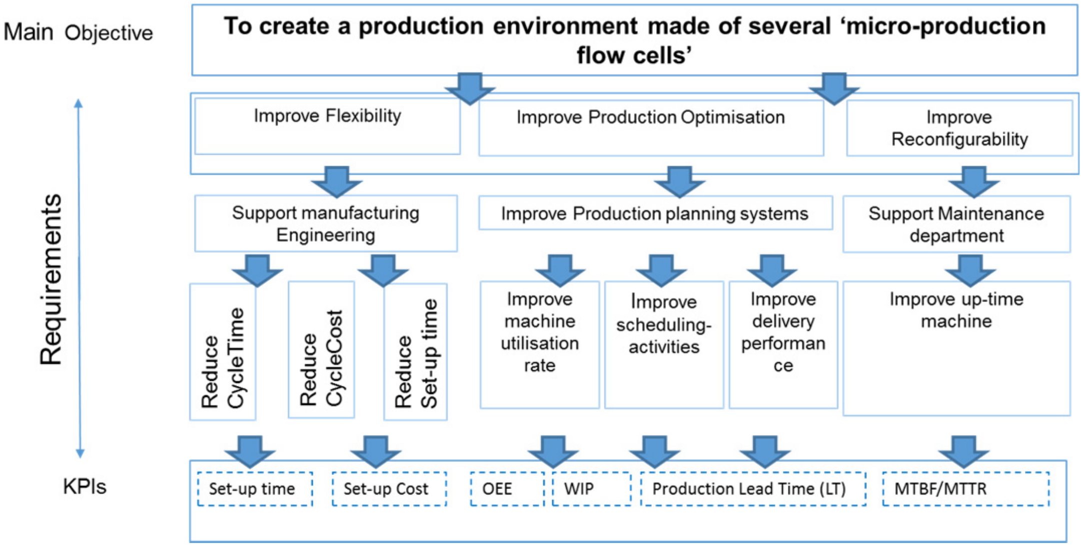

Main Objective
To create a production environment made of several 'micro-production flow cells'
Requirements
Improve Flexibility
Improve Production Optimisation
Improve Reconfigurability
Support manufacturing Engineering
Improve Production planning systems
Support Maintenance department
Reduce CycleTime
Reduce CycleCost
Reduce Set-up time
Improve machine utilisation rate
Improve scheduling-activities
Improve delivery performan ce
Improve up-time machine
KPIs
Set-up time
Set-up Cost
OEE
WIP
Production Lead Time (LT)
MTBF/MTTR

Figure 9. GKN: Framework for identifying requirements and KPIs

4. Foundations of the PERFoRM architecture

The PERFoRM architecture defines a method to address the seamless production system re-configuration, combining the plug-and-produce concept and the human element as a flexibility driver in future production systems. The overall system architecture of the PERFoRM project can be seen in

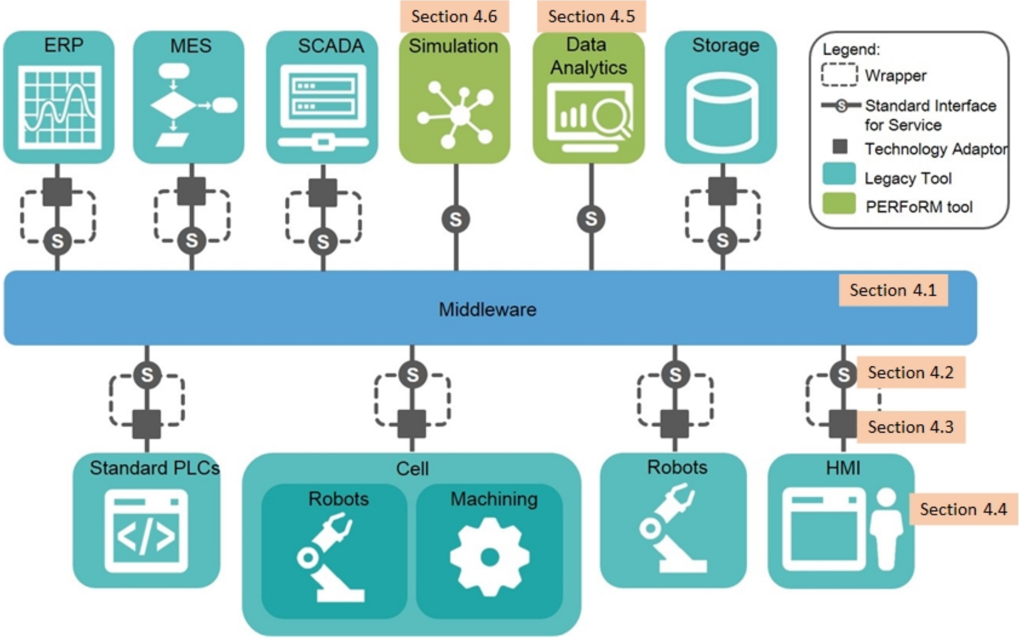


Figure 10. The different elements of this proposed architecture are presented in the following sub-sections. The ERP, MES, SCADA and DB are proprietary elements (seen in light blue boxes within

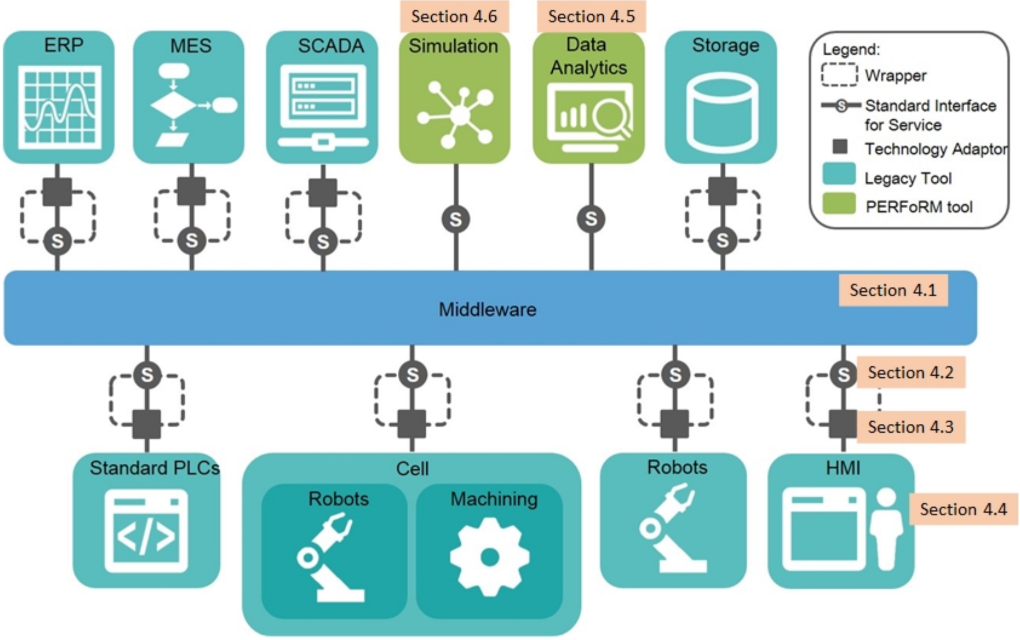
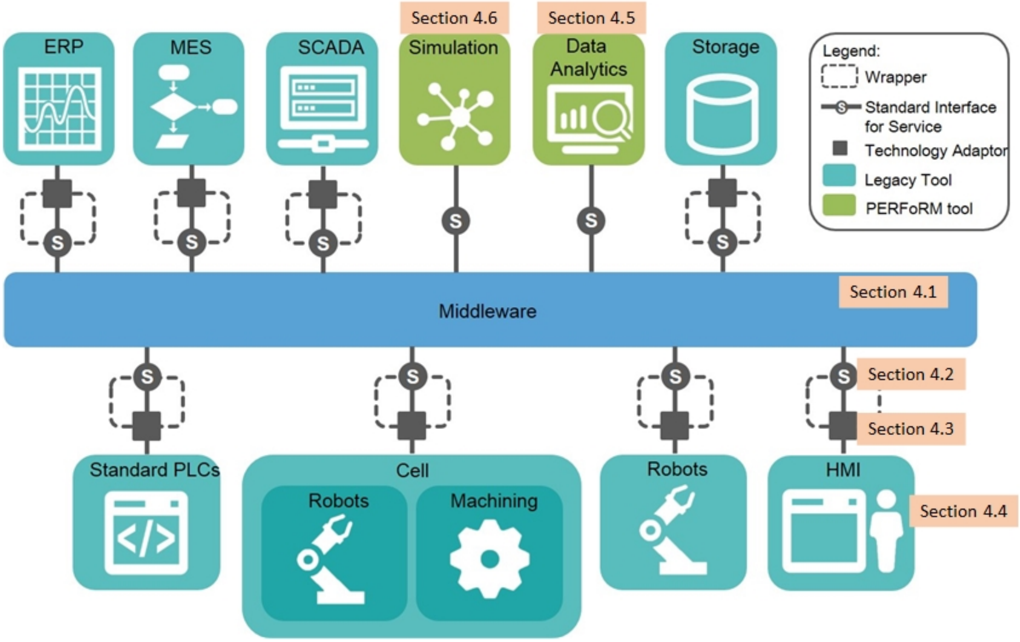
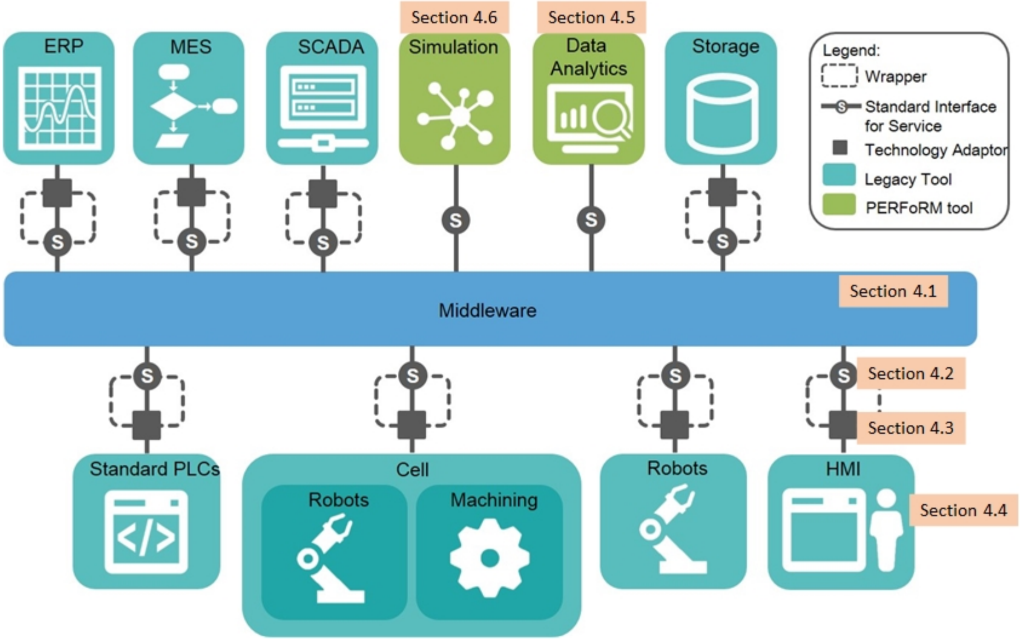


Figure 10) and will be used in the industrial use cases.

The boxes in green colour (i.e. *Simulation* and *Data Analytics*) are being developed as part of this project. Even though validation of the *Simulation* and *Data Analytics* falls within the realm of the *architecture view* (Task 6.2, see Figure 10), a brief description about these is provided in this document. The sub-sections corresponding to each of the elements can also be seen in Figure 10.





The impact and influence of human integration as a flexibility driver in the production systems have been addressed in Task 2.1: “Influence of Human as flexibility driver in production systems” and the results from this task will act as a basis for Task 3.3: “*Incorporation of Human observation*” and Task 4.3: “*Automatic Monitoring of visualisation of KPIs*”, where human observations, monitoring and visualisation of KPIs will be addressed.

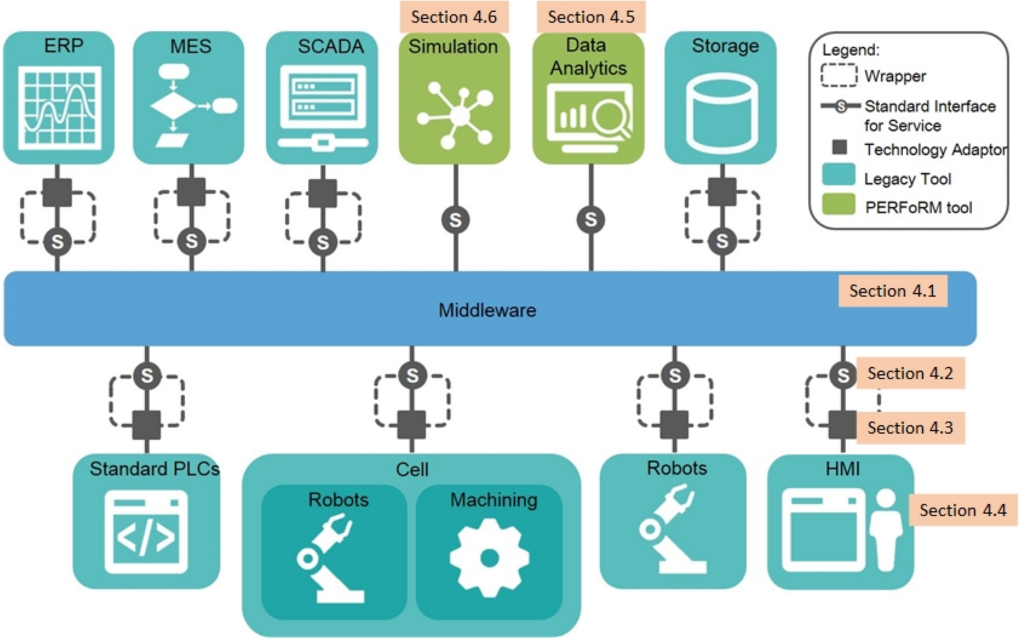


Figure 10. PERFoRM architecture – indicates the section numbers within the document that explains each element

4.1. Middleware

Interoperability is a key issue when connecting different applications and systems. One way of solving this problem is using a Middleware. A Middleware in general is a software component, targeted at connecting different applications or systems in a way that enables communication between the involved actors without them having to know about their inner structure and specific interfaces. These components are becoming increasingly important for industrial systems, as they can act as an interfaces between production level devices or systems and higher level production management systems (vertical integration). While this is one of the main aspects, the horizontal integration of different systems on the same level (e.g. different control systems communicating with each other) can also be achieved.

In PERFoRM, Task 2.4: "*Industrial manufacturing middleware*" is researching different Middleware technologies in depth by defining multiple basic functionalities that a Middleware should provide to meet the requirements that are being set by the goals of PERFoRM. One of the main requirements is to find Middleware solutions that are viable for productive use in industrial applications. This means, that the research is concentrating on evaluating and using existing Middleware solutions instead of developing new solutions. To do that, multiple existing Middleware solutions of leading developers in this area are being tested on their general abilities and also on their fit to the individual use cases of the PERFoRM project. These solutions will act as the core component of the PERFoRM Middleware solution. Another major requirement is the possibility to add new functionality to these existing solutions, to adapt them to the specific PERFoRM requirements.

Figure 11 is showing the inner structure of the PERFoRM Middleware, where the aforementioned industrial solutions will act as the core component to deal with the basic functionalities needed for the Middleware. This core is going to be interchangeable to avoid focusing on one specific solution. If some basic functionalities, such as a Yellow Pages system is not provided by the core itself, these need to be added externally. More complex abilities like a complex orchestration engine are not planned to be part of the basic Middleware itself, but the Middleware should provide possibilities to add these as an Add-On. Furthermore, the development of wrappers (which is a piece of software which is necessary whenever existing solutions are used within the PERFoRM Middleware) from the core individual solutions to the Standard Interfaces defined in Task 2.3: "*Design of standard interfaces for machinery, control systems and data backbone*" is necessary. Deliverable D2.4: "*Industrial manufacturing middleware: specification, prototype implementation and validation*" gives a more in depth description of all the evaluation approach, the different components of the Middleware and the development made.

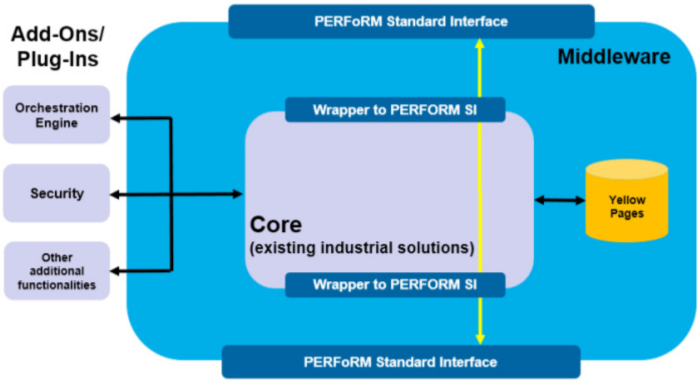

Add-Ons/ Plug-Ins
Orchestration Engine
Security
Other additional functionalities
PERFoRM Standard Interface
Middleware
Wrapper to PERFORM SI
Core
(existing industrial solutions)
Wrapper to PERFORM SI
Yellow Pages
PERFoRM Standard Interface

Figure 11. Inner structure of the PERFoRM Middleware

4.2. Standard Interfaces

Following the Industrie 4.0 movement, PERFoRM targets the full interoperability of heterogeneous devices in real industrial environments. This aspect presents a strong relation to the representation and seamless exchange of manufacturing data between varied entities across the entire value chain. As such, a key point to be considered is the integration and interconnection of legacy hardware devices (e.g. robots and the respective controllers) and software applications, such as databases, SCADA, planning and scheduling, simulation and other management and logistics applications.

For this purpose, the adoption of standard interfaces as the main drivers for plug-ability and interoperability is used as the means to enable the connection between the different system elements in a seamless and transparent manner. Through these interfaces each component is capable of fully describing and exposing its services in a standardised way, by means of a clear specification of the semantics and data flow involved in these interactions.

In order for such seamless connectivity to be possible, a common language needs to be "spoken" by each of the intervenient. Hence, a shared data model is also adopted, serving as the data exchange format shared between the PERFoRM-compliant system elements. The semantic needs associated to each of these elements, particularly the requirements related to each of the ISA-95 layers in the case of manufacturing automation, are therefore contemplated within the PERFoRM Data Model. In this context, two particular data abstraction levels are considered, one covering mainly layers one (automation control) and two (supervisory control), hereby named "Machinery Level", and the "Data Backbone Level", which covers layers three (manufacturing operations management) and four (business planning and logistics). This division can be seen in

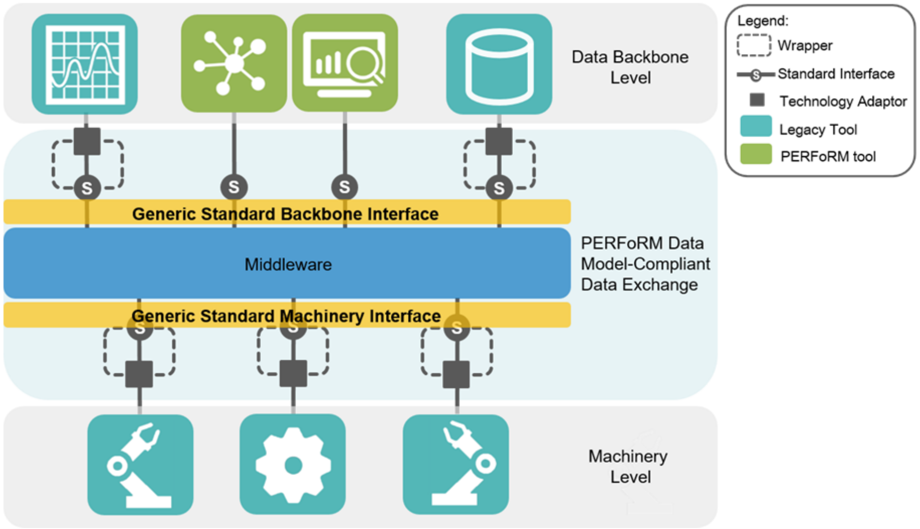


Figure 12.

As illustrated in

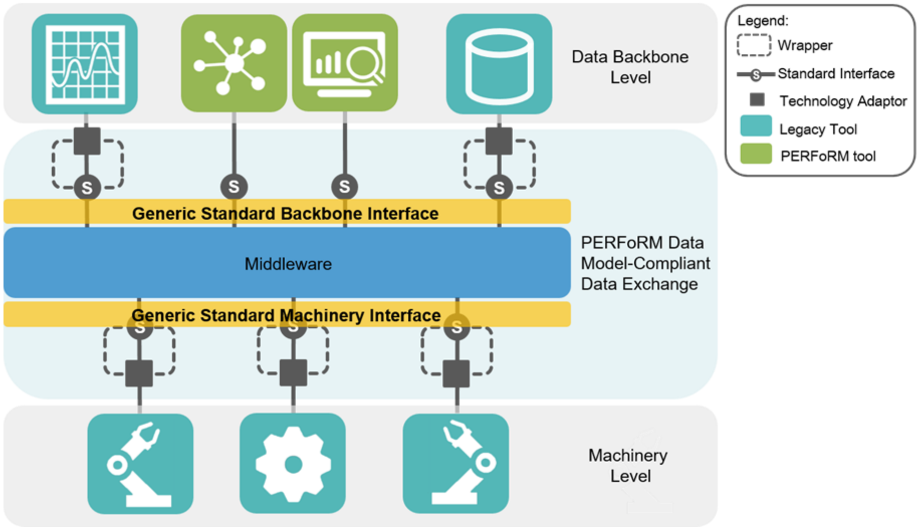


Figure 12, full interoperability and harmonisation of data at a system of systems level is achieved by empowering the standard interfaces with a common representation of data and well-defined semantics through the PERFoRM Data Model. Regardless, when referring to the integration of legacy devices one needs to take into account the existence of their own individual data models and semantic requirements. Hence,

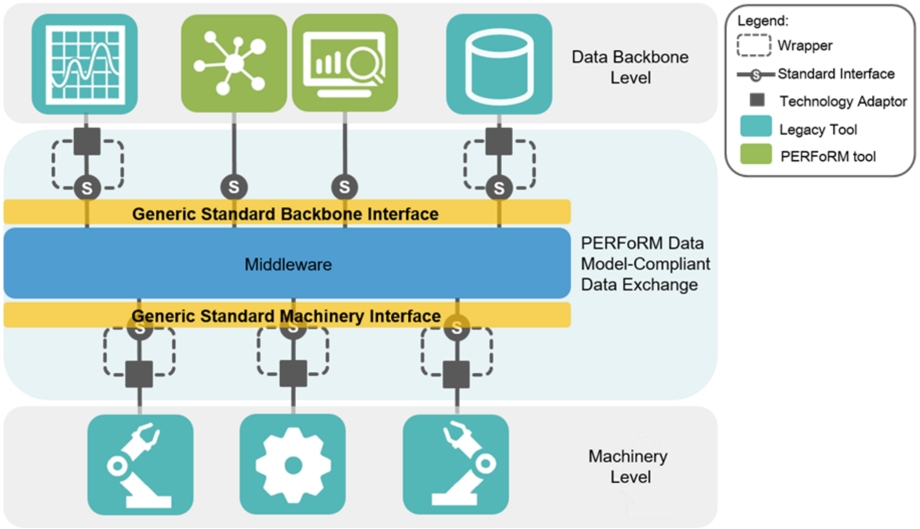


Figure 12 shows the combination of the standard interfaces with technology adaptors which enable the translation and mapping of legacy data into the common PERFoRM representation. This combination makes it possible for these devices to be conferred additional intelligence and integrated seamlessly into the PERFoRM context. The description of the technology adaptors will be addressed in further detail in subsection 4.3.

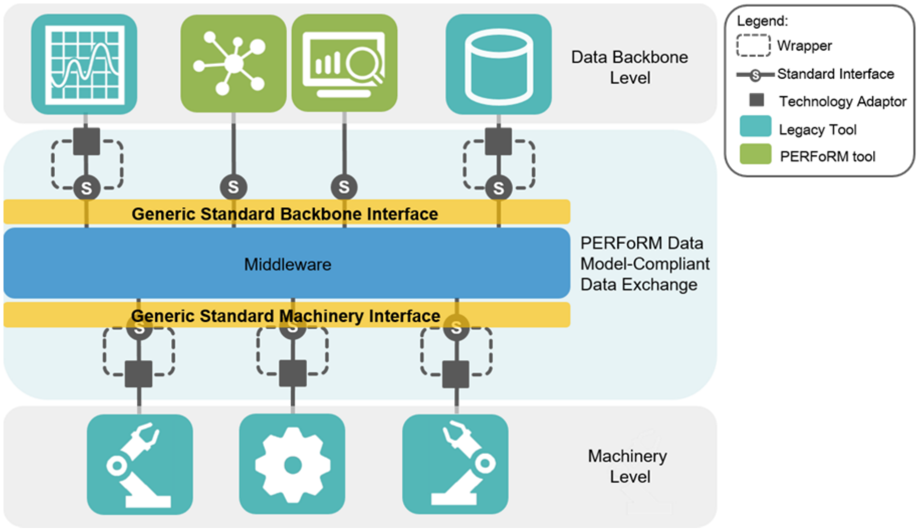


Figure 12. Task 2.3's integration into the PERFoRM Architecture

4.3. Technology Adaptors

Manufacturing companies are usually characterised by the use of legacy and heterogeneous systems for the management and the execution of their production process. At machinery level (L1 and L2 layers of ISA 95 standard) example of these systems are robots, CNC machines, PLCs and Human Machine Interfaces (HMIs); at backbone level (L3 and L4 layers of the ISA 95 standard) examples of these systems are ERP, MES and Production Databases (DBs). The innovative architecture proposed in the PERFoRM project can be industrially accepted and really adopted only if the possibility to integrate the legacy systems is presented. For this reason, technology adaptors are key elements to connect legacy systems to the PERFoRM middleware and to transform the legacy data model into the standard interface data model defined in Task 2.3: "*Design of standard interfaces for machinery, control systems and data backbone*" of the project.

As depicted in Figure 13 (RT in the diagram implies real-time), three different kinds of adaptors are being considered in the WP3 and addressed within its tasks. These adaptors respond to the different types of legacy systems which can be found in a production environment and are able to seamlessly connect these systems with the industrial middleware and the higher level of the enterprise network (ERP, MES, etc.).

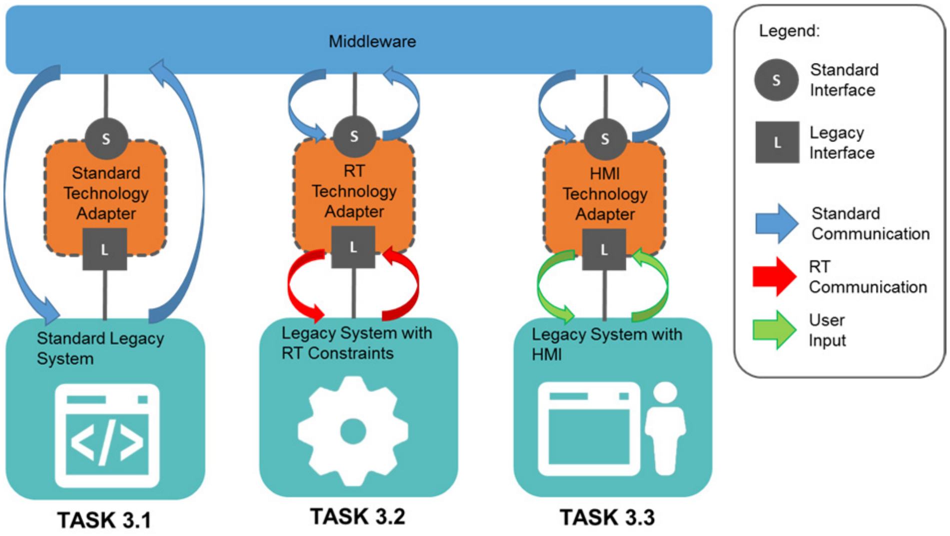


Figure 13. Technology Adaptors Types

Real-Time constraints are particularly important when considering CNC machines and robotic cells as they may need quick adjustments and corrections according to the data acquired from low-level sensors locally installed in the production resource (e.g. vibration analysis of spinning spindles). Currently, no hard real-time constraints were identified in the use cases. However, should they arise, this aspect will be tackled within Task 3.2: "*Real-time process information*", as depicted above. The HMIs, instead, can be used not only for monitoring and controlling the production resource but also for capturing human expert knowledge and support following human activities from past experience (e.g. change over and ramp-up operations can be supported by policies derived from past cases). Following the indications coming from the WP1, for each of the four use cases addressed in the PERFoRM project a list of the legacy systems that need to be connected with industrial middleware has been created and reported in Table 2.

Table 2. Legacy system that needs connection within each use case

Use Case	**Objective**	**Legacy Systems**
Siemens Compressors	Integration of a predictive maintenance system	➢ EPR System (SAP APO) ➢ BDE Data Logging System (Oracle DB) ➢ LHnet Ticketing System (SQL DB) ➢ CNC Machines (SINUMERIK 840D)
IFEVS Micro-	Automation of the	➢ Welding Robotic Cells

Electrical Vehicles	production line	and Powertrain Testing Stations (Siemens PLC IM-151)
Whirlpool Microwave Ovens	➢ Implementation of a KPI real-time monitoring system ➢ Reconfiguration of the path of the robot for the leak test	➢ PERFoRM DB (SQL DB) ➢ PLM Repository (txt file) ➢ Leak Robot Station (UR10 Controller)
GKN Turbine Vanes	Construction of a reconfigurable robotic cell	➢ Robotic Cell PC/PLC ➢ Roughness Process (Mitutoyo SJ-210)

The integration of the hardware equipment and software applications listed above requires the use of appropriate technological adaptors to transform the native data format into the data model defined by PERFoRM. The specifications of the data model (comprised of the standard interfaces) will be covered in Task 2.3: "*Design of standard interfaces for machinery, control systems and data backbone*", Fupreliminarily work considers using a joint solution using B2MML and AutomationML (R.S Peres, 2016): B2MML (IEC 66264) implements the ANSI/ISA-95 family standards, but lacks of low-level data (PLC signals, I/O and control sequences), which is covered in AutomationML (IEC 62714).

Moreover, the implementation of the adaptors is strongly dependent on the selected technology for the industrial middleware (Task 2.4: "*Industrial manufacturing middleware*" of PERFoRM). For example, Siemens WinCC OA (Open Architecture) provides a direct PLC interface that greatly simplifies the implementation of adaptors for such kind of hardware equipment. PLC integration is particularly important for the IFEVS and GKN use cases, where PLCs are used to control the robotic cells. Another example of middleware technology considered in the PERFoRM project is the IBM Integration Bus. This solution offers, among other interesting features, a Database Input node that permits retrieval of updated data directly from a database: it creates a message flow that quickly reacts to changes to application data held in the database. Database connection and integration is particularly important for the Siemens and Whirlpool use cases, where databases contain the information needed for feeding the predictive maintenance system (Siemens use case) and the KPI monitoring systems (Whirlpool use case).

4.4. Human-machine interfaces

The analysis of possible scenarios for human integration in the flexible production systems within the PERFoRM industrial use cases has been undertaken in Task 2.1: "*Influence of Human as flexibility driver in production systems*" and the results have been reported in D2.1: "*Guidelines for seamless integration of Humans as flexibility driver in flexible production systems*" (POLIMI, 2016). In particular, two types of human roles have been identified in Task 2.1: Human-in-the-Loop (see Table 3) and Human-in-the-Mesh (see Table 4). The results present some implications with regard to the requirements for design human-machine interfaces for each role.

Table 3. Human-in-the-Loop requirements

Mobile devices with context aware (role, location) support

Support visual inspection with sensors
Support testing (geometrical, power train, fatigue, etc.)
Virtual presence (for consulting expert colleagues: sharing view, screen, info, voice connection or chat)
Multimodal interaction (voice, image, gesture recognition, sound lights, etc.) to alert and to support field-work
Suitable/wearable device to support field-work
Asset tracking (tools and spare parts)
Localisation and turn-by-turn navigation to retrieve machines, tools, spare parts.

Table 4. Human-in-the-Mesh requirements

Mobile, context aware (role, location) support
Intuitive representation of alternatives and trade-offs
Decision support enhanced by experts' decision-making patterns

It is important to underline that these results were achieved in the first few months of the project and consequently reflect the still rather unclear definition of the use cases at this early stage of the project. Deliverable D2.5: "*Guidelines for seamless integration of Humans as flexibility driver in flexible production systems 2nd iteration"* will conduct further work and will present the following results:

- more detailed Human requirement definition;
- final human requirement specification and their related human-interface recommendation;
- Prioritisation of human-interface, identifying those are more coherent with use case needs.

4.5. Analytics and Visualisation

Within Task 4.3: "*Automatic monitoring and visualisation of KPIs*" different data analytics methods will be developed based on results of previous R&D projects (as indicated within the proposal). The data analytics methods should provide information about the condition of a production machine and enable to run maintenance processes in an optimal manner. Such a system consists of data acquisition, pre-processing, analysis and interpretation. A connection to the PERFoRM Middleware is needed to provide production process data. To evaluate the data analytics methods developed, a pre-test bed environment is provided. The *Data Analytics* aspect of the PERFoRM project falls within the realm of the *architecture view* (Task 6.2, as seen in Figure 2) and consequently the corresponding test scenarios will not be described in this deliverable.

The *Data Analytics* system is based on various sources of machining and process data. The data transportation must be ensured by the middleware such that relevant data from machine / ERP / MES level is available to the data analytics module (as shown in Figure 14). In general, data regarding the actual machining behaviour (e.g. active power, reactive power, power factor, vibration) recorded by external sensor nodes, process data (e.g. actual process parameters, use of specific tools) and maintenance information, are required with a predefined data rate depending on the methodological approach of data analysis and extraction. The Middleware must satisfy these specific requirements for data transfer rates as well for latency needs. The required data depends strongly on the demands and the properties of the specific use case.

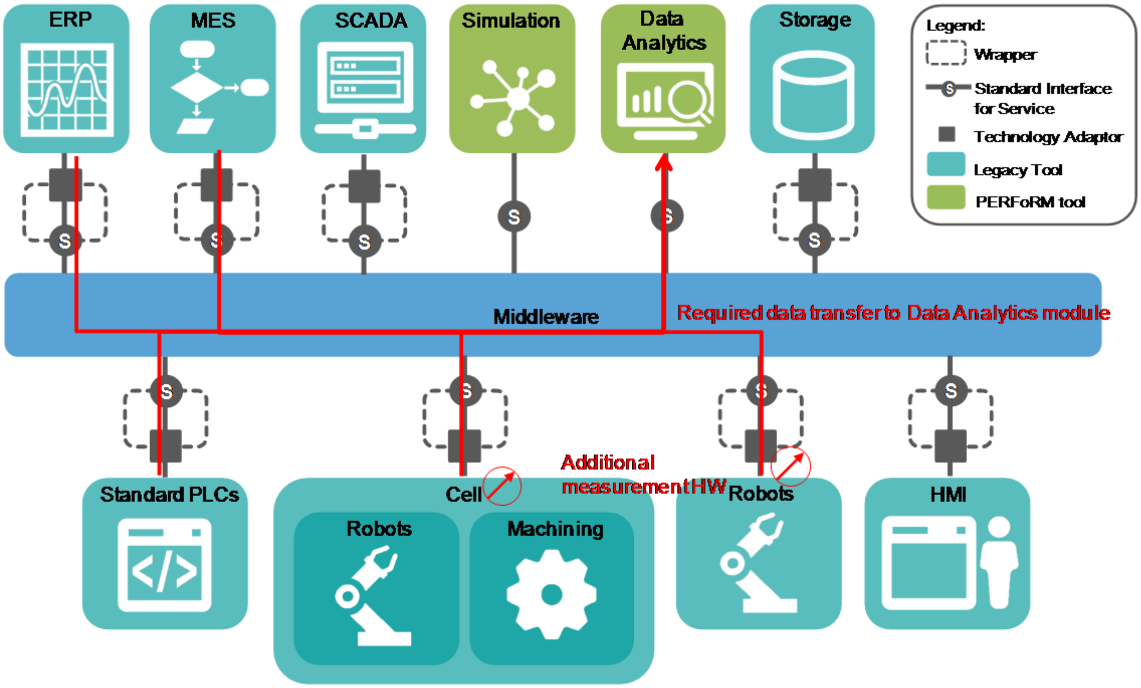


Figure 14. General overview over the PERFoRM system; data transfer between the Data Analytics module and the entities through the Middleware highlighted

4.5.2 Overview of Data Analytic functionalities

The measurement of power and vibration data and the correlation with the data of maintenance activities and production information (see

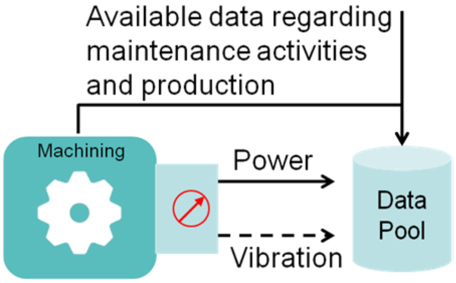

Available data regarding maintenance activities and production
Machining
Power
Vibration
Data Pool

Figure 15) present a basis for predictive maintenance methodologies. Ongoing research projects are working on defining the placement and location of power and vibration signals measurement on production machines.

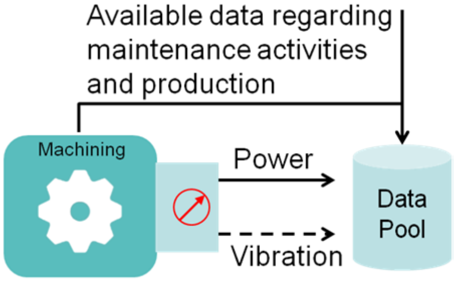

Available data regarding maintenance activities and production
Machining
Power
Vibration
Data Pool

Figure 15 demonstrates the placement of one power sensor and one vibration sensor on the machine housing additionally to the available data from a factory data acquisition system. The acquired data will be saved within a data pool and computational analysed to find correlations between power/vibration signals (state variables) and industrial/operating data capture.

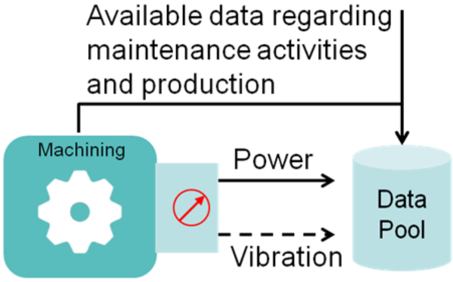

Available data regarding maintenance activities and production
Machining
Power
Vibration
Data Pool

Figure 15. Entity based data acqusition

Fast Fourier Transformation (FFT) is a widely used method for data extraction within the industrial environments on steady state conditions. In the case of non-stationary states, short-time Fourier transformation can be used to create a signature/fingerprint of the actual machine's state. With this procedure the machine condition can be tracked over time. In the case of deviations of the signature, an alarm signal will be triggered and maintenance action is suggested.

4.5.3 Overview of Visualisation Functionalities

The PERFoRM project will lead to enhancements in production environments due to simulation, re-configurability, rescheduling and data analytic approaches. A proper monitoring tool is needed to assess the KPIs. The monitoring tool should satisfy the miscellaneous facets of the project results. To fulfil these requirements a generic visualisation tool is needed. Figure 16 depicts the first generic user interface wireframe model for different entities and a whole process chain. Cells could be individually arranged and additional functions could be deployed with a single click. Basic analysis, e.g. basic feature extraction, could be easily established by using the graphic user interface. More complex analysis functions could be realised by implementing them as new modules in Node.js. The visualisation is web based (HTML5, Node.JS) and fits automatically to several devices, e.g. smart phones, tablets, desktops. Navigation through menus is intuitive and easy to learn.

Different channels of communication to the PERFoRM middleware or to other data sources are possible. The visualisation kernel can act as a web socket client or a web socket server.

Connection to different data base formats (e.g. MySQL) can be established and the data gathered by a periodic data pull. The possibility to establish an OPC UA connection to the middleware is added.

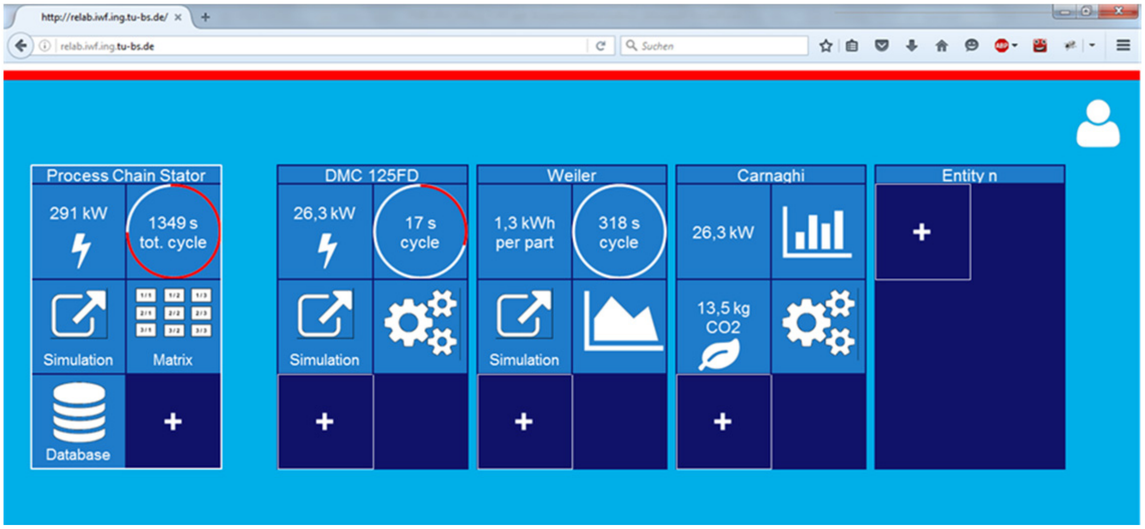


Figure 16. Wireframe model of generic user interface

4.6. Simulation Environment (Siemens AG)

Within the PERFoRM project, plant simulation tools (e.g. Technomatix PlantSimulation and AnyLogic) can be used to simulate the material flow through the workstations regarding logistic aspects of production process like lead time, MTTF, scheduling (could also be optimised with evolutionary algorithm). These evaluations could be done offline for planning issues, like reconfiguration or maintenance, and online in parallel to operation (e.g. for KPIs) taking current conditions (e.g. change request in current production line, machine break down) into account. The simulation environment is a wrapper for plant simulation tools to be used within PERFoRM architecture, connected to middleware, for performance evaluation and reconfiguration aspects including appropriate control logic. WP6 needs to provide a test environment to validate and optimise the different workflows for and within the simulation environment. Further details on the testing approach will be provided within Task 6.2, as this validation of the Simulation Environment falls within the realm of the *architecture view* (as seen in Figure 2).

Figure 17 illustrates the Simulation Environment test set up, with four different interfaces as listed below:

(1) Interfaces for getting data from the middleware into the simulation environment. These interfaces are used to validate model generation and configuration of specific plant simulation models in different use cases.

(2) Interfaces for sending data from the simulation environment to the Middleware. These interfaces are used to validate simulation outputs and KPIs.

(3) Interfaces for getting data from control planning logic into the simulation environment. These interfaces are used to validate the self-adaptive and reconfigurable mechanism as part of the control planning logic.

(4) Internal interface between Extendable model library base and Specific Simulation Model (refer Figure 17) within the simulation environment for execution of the simulation model

itself (actual simulation tools are Plant Simulation and AnyLogic). These interfaces are used to validate the self-adaptive and reconfigurable mechanism as part of the control planning logic.

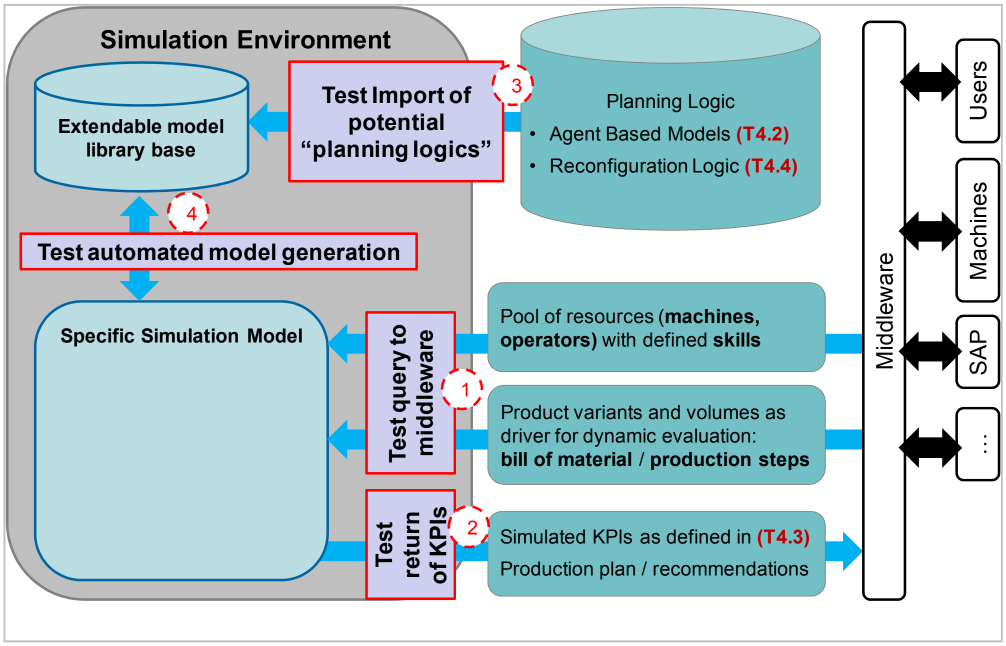

Simulation Environment
Extendable model library base
Test Import of potential "planning logics"
3
Planning Logic
• Agent Based Models (T4.2)
• Reconfiguration Logic (T4.4)
4
Test automated model generation
Specific Simulation Model
Test query to middleware
1
Pool of resources (machines, operators) with defined skills
Product variants and volumes as driver for dynamic evaluation: bill of material / production steps
Test return of KPIs
2
Simulated KPIs as defined in (T4.3)
Production plan / recommendations
Middleware
Users
Machines
SAP
...

Figure 17. Details for the necessary test interfaces for simulation environment

Further details of the interfaces will be provided in the deliverables in Task 6.2. A test environment consisting of test cases / requirements for the above mentioned interfaces, will be defined as well as realised within Task 6.2.

5. Data Flow Diagrams

As the architectures of each use case are complex and diverse, detailed flow of information within the sub-systems for each use case are explored within this section. The main concept behind the Siemens use case is to retrofit existing production equipment to become decentralised intelligent machines which improves availability and productivity. Consequently, the architectures depicted for the Siemens use case illustrate systems that are already in place.

The IEFVS use case will design new production line. The architecture has been proposed by IEFVS and COMAU. WP6 has worked in conjunction with IEFVS to design the flows described in the following sub-sections.

The WHR use case will involve transforming existing production resources (e.g. assembly and quality control stations) into plug and produce production resources by implementing different communication interfaces to link the existing stations with the PERFoRM platform. The architecture for this use case was proposed by WHR and WP6 has worked in conjunction with WHR to design the flows described in the following sub-sections.

The GKN use case involves the development of a "micro-flow" cell that can adapt to multiple products and processes. The architecture of this use case has been proposed by GKN with inputs from other members of the consortium.

5.1. What is a Data Flow Diagram?

A data flow diagram (DFD) is a visual representation of the data movement, inputs and outputs in a system. Creating DFDs can be considered as a preliminary step to create an overall view of the whole system. This view can be extended later. Though a DFD mainly shows how data moves from one process to another, as well as its logical storages it does not show any timing or information about sequence of processes (this information is typically depicted via a flowchart) (Donald S. and Le Vie, 2000). The logic of data flow within the system is supported by DFDs. A DFD mainly explains:

1. what are the inputs and outputs within the system
2. where are the source and destination of data
3. where the data is be stored

A DFD can be decomposed into levels, each level provides more detailed information flows in comparison to the previous level. The process of creating DFDs typically consists of a high level diagram (Level 0) followed by lower level diagrams (Level 1, Level 2…) (Aleryani, March 2016).
"Level 0" (or "Context diagram") is the first step of decomposing the system and consists of:

1. All external entities, e.g. Customer, Supplier
2. A single process labelled "*0*" which represents the main functionality of the system
3. The data flows between the external entities and the main process

A *Level 1* DFD is the second step of decomposition and it involves:

1. Providing an overview of how the main process in *Level "0"* is broken into sub-processes and illustrates the data flows through the system
2. Identifying the potential data stores that are used by the major processes within the system
3. Providing information of all the assumptions about the data flow in the system

Depending on a system's complexity, it is possible to produce *Level 2 DFD, Level 3 DFD* etc.

5.1.2 Why use a DFD?

- It helps in describing the boundaries of the system,
- It helps to transmit existing system knowledge to the users,
- It is a straightforward graphical technique,
- DFDs can give a point by point representation of the system segments,
- It is utilised as the part of system documentation file,
- DFDs can be understood by technical and non-technical people. (Aleryani, March 2016)

There are other methods for business process modelling such as Process maps and Activity diagrams. But the DFDs have been preferred for (Advanced Strategies, 2008) this work as:

- Process maps don't work very well if the flow of information is complex (Advanced Strategies, 2008),
- Activity diagrams are not very suited for showing essential information dependencies (Advanced Strategies, 2008).

5.1.3 DFD Symbols and Definitions

DFDs are created using a number of symbols and the definition of these can be seen in Table 5.

Table 5. DFD's symbols and definitions, (Yen, 2011)

		A process transforms data, e.g. creates, modifies, stores, delete, etc. It can be manual or

Process		supported by computer.
Data store		A data store contains information, which is stored and accessed. It can be both in paper file folder or a database.
External Entity		External entity is the source or destination of system's data. Entities are objects outside the system.
Data flow		A Data flow is the pipeline that allows data to flow between processes, data stores and external entities.

5.1.4 What does a DFD look like?

In order to explain a DFD, an example of a Salad Ordering System has been illustrated by Figure 18. The process consists of three processes, four external entities (Customer, Kitchen, Manager and Supplier) and two data stores (Inventory, Order data). In this DFD (as following DFDs within this document), the processes have been numbered and the numbers will be further used in the text in the form of "ProcessName (ProcessNumber)".

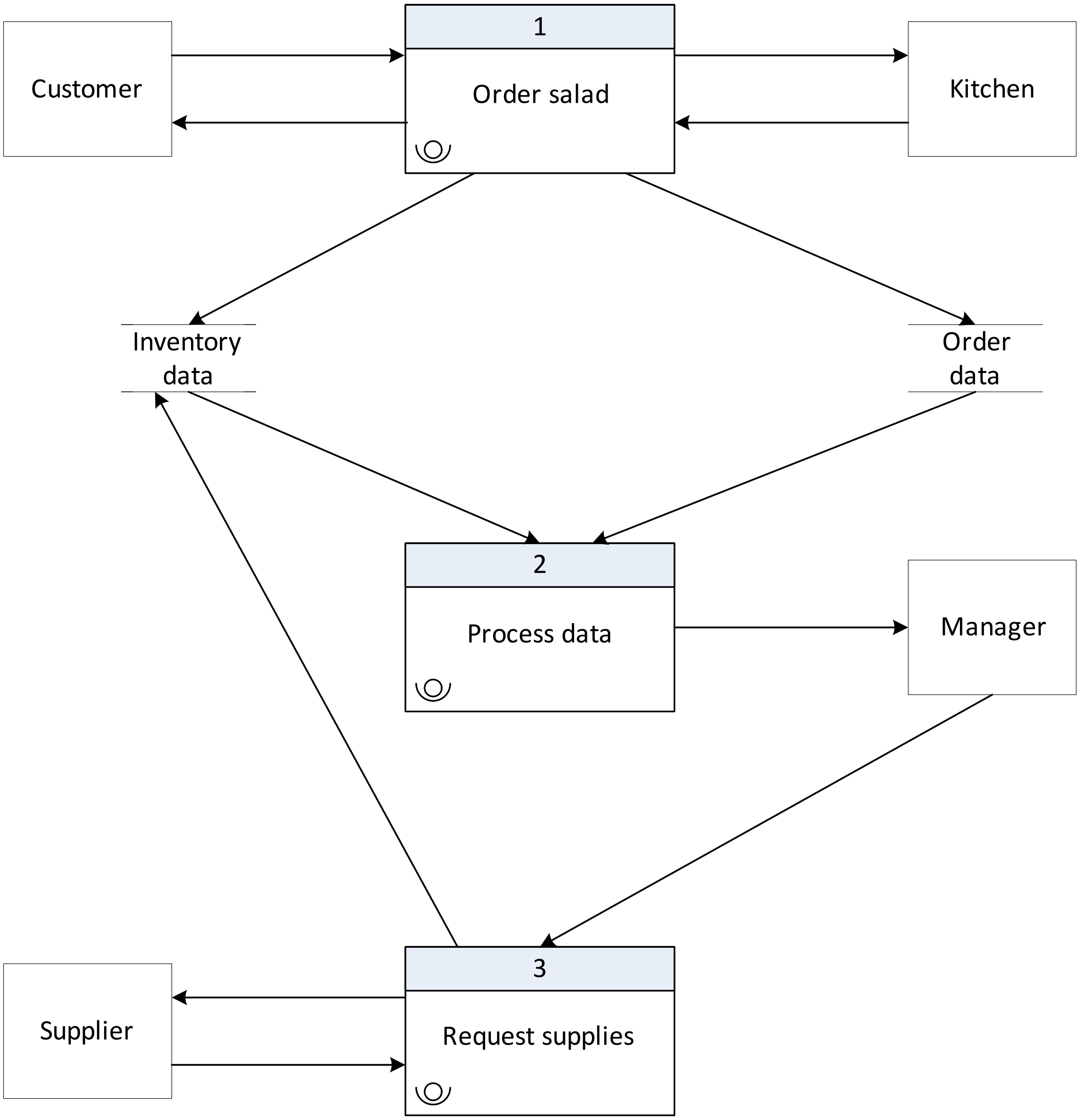


Figure 18. Level 1 DFD for Food Ordering System

As observed in Figure 18 a Customer places an Order for a salad. The Order salad process sends the data to the Kitchen, updates the Inventory data store and keeps new data in Order data store. Likewise, the Kitchen after preparing the requested order will update the Order data process, which is responsible for transferring a receipt back to the Customer. The Order and inventory data are then processed and can be used in the creation of reports to keep the Manager process informed. The Manager process is to order new supplies. The Request Supplies process forwards the order to the Supplier and updates the Inventory's data store. Finally, after the new supplies are sent, the Request supplies process updates the Inventory accordingly.

5.2. Siemens use case

In the Siemens's use case, the PERFoRM system will be implemented to the Duisburg factory, which is the head-quarter of the Business Unit Compression. The Duisburg factory is responsible for the manufacturing of tailored compressors trains for oil and gas applications, such as air separation units or for the Liquefied Natural Gas (LNG) production. At the moment, production of compressors is characterised by the production of small lot sizes (1-30), machining, manual labour and a high complexity in assembly, as a high number of individual parts have to be combined to assemble the final product. It is expected that each failure or machine breakdown can lead to delays of the productions and missing parts to semi-finished products.

5.2.1 Level 0

The *Level 0* DFD (or Context Diagram) for the Siemens's *Production System* is illustrated by Figure 19. The "*Production System*" process represents the system to be modelled. Figure 19 also depicts the participants who will interact with the system, referred to as the external entities: *Customer, External Maintenance, and Shop floor*. In between the process and the external entities, there are data flows that indicate the existence of information exchange between the entities and the system.

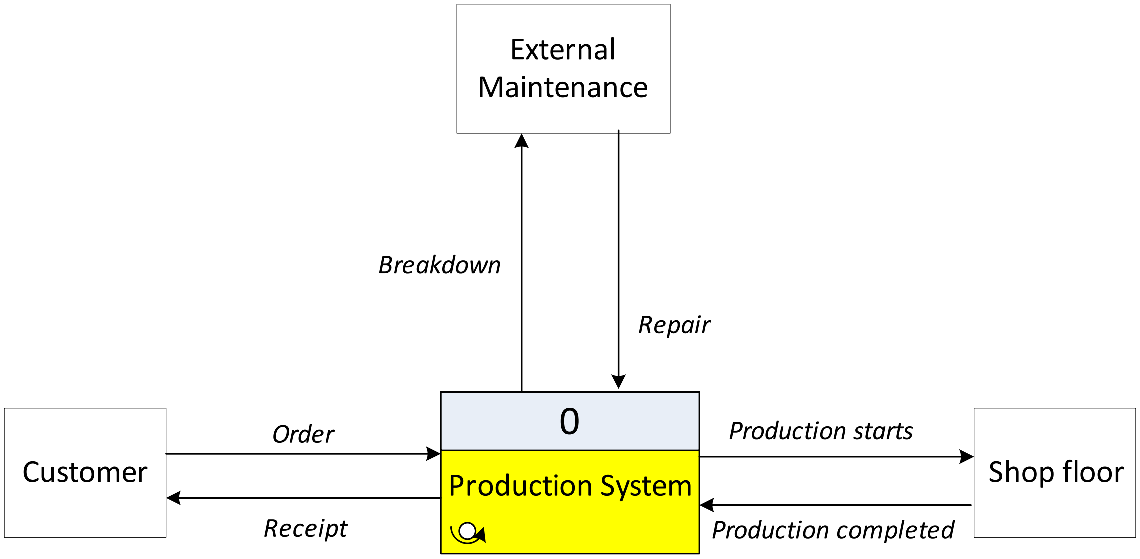


Figure 19. Siemens's Level 0 Data Flow Diagram

The *Level 0* DFD provides a general overview of the system. It can contain only one process and the external entities that interact with it. Data stores cannot be depicted in the *Context diagram*. (Donald S. and Le Vie, 2000) In the above diagram, it can be observed that a *Customer* can place an order. The *Production System process (0)* has to schedule and transfer the information for the production to the *Shop Floor,* upon receiving the order. In case a breakdown of a machine cannot be repaired from the internal maintenance department, *Production System* has to inform the *External Maintenance*. After the production is completed, the customer will receive a receipt.

5.2.2 Level 1

Figure 20 depicts the *Level 1 DFD* of the *Production System* process. The *Level 1* diagram contains are: (1) seven processes (*ERP (SAP), Data process, Maintenance System, Scheduling System (External), Data process, CNC Machine (840D Controller) and Ticket System on the Shop Floor)*), (2) one external entity (*Customer*) and (3) seven data stores (*Production Orders, Maintenance Orders, Oracle DB, Production Schedule, Breakdown Prediction, LH Net Data (SQL) and OEE Data*).

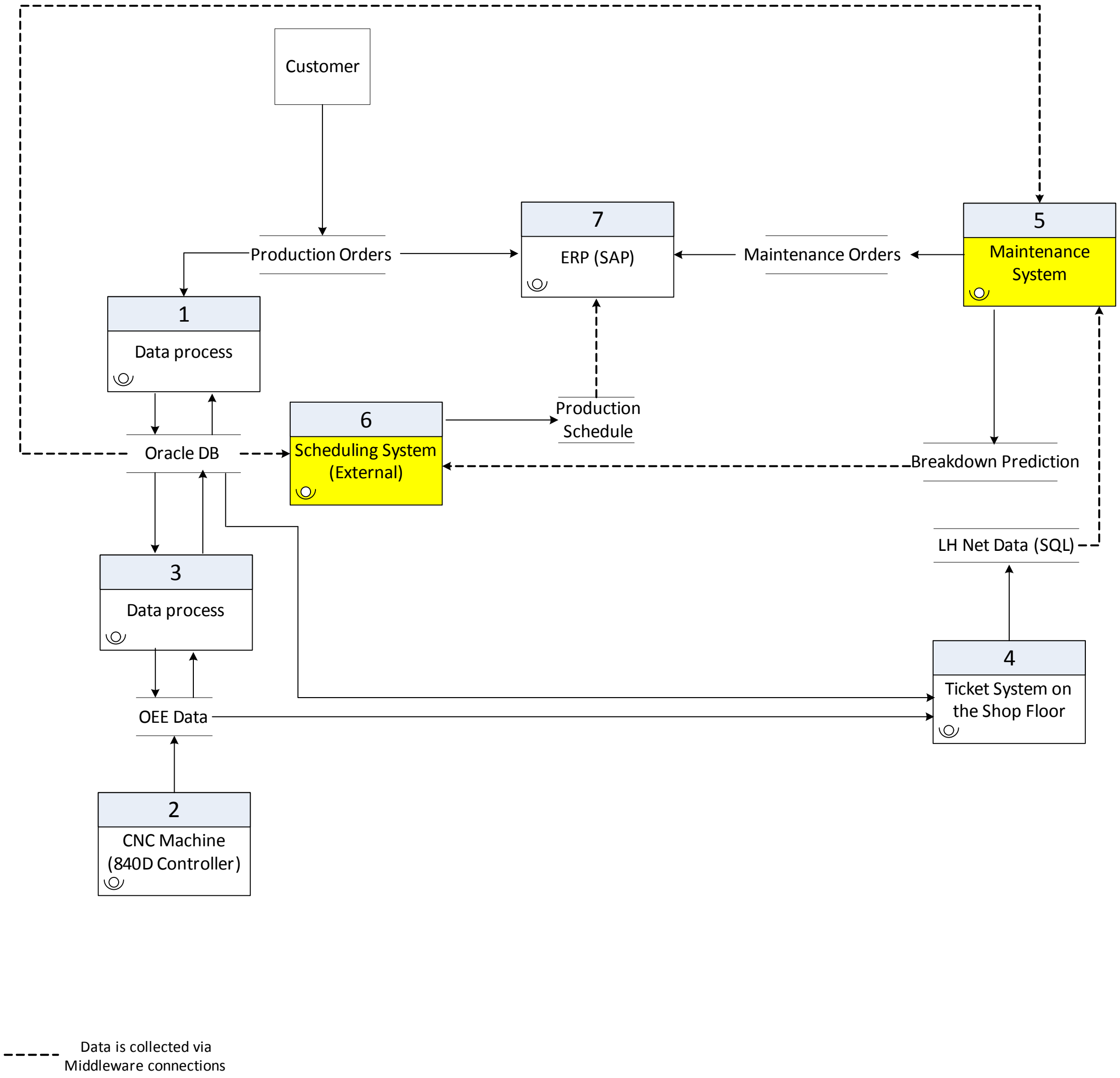


Figure 20. Siemens's Level 1 Data Flow Diagram

When a *Customer* places an order, the data will be stored in the Production Orders data store and from there will be forwarded through the *Data process (1)* to the *Oracle DB* data store. Moreover, all the data from the *CNC Machine (2),* which has particular an 840D Controller, will be stored to the *OEE Data* and after the suitable transformation (i.e. a data conversion between the data types) *(3)* will be forwarded to the *Oracle DB* as well.

Every time that a machine in the shop floor has a breakdown, the related to the failure data, from the *OEE Data* and the *Oracle DB* data stores, will be transferred to the *Ticket System on the Shop Floor process (7).* A ticket will be created automatically and will store all the information that are related to the specific failure. *Maintenance System (5)* will take as input the data from the *Oracle DB* and from the *LH Net Data Store*, which is an *SQL database* that stores all the data from the tickets that have been created via the *Ticket System* on the *Shop Floor process (4).*

As seen in Section 3.1, the two main objectives of Siemens use case are: (1) to implement a predictive maintenance system and (2) to integrate it with the scheduling system. The *Maintenance System* process *(5)* after processing the related information, will store the updated data to the *Maintenance Orders* and to the *Breakdown Prediction* data stores accordingly. The *Scheduling System*

(6) will then take the data from the *Oracle DB* and from the *Breakdown Prediction* data stores, in order to schedule the production. The final schedule of the production data will be stored to a database which is called *Production Schedule*.

Finally, the *ERP system (7)* will use the data from the *Production Orders, Maintenance Orders and Production Schedule* in order to finalise the plan of the production. Further details on the *Maintenance System* process *(5)* and the *Scheduling System (6)* are represented via the creation of Level 2 diagrams (see Section 5.2.3).

5.2.3 Level 2

The interaction between the system and the *Middleware* can be seen in the *Level 2 Diagram* (see Figure 21), which contains: (1) 14 processes (the *ERP (SAP), Data process, Data process, Maintenance System, Scheduling System (External), Data process, CNC Machine (840D Controller), Middleware,* and six *Data translation processes)*, (2) one external entity (*Customer*) and (3) seven data stores (*Production Orders, Maintenance Orders, Oracle DB, Production Schedule, Breakdown Prediction, LH Net Data (SQL) and OEE Data*). It is to be noted that technology adaptors (developed within WP3) have been referred to as *Data translation process* in all *Level 2* diagrams the use cases.

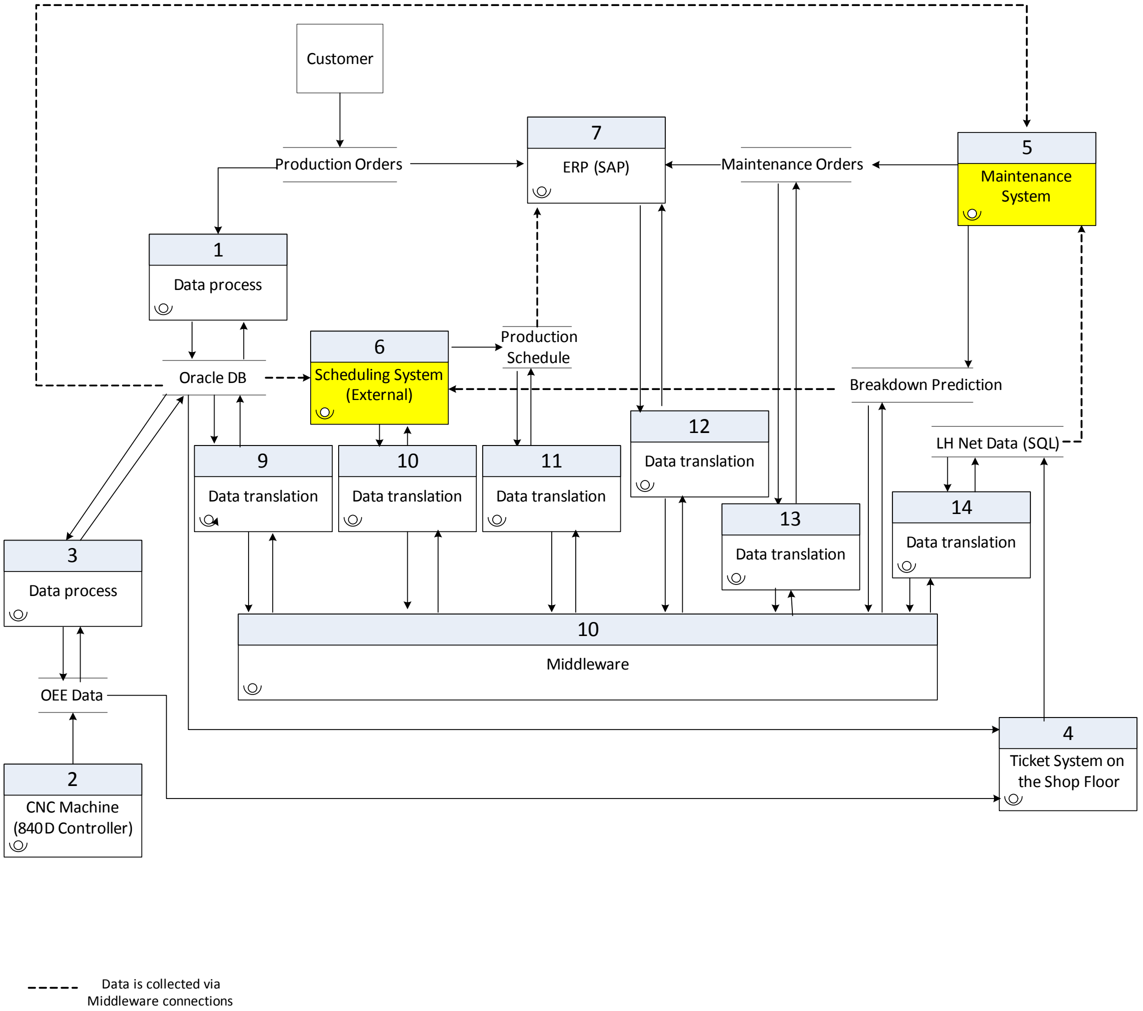
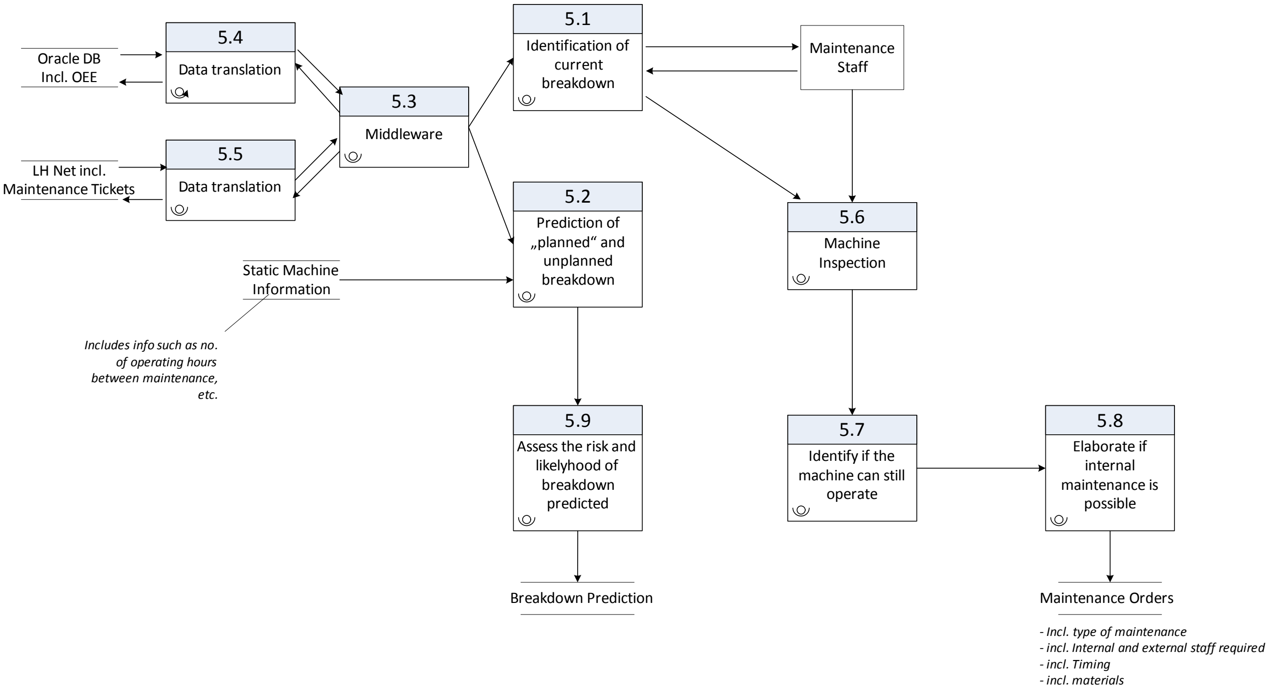


Figure 21. Siemens's Level 2 Data Flow Diagram

Within the PERFoRM project, the Siemens use case will be introducing the middleware for routing the communication between the different entities within the system. Figure 21 in particular depicts the introduction of the *Middleware* to the *Level 1* diagram (see Figure 20). The main difference between Figures 19 and 20 is the routing of information via *Middleware (8)* and Figure 21 depicts the to-be architecture. Each entity will interact through a technology adaptor (referred to as the *Data Translation* process referred using numbers *(9-14)* in Figure 21) with the *Middleware (8)* in order to allow the exchange of the data. The *Oracle DB*, *Production Schedule, Maintenance Orders and LH Net Data storage,* as well as *Scheduling System (6)* and *ERP system (7)* will connect with the *Middleware (8)* via a *Data translation processes (9- 14)*. The *Data translation process (9 -14)* is not needed in case of the *Breakdown Prediction* data store, as the data can be sent directly to the *Middleware,* without the need for a technology adaptor. This is because the *Breakdown Prediction* data store will be implemented as a part of the PERFoRM project and therefore the data model will conform to the PERFoRM data model. The *Maintenance system* (process *(5)* in Figure 21) and the *Scheduling system* (see process *(6)* in Figure 21) processes have been further decomposed within the following sub-sections.

5.2.3.1 Level 2 – Maintenance system

The *Maintenance System process* is analysed in this section and illustrated in Figure 22. It consists of: (1) nine more processes (*Identification of current breakdown, Prediction of "planned" and "unplanned" breakdown, Data translation, Data translation, Middleware, Machine Inspection, Identify if the machine can still operate, Elaborate if internal maintenance is possible*), (2) one external entity (*Maintenance staff*) and (3) five data stores (such as *Oracle DB, LH Net, Static Machine information, Breakdown Prediction and Maintenance Orders*).


Oracle DB
Incl. OEE
5.4
Data translation
LH Net incl.
Maintenance Tickets
5.5
Data translation
5.3
Middleware
5.1
Identification of current breakdown
Maintenance Staff
5.2
Prediction of „planned" and unplanned breakdown
Static Machine Information
5.6
Machine Inspection
5.9
Assess the risk and likelihood of breakdown predicted
5.7
Identify if the machine can still operate
5.8
Elaborate if internal maintenance is possible
Breakdown Prediction
Maintenance Orders

Includes info such as no. of operating hours between maintenance, etc.

- Incl. type of maintenance
- incl. Internal and external staff required
- incl. Timing
- incl. materials

Figure 22. Siemens's Level 2 - Maintenace system Data Flow Diagram

The *Maintenance System* process internally consists of two main processes: (1) *Identification of current breakdown (5.1)* and (2) *Prediction of planned and unplanned breakdown (5.2)*. The input for both processes come from the *Oracle Database* (which includes the data from CNC Machine) and the *LH Net* data store (which includes the data from the Maintenance Tickets). The data will be transformed through special adaptors (*Data translation* processes, referred using numbers *5.4* and *5.5* in Figure 22) and then transferred to the *Middleware (5.3)*. *Identification of current breakdown (5.1)* and *Prediction of "planned" and "unplanned" breakdown (5.2)* will take as input the output of the *Middleware (5.3)*. *Identification of current breakdown* process *(5.1)* is responsible for sending the information to the *Maintenance Staff* and to the *Machine Inspection* process *(5.6)*. Depending on the information that the ticket carries, the system will have to decide whether the inspection of the failure can be done automatically or the maintenance staff need to be involved. If the data are not enough to identify the reason of the breakdown, process number *(5.1)* (*Identification of current breakdown*) will notify the maintenance staff in order to examine manually the machine that has failed. The next step involves the identification whether the machine is still capable of operating. Finally, based on the type of the failure, the system will evaluate if the internal Maintenance staff is able to fix the machine or external maintenance staff need to be brought in. All the information that are related to the maintenance (such as the type of the maintenance, the time and the materials that were used and the staff that repaired the machine) will be stored in the *Maintenance Orders* data store.

5.2.3.2 Level 2 – Scheduling system

The *Scheduling System* is responsible for the optimisation of the production plan and schedule. The *Level 2 Scheduling system* diagram contains (1) five more processes (*Data translation, Middleware, Simulation of What – If Scenarios, Calculation of Scenario KPI's, Select production schedule)*, (2) one external entity (*Shift Leader)* and (3) four data stores (*Oracle DB, Breakdown Prediction, Maintenance Orders and Production Schedule).*

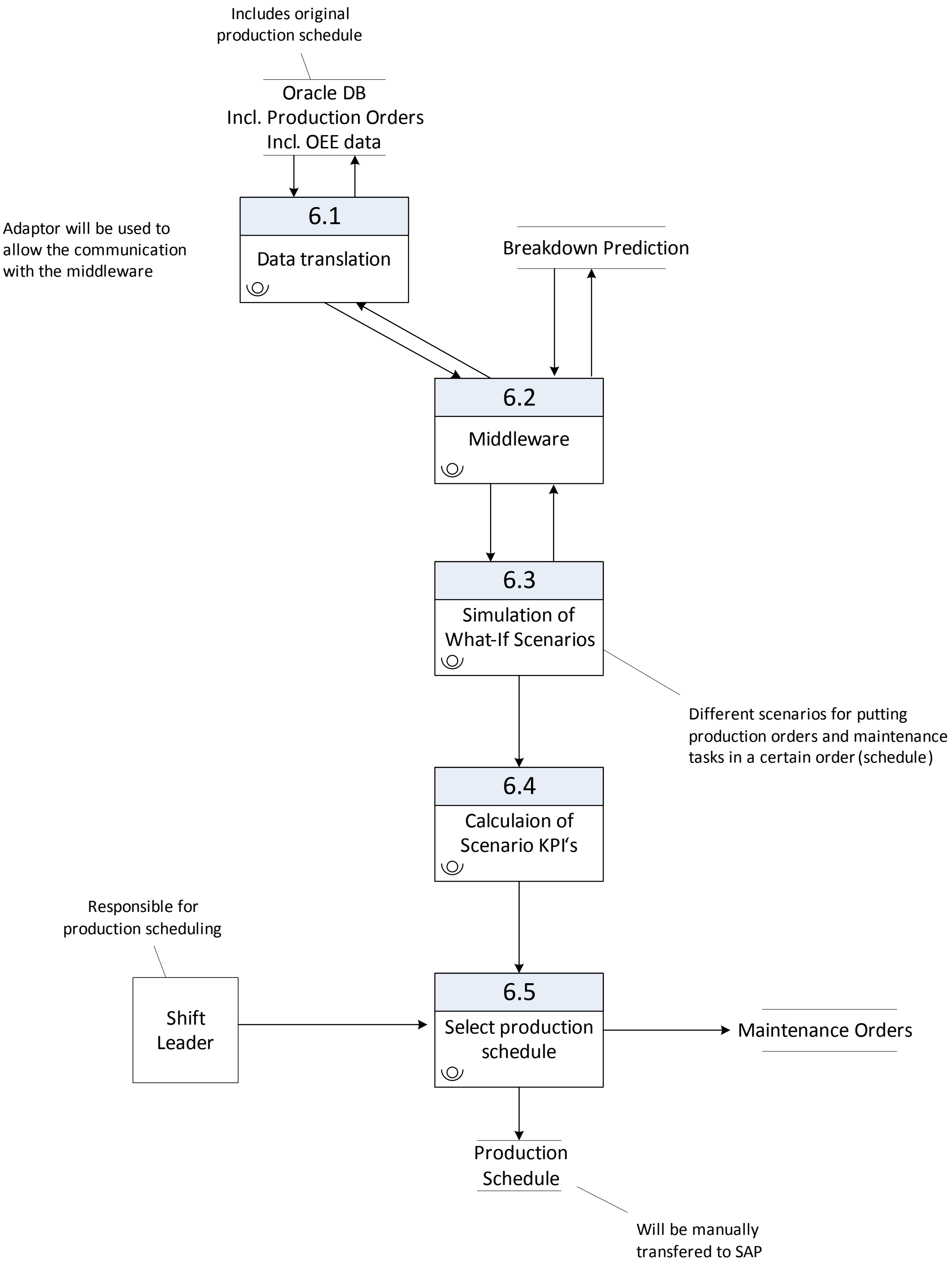


Figure 23. Siemens's Level 2 – Scheduling process Data Flow Diagram

As seen in Figure 23, all the data from the *Oracle DB* and the *Breakdown Prediction* data stores will be transferred through the *Middleware (6.2)* to the *Simulation of What – If Scenarios* process *(6.3)*. The data from the *Oracle Database* will be first transformed through a *Data translation* process *(6.1)* and then sent to the *Middleware.* Different production scenarios will be simulated in order to prioritise and schedule orders and maintenance tasks within the *Simulation of What – If Scenarios* process *(6.3)*. All this information will be transferred to the process number *(6.4)* (*Calculation of Scenario KPI's*), where the KPIs of each possible scenario will be calculated. The *Select production schedule* process, will combine the data from process *(6.4)* and from the *Shift Leader* external entity and will create the final production schedule. The final schedule will be stored in the *Production Schedule* data store and will also be sent to the Maintenance Orders data store, as it will be used for defining the suitable time slots that each machine's maintenance can take place.

5.3. IEFVS use case

The E-district's use case focuses on the production of different Micro Electric Vehicle architectures. The main goal is to achieve a flexible and agile manufacturing system that allows the assembly of high quality customised vehicles e.g. small lots of specific vehicles to larger lots of passenger cars, using low budget assembly lines. With the implementation of the PERFoRM system, production will achieve a higher degree of automation. The most critical operations, such as welding, will be performed by robots to improve the quality, as well as the efficiency and reproducibility of the processes.

5.3.1 Level 0

Produce cars System represents the E-district's *Level 0* DFD and can be seen in Figure 24. This diagram represents the "Produce cars" process and all the external entities that will interact with the system: *Warehouse*, *Shop floor, Customers* and *Suppliers*.

Figure 24. E-district's Level 0 Data Flow Diagram

As seen in Figure 24, the *Customers* entity can place orders for cars. The *Produce cars* system has to verify if all the subcomponents that are essential for the production of the order are available at that time in the *Warehouse.* If certain subcomponents are missing, the *Produce cars* system has to contact *Suppliers* and place orders accordingly. After receiving the components, the production of the vehicle can start in the *Shop floor*. After the production is complete, the *Customer* will receive the receipt.

5.3.2 Level 1

The *Produce cars* System process has been decomposed to represent the *Level 1 Data Flow Diagram* (as seen in Figure 25). This diagram contains: (1) five processes (*Order cars, Production planning, Data processing, Forecast, Monitoring*), (2) three external entities (*Customer, Shop floor, Purchasing office*) and (3) seven data stores (*Suppliers cloud data, Warehouse cloud data, Orders cloud data, Customers cloud data, Traceability's cloud data, Production cloud data, Forecasting cloud data*).

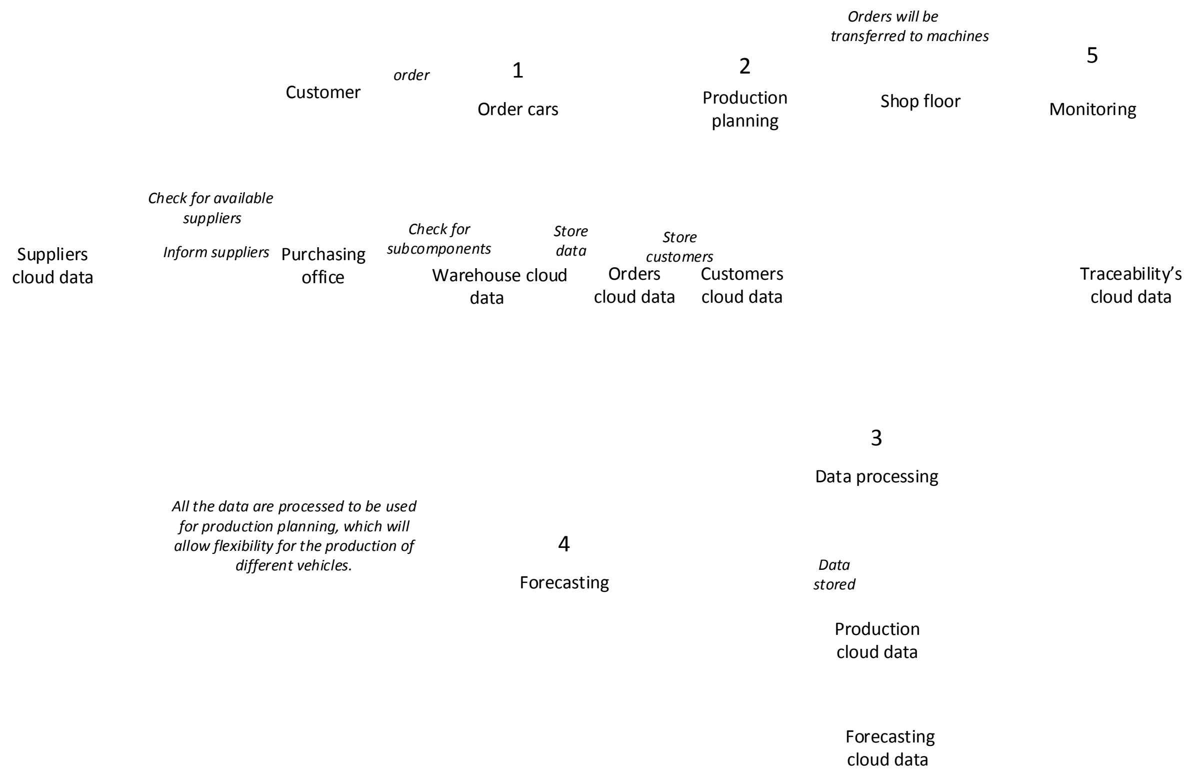


Figure 25. E-district's Level 1 Data Flow Diagram

As seen in Figure 25, the *Order cars* process (process number *(1)* in Figure 25): (1) receives the *Order*, (2) stores the updated details in the *Orders* and *Customers data stores* (which are part of the cloud system of the company), (3) checks for the available subcomponents in the *Warehouse data store* and (4) informs the *Purchasing office* on the subcomponents that are not available. The *Purchasing office*, then, has to find a suitable suppliers from the *Suppliers data store* and inform them on the items needed. Suppliers will send the receipt for the requested equipment to the company. Moreover, the *Order cars* process *(1)* has to transfer the *Order* to the *Production planning* process *(2)*. The production of the cars is represented by the *Shop floor* entity. All the data from the *Production planning (2)* and from the PLC's and Robots (that will be used during the production), will be transferred to the *Data processing* system *(3)* and will be stored in the *Production data store*, within the company's cloud architecture. The *Forecasting* process *(4)*, which takes *Production, Orders, Customers, Suppliers* and *Warehouse cloud data* as input, will be used for planning, to allow flexibility in the production of different types of vehicles. Finally, the *Forecasting* data will be stored in a data store in the cloud and in combination with the data that will be received from the shop floor, will be used in the traceability system during the *Monitoring process (5)*. All the data from the *Monitoring (5)* will be stored in the *Traceability cloud data store*.

5.3.3 Level 2

5.3.3.1 Production process

The *Production planning* process (2) as seen in Figure 25 will be decomposed via a Level 2 diagram within this section. The diagram below (see Figure 26), which represents the PERFoRM's architecture, contains seven processes (*Scheduling system, Data translation, Middleware, Data* *translation, Control data, Control data and Data translation*), five external entities (*PLC,* Robot *Controller, HMI, R&D and Maintenance*), as well as one data store (in this case the data store is the whole cloud service of the company, which is considered as a storage unit, where all the data that are related to the production are collected).

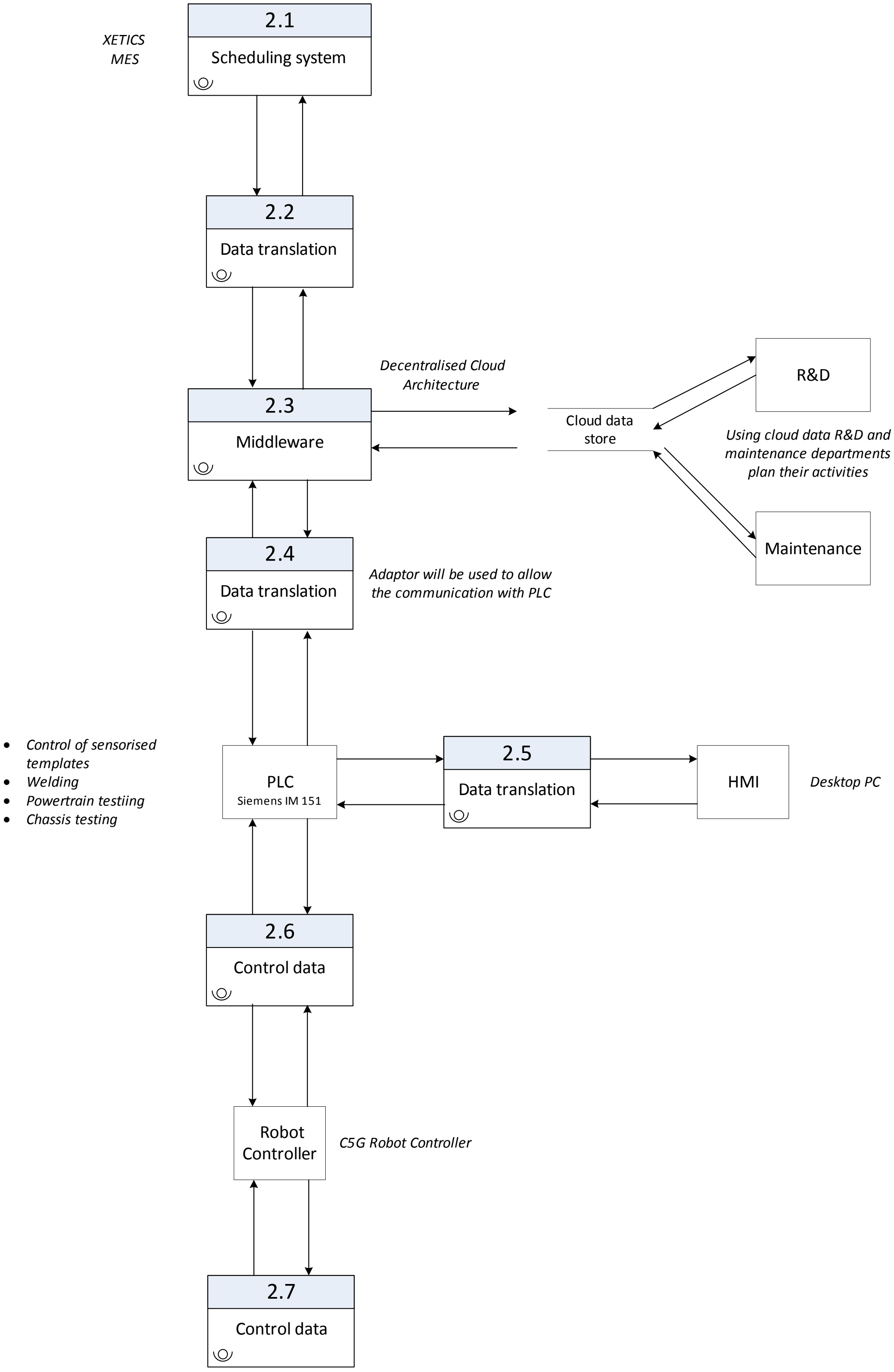


Figure 26. E-district's Level 2 Data Flow Diagram-Production planning Process

The first process that will need to run during the *Production Planning (2),* is the *Scheduling system* process *(2.1)*. The Xetics MES system will analyse all the data from the orders and process them, through an adaptor (*Data translation* process, referred with number *2.2*, in Figure 26), to the Middleware *(2.3)*. The Middleware *(2.3)* will interact with the decentralised cloud architecture, which will update the data stores if needed. The data saved in the cloud data storage will be used by the: (1) Research and development (R&D) department for the continuous updating and improvement of the system and (2) the maintenance department for predicting maintenance operations. The Middleware *(2.3)* will also forward all the data to the Siemens IM-131 PLC, which is responsible for the control of the sensorised templates. Each template is a welding mask, which specifies the position of the metal tubes. Special sensors, which have been placed across those templates, will be responsible for checking if every welding part has been placed in the right position. Welding process will only start after the check from the sensors is finished and successful. The PLC can be controlled by a HMI, which in this case is a desktop PC. After the welding operation is completed, the data will be transferred to the *Control data process (2.6)* and then the essential transformation will be forwarded to the C5G *Robot Controller*. Finally, the data will be processed for the final *Control (2.7).*

5.3.3.2 Monitoring process

During the *Monitoring* process (process number *(5),* in Figure 25), vehicles will be monitored, so possible problems could be identified at early stages of production. The *Level 2* Diagram for the *Monitoring* process contains five processes (*Scheduling system*, *Product traceability, Middleware, Data translation and Control data*), two external entities (*PLC* and *User panel*) and one data store (*Vehicles cloud data*), as illustrated by Figure 27.

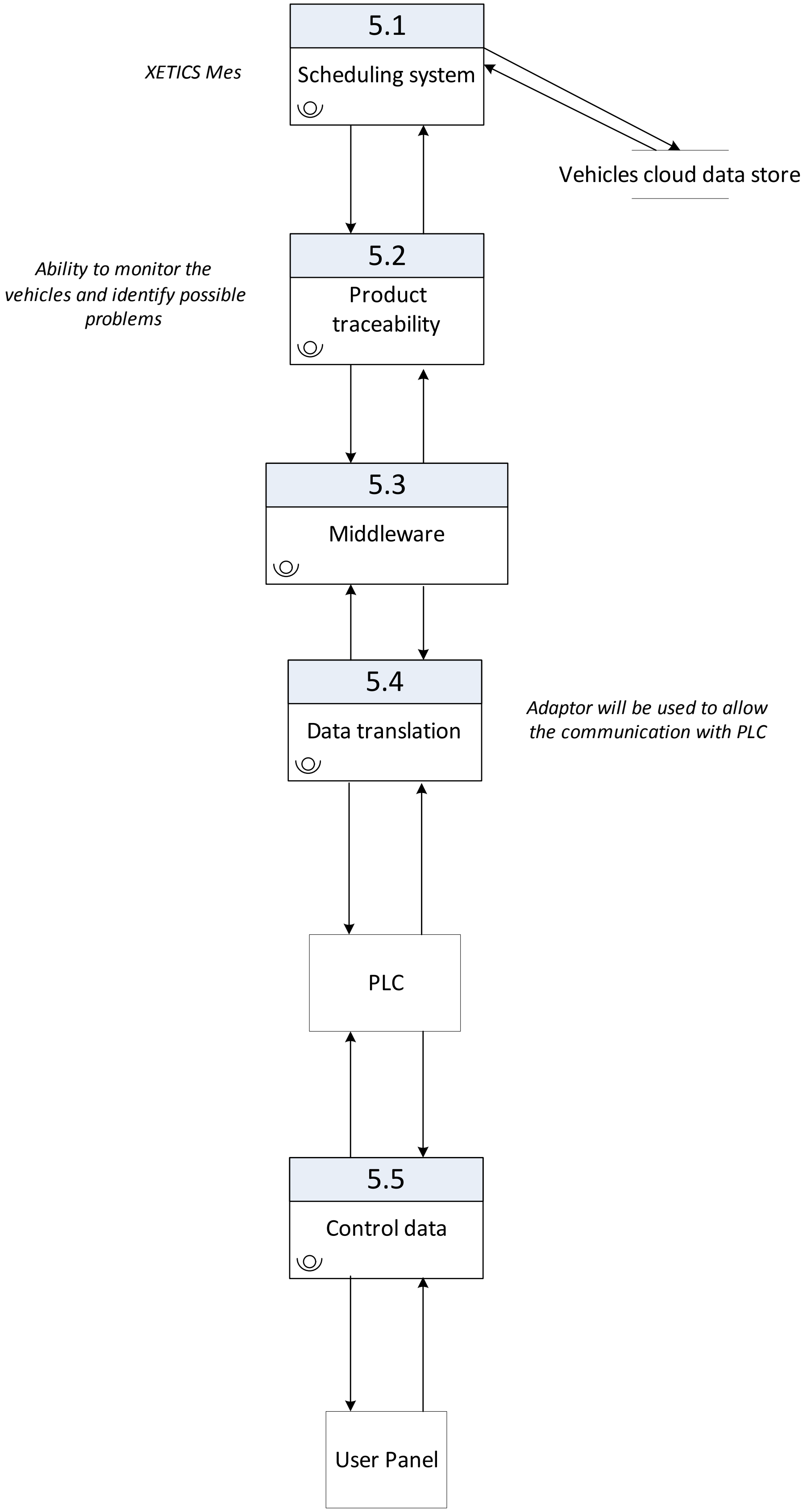

5.1
Scheduling system
XETICS Mes
Vehicles cloud data store
5.2
Product traceability
Ability to monitor the vehicles and identify possible problems
5.3
Middleware
5.4
Data translation
Adaptor will be used to allow the communication with PLC
PLC
5.5
Control data
User Panel

Figure 27. E-district's Level 2 Data Flow Diagram-Monitoring Process

The *Monitoring process (*referred as process number *(5)* in Figure 25) will allow the identification of possible problems at early stages of production. All the materials that have been used during the welding phases, will be associated to a unique ID. This ID will correspond to a specific product and its supplier. A *User Panel* (possibly an HMI) will control the *PLC* and keep record of all the materials that have been used during the welding phase. When a problem will be identified all that data will be sent to the middleware and from there to *Product traceability (5.2)*. The scheduling system will take as input the data from process number *(5.2)* and will update the *Vehicles data store,* which is stored in the decentralised cloud architecture of the company.

5.4. Whirlpool use case

Whirlpool's use case focusses on the development of a Reconfiguration support system, which combines a monitoring and simulation system that will be able to work at very high level of abstraction, in order to improve the current reconfiguration process. This new reconfiguration system will help to achieve a closer and faster correlation between actions and expected performance of the whole system. Whirlpool will improve the respond to customer demands and will also be able to adapt more efficiently to the new technologies. The overall factory macro-process has been divided into most important sub-processes and a further analysis allowed the selection of three most significant use case scenarios: Leakage test at cell level, Cavity Fabrication and Value Stream at department level. Performing an automatic leakage test will be the first objective out of this project (Robotic Cell Reconfiguration), while Reconfigurability of the Cavity Fabrication and Value Stream is the second (Cavity Fabrication and Value Stream Reconfiguration). PERFoRM experimentation and demonstration activities will be used in the microwave factory, which is located in Biandronno.

5.4.1 Level 0

Whirlpool's *Integrated Supply Chain System*, which is illustrated by Figure 28 displays the *Level 0 DFD*. The "*Integrated Supply Chain System*" process interacts with *Logistics*, *Warehouse*, *Shop floor, Suppliers and Customers*. Data flows indicate the existence of information exchange between the entities and the system.

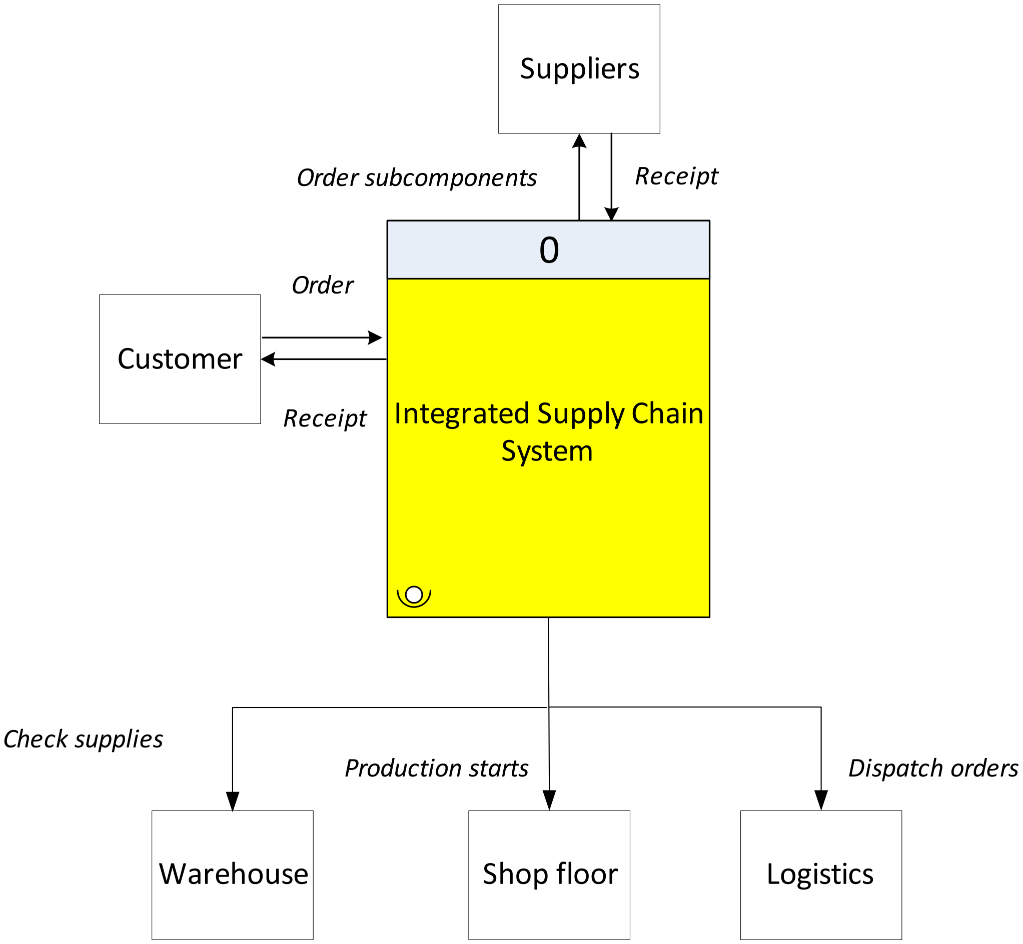


Figure 28. Whirlpool Level 0 Data Flow Diagram

When a *Customer* places an order, the *Integrated Supply Chain System (0)* will check the availability of the subcomponents in the *Warehouse*. In case that the essential subcomonents are not available, the *Integrated Supply Chain System* will make a request to the *Suppliers*. The next step is to transfer the instructions for the order to the *Shop floor*. Finally, *Logistics* will deal with the dispatching instructions of the the requested product.

5.4.2 Level 1

The *Level 1 DFD* of the *Integrated Supply Chain System* process can be seen below (Figure 29). Figure 29 contains: (1) six processes (*Ordering System, Scheduling Systems, Supervisory System, PLM System, Reconfiguration System and Warehouse data processing System*) and (2) two data stores (*Orders* and *Warehouse*).

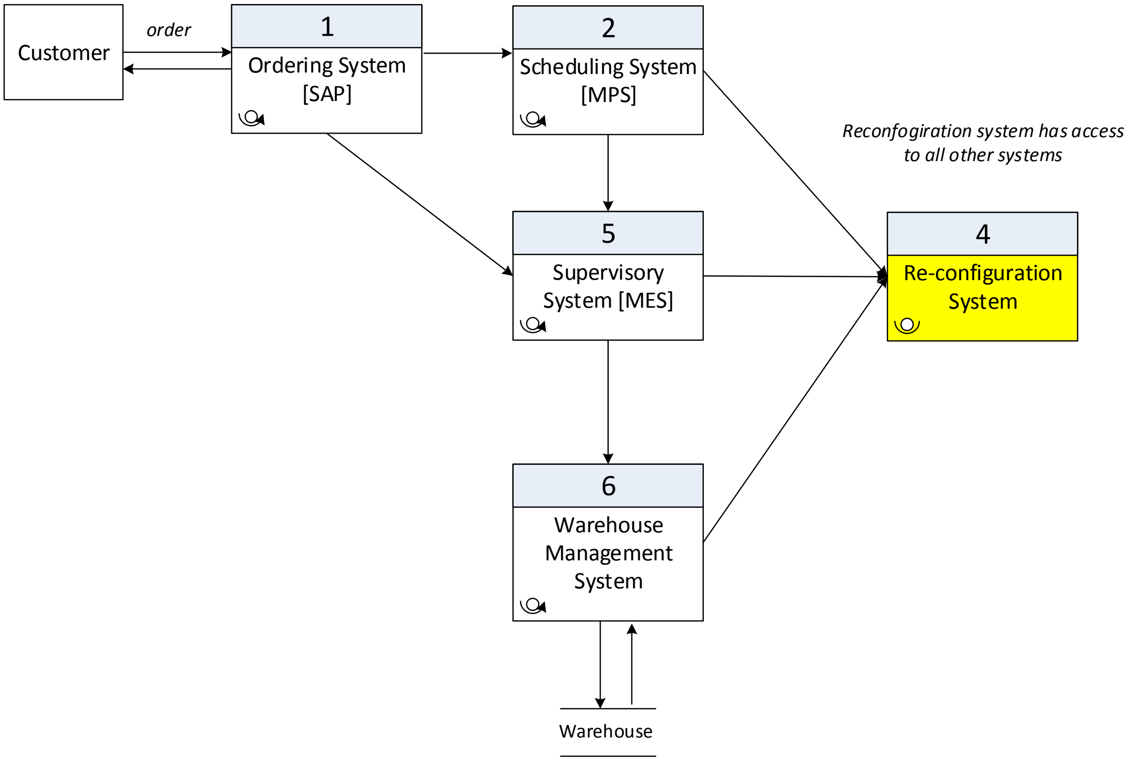


Figure 29. Whirlpool Level 1 Data Flow Diagram

SAP (process number *(1)* in Figure 29) receives the orders and supervises the production process of the orders with two main functions: Master Production Schedule and MES. The *MPS (2)* sends a 5 day frozen plan of the production to the *MES system* and *MES* subsequently interacts with the *Warehouse Management System (6)* in order to increase the available stock and decrease the component and material stock. *Warehouse Management System* will then update *Warehouse data store* accordingly. The Reconfiguration system, in order to achieve a high level of abstraction, which will allow the overall factory to be considered as the central process, will interact with all the active information systems of the factory like the *Scheduling system*, the *Supervisory System* and the *Warehouse Management System*.

5.4.3 Level 2

In this section, the *Level 2* diagram *Re-configuration System (*process number *(4),* in Figure 29*)* will be further analysed. The system will be separated in two different use cases, based on Whirlpool's two objectives.

5.4.3.1 Level 2- Establishing a reconfiguration System: Objective 1

The below diagram, is the extension of "Re-configuration System", which has been designed based on the Objective 1: *Establishing a Reconfiguration System*. This diagram contains five more processes (*PLM, Robot re-configuration tool, Middleware, Data translation and Control execution*), two external entities (*Robot and Robotic controller*), as well as one data store (*Robotic Programs*).

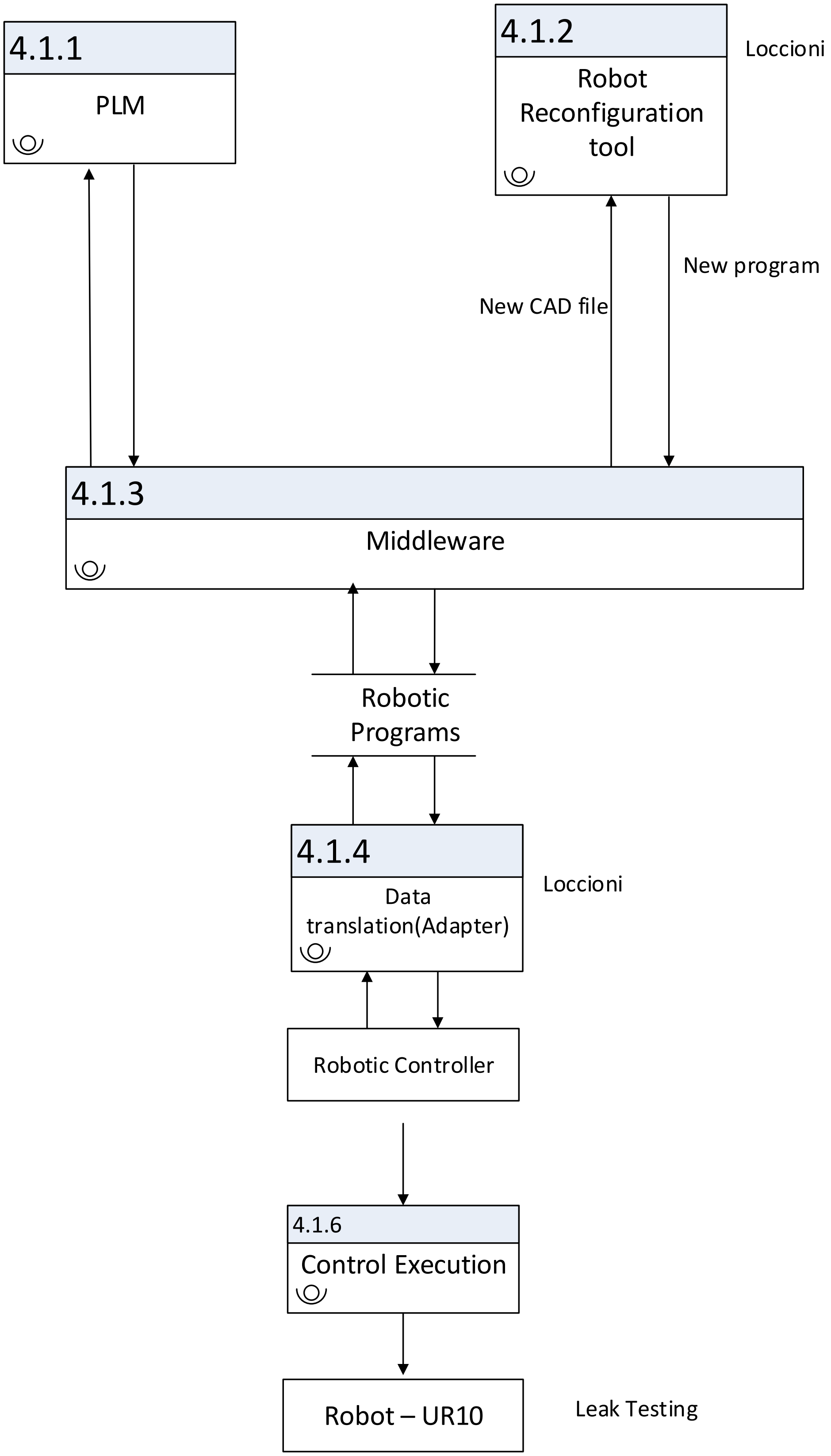


Figure 30. Whirlpool's Level 2 Data Flow Diagram – Objective 1

The goal of this objective is to stablish an independent robotic system to test leakages. The system consists of a Universal robot (UR10) robot, equipped with a microwave probe. Instead of a direct connection between the *PLM (4.1.1)* and the rest of the system, a CAD file (which has been generated on design level) is being transferred to the *Robot Reconfiguration tool (process (4.1.2))* (see Figure 30). This CAD file contains all the information for the probe path. *Robot Reconfiguration tool (4.1.2)* will translate the CAD file to a robot path program and transfer this new program to the *Robotic Programs data store* via the *Middleware (4.1.3)*. From there the suitable program will be transferred to the *Robotic controller* via an *Adaptor* (process *Data translation (4.1.4),* see Figure 30). Data will be processed for *Control Execution (4.1.6)* and will finally sent to UR 10 for the requested Leak testing.

5.4.3.2 Level 2- Cavity Fabrication and Value Stream Reconfigurability: Objective 2

The second diagram, which presents the Objective 2: Cavity Fabrication and Value Stream Reconfigurability, contains four processes (*Data Integration System, Simulation, Monitoring and Middleware*) and one data store (*Perform Database*).

The diagram below (see Figure 31) illustrates the interaction between the *Data Integration System (4.2.1)* and the *Reconfiguration System (4.2.2), (4.2.3)* via Middleware (4.2.4). All the data from OEE, Andon, DCS and F-test are being pulled to the *Data Integration System* in order to be integrated and transformed. The Data Integration System ensures that all data in the factory is accessible and right formatted. This new data will be stored on PERFoRM database in order to be utilised by Monitoring and Simulation systems. Monitoring system will visualise the dynamic real-time data. It will represent the dynamic behaviour of factory in a high level and generate link a between the KBFs and KPIs. Once this new visualisation system been tested, it will be consequently used to implement the Simulation system.

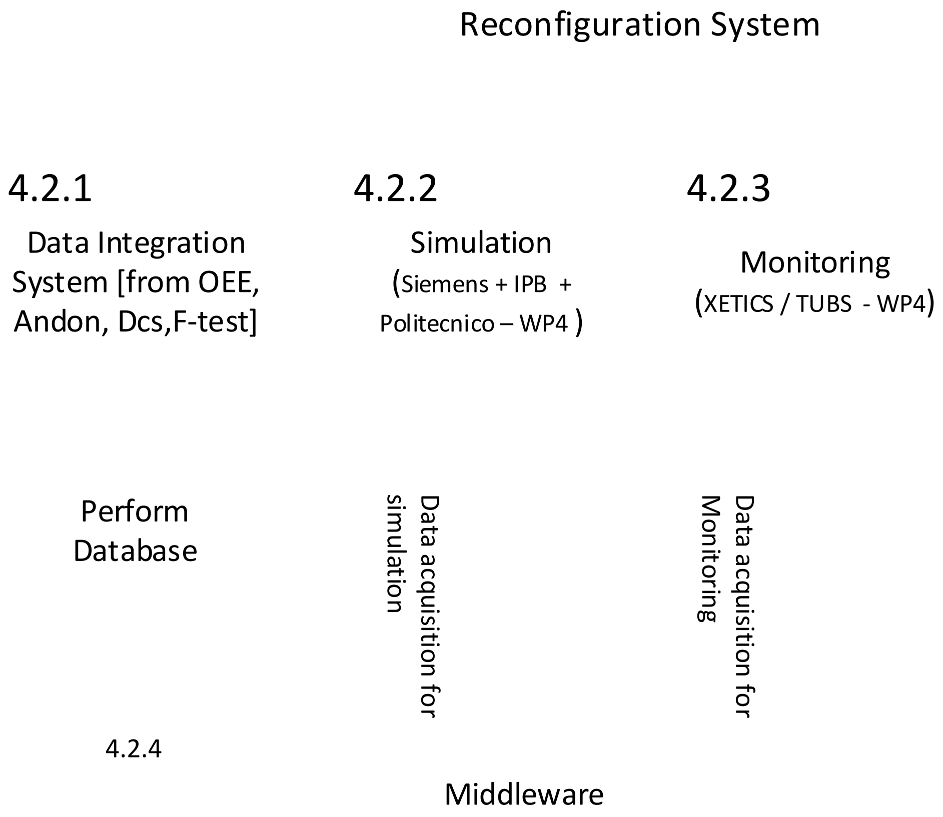

Reconfiguration System
4.2.1
Data Integration System [from OEE, Andon, Dcs,F-test]
4.2.2
Simulation (Siemens + IPB + Politecnico – WP4)
4.2.3
Monitoring (XETICS / TUBS - WP4)
Perform Database
Data acquisition for simulation
Data acquisition for Monitoring
4.2.4
Middleware

Figure 31. Whirlpool Level 2 Data Flow Diagram – objective 2

5.5. GKN use case

GKN produces high value jet engine components with high stringent quality. Advanced material and processes are being used to produce these components in order to meet high quality parameters. GKN's main aim in the PERFORM project, is to demonstrate a high agile production process using a new integrated system. This new system should be able to complete a sequence of the operations in the value adding process chain. Achieving this new system, demands a construction of a reconfigurable robotic cell that optimises productivity and reduces change over time.

5.5.1 Level 0

The below diagram (see Figure 32), represents the *Level 0* of GKN's system. The main process is the *Production System* process and interacts with five external entities (*Suppliers, Customer, Product Engineering, Logistics/ Transports, Supply Chain & Warehouse*).

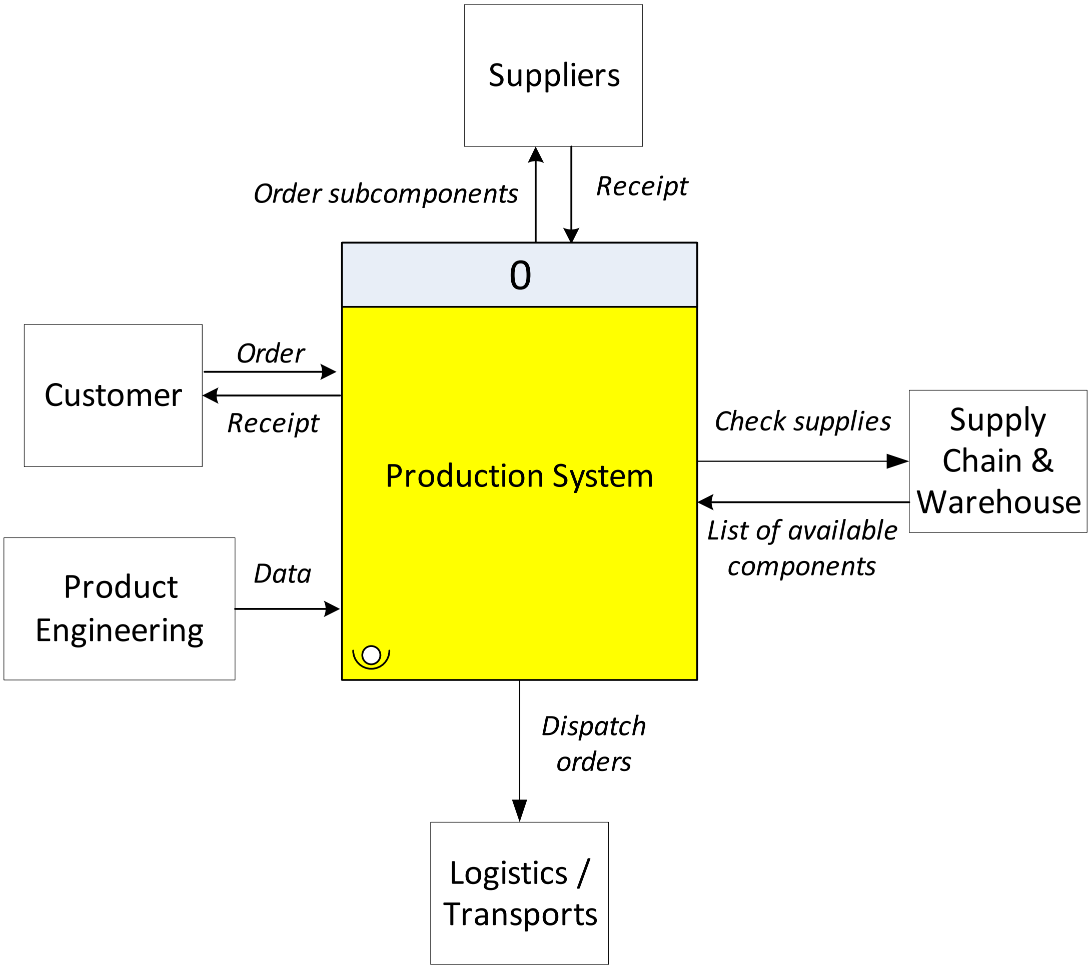


Figure 32. GKN Level 0 Data Flow Diagram

When a *Customer* places an order, *Production System (0)* is responsible for checking the *Warehouse* for the available subcomponents. The process has to identify if there is need for new supplies, and communicate with the suitable *Suppliers*. *Product Engineering* entity will also forward all the necessary data for the production. Finally, after the production is finished, *Production System* will notify the *Logistics/ Transports* department in order to deliver the product to the *Customer*.

5.5.2 Level 1

The *Level 1* DFD of the GKN's Production System can be seen in Figure 33. It contains: (1) thirteen processes (*Ordering System (SAP), Production and Capacity Planning, Material Requirements Planning (MRP), Production Scheduling, ERP, PLM (Team Center Manuf.), CAM/ NC (Siemens NX), Robot program (Process Simulate), Operation documents/instructions, Equipment, tools and fixtures design, Quality/SPS (QSYS), Stop time/OEE (AXOS), Production/Execution*), (2) five external entities (*Customer, Suppliers, Supply Chain & Warehouse, Logistics/ Transports and Product Engineering*) and (3) one data store (*DNC download (Hi-FIT)*).

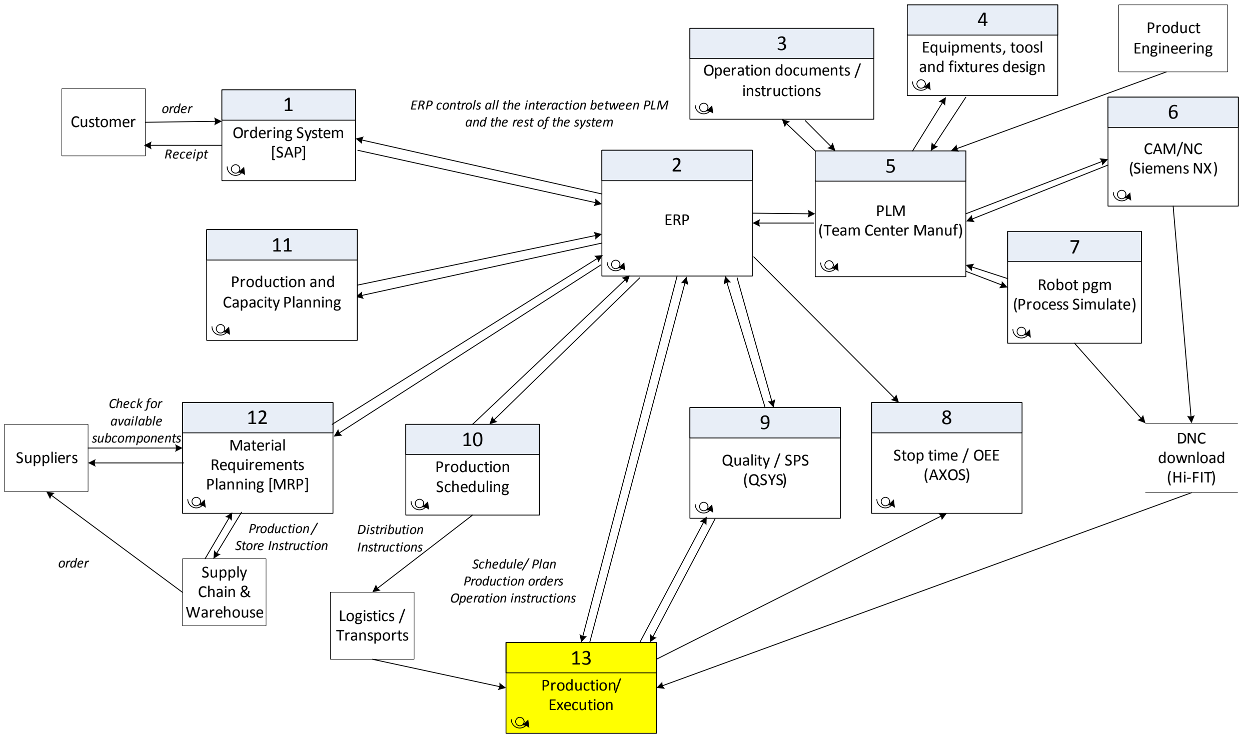

Customer
order
Receipt
1
Ordering System [SAP]
ERP controls all the interaction between PLM and the rest of the system
3
Operation documents / instructions
4
Equipments, toosl and fixtures design
Product Engineering
2
ERP
5
PLM (Team Center Manuf)
6
CAM/NC (Siemens NX)
11
Production and Capacity Planning
7
Robot pgm (Process Simulate)
Check for available subcomponents
Suppliers
12
Material Requirements Planning [MRP]
10
Production Scheduling
9
Quality / SPS (QSYS)
8
Stop time / OEE (AXOS)
DNC download (Hi-FIT)
order
Production / Store Instruction
Distribution Instructions
Supply Chain & Warehouse
Schedule/ Plan Production orders Operation instructions
Logistics / Transports
13
Production/ Execution

Figure 33. GKN Level 1 Data Flow Diagram

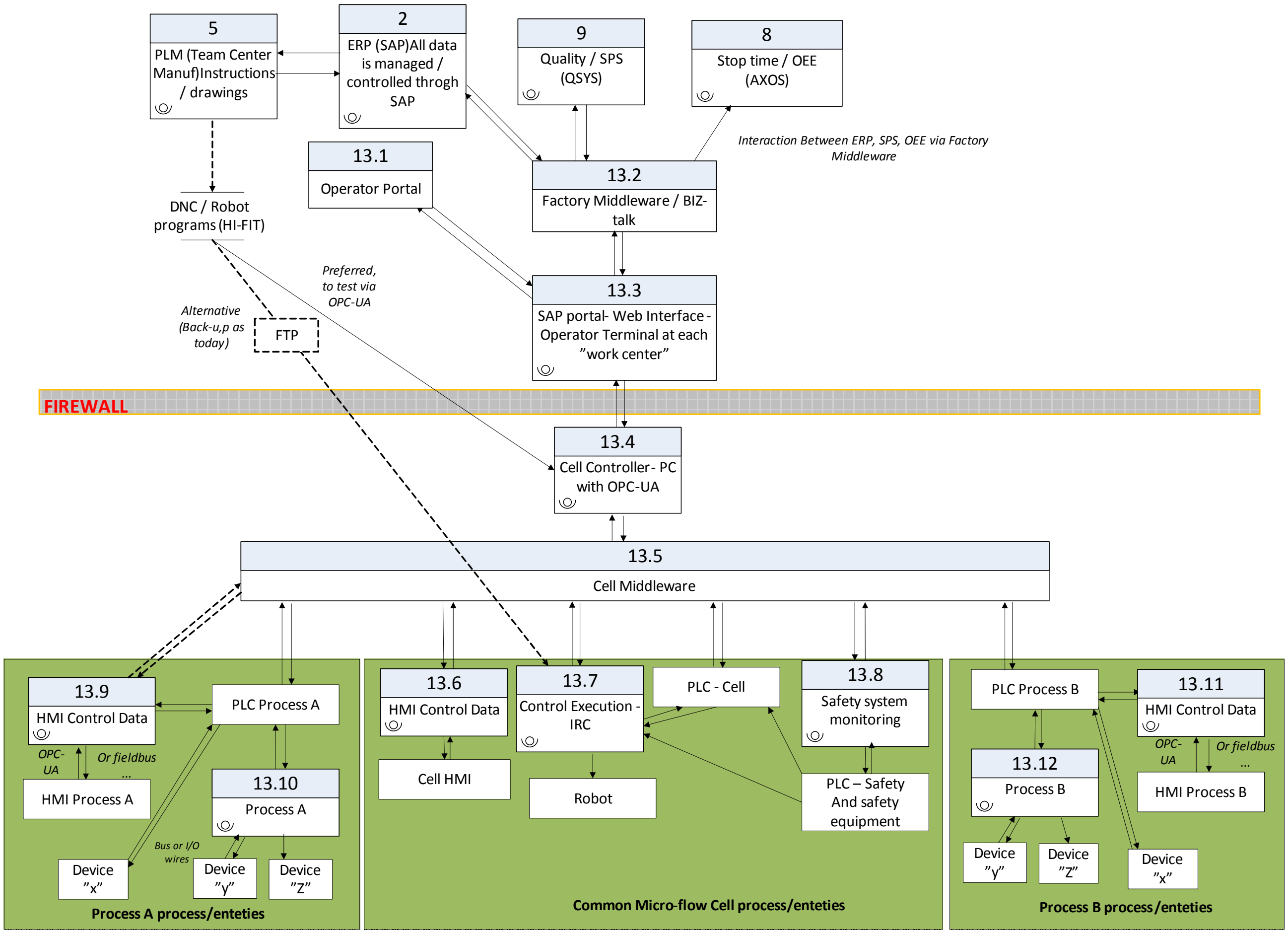
When a *Customer* places an order, the *Ordering System (SAP) (1)* receives all the data and forward them to the *ERP* System (process number *(2)* in Figure 33). The *ERP (2)* will send the data to the *Production and Capacity Planning* process *(11)*, to the *Material Requirements Planning (MRP) (12) (*which is responsible for checking the *Supply Chain & Warehouse* department for the available subcomponents and communicating with the *Suppliers)* and to the *Production Scheduling System*. Each of those three processes will send the updated information back to the *ERP* System. *ERP (2)* will then send the production schedule to the *Production Scheduling (*process number *(10) in* Figure 33*), Quality/SPS (QSYS) (9), Stop time/OEE (AXOS) (8) and PLM* (process number *(5)*). *PLM*, which will be fed also with information from the *Product Engineering* department, will update the *Operations documents/instructions (3) and Equipment, tools and fixtures design (4)* processes, as well as the *CAM/NC (Siemens NX) (6) and Robot program (7)* processes. Process *(6)* and *(7)* are responsible for the program that will be used by the main PLC. This kind of programs are stored in a data store called *DNC download (Hi-FIT)*. Finally *Production/Execution* process (process number *(13)* in Figure 33) will take as input the information from the *Logistics/Transports* department, the *ERP* system, the *Quality/SPS (QSYS)* and the *DNC download* database, to perform the objectives in the new integrated system.

5.5.3 Level 2- Production/ Execution process

The *Production / Execution* process (process *(16)* as seen in Figure 33) will be decomposed in a *Level 2* diagram in this section. The process contains: (1) 12 processes (*PLM, ERP, SAS, OEE, Operator Portal, Factory Middleware, Sap Portal, Cell Controller, Cell Middleware, three HMI Control Data, Process A, Process B, Control Execution and Safety System Monitoring*), (2) seven external entities

(*HMI Process A, HMI Process B, Cell HMI, PLC Process A, PLC Process B, PLC Cell and the Robot*) and (3) one data store (DNC/ Robot Programs).

Level 2 DFD contains:


5
PLM (Team Center Manuf)Instructions / drawings
2
ERP (SAP)All data is managed / controlled throgh SAP
9
Quality / SPS (QSYS)
8
Stop time / OEE (AXOS)
Interaction Between ERP, SPS, OEE via Factory Middleware
13.1
Operator Portal
13.2
Factory Middleware / BIZ-talk
DNC / Robot programs (HI-FIT)
Preferred, to test via OPC-UA
Alternative (Back-u,p as today)
FTP
13.3
SAP portal- Web Interface - Operator Terminal at each "work center"
FIREWALL
13.4
Cell Controller - PC with OPC-UA
13.5
Cell Middleware
13.9
HMI Control Data
PLC Process A
OPC-UA
Or fieldbus ...
HMI Process A
13.10
Process A
Bus or I/O wires
Device "x"
Device "y"
Device "Z"
Process A process/enteties
13.6
HMI Control Data
Cell HMI
13.7
Control Execution - IRC
Robot
PLC - Cell
13.8
Safety system monitoring
PLC – Safety And safety equipment
Common Micro-flow Cell process/enteties
PLC Process B
13.11
HMI Control Data
OPC-UA
Or fieldbus ...
HMI Process B
13.12
Process B
Device "y"
Device "Z"
Device "x"
Process B process/enteties

Cell PLC is responsible for the Security of the Cell

PLC Process 'A' and PLC Process 'B' are responsible for 4 main processes:

- Brushing,
- Surface Roughness Inspection,
- Marking
- Dimensional Inspection

Figure 34. GKN Level 2 DFD

New orders will be analysed in the *PLM (5)* and the necessary instructions will be sent to the *ERP (2)*. Actually, *PLM* does not communicate with the rest of the system and all the communications and controls occur via the *ERP*. The factory middleware is transferring the data between *ERP*, *SPS* and *OEE* systems. Data transfers in this upper IT level happen in parallel and continuously, but all the data will be finally used by the *Production / Execution* process.

Currently, all the robot programs stored in *HI-FIT* are being transferred via a file transfer protocol (*FTP*) to the *Control Execution IRC (*process number *(13.7)* in Figure 34*)*. This protocol is likely to be replaced with a new transfer system based on the OPC-UA standard and will be responsible for transferring robot programs from *HI-FIT* to the *Cell Controller (13.4)*.

When the robot needs a program, a request is sent to the *HI-FIT* which, in turn, sends back the necessary program via middleware to the *Control Execution IRC (13.7)*. *PLC Cell* is responsible for the security of the cell and *PLC Processes "A"* and *"B"* are responsible for each single process (Brushing, Surface Roughness Inspection, Marking and Dimensional Inspection). The *Cell HMI* is responsible to monitoring and controlling the cell equipment and the execution of key functions.

5.6. Use case Asset mapping

A detailed questionnaire was prepared as a part of WP2 and WP3 (refer to Appendix II) and sent to the use cases. The intention of this questionnaire was to enable a better specification of the middleware, standard interfaces and adaptors based on the technical assets that the use cases would like to use within their to-be architecture. The assets have been collated and presented in Table 6.

Table 6. Assets that will be used within the to- be architecture of the use cases

Use case	Data Model		Databases	Robots	Controllers
	Proprietary Data Model	Standardised Data Model			
WHR	Ontology		SQL2000	UR10, ABB	
eDistrict		OPC-UA		Comau	PLC: Siemens IM-151
Siemens	Unspecified		Oracle, SQL, SAP		Sinumerik 840D
GKN			FIT-HI	ABB, 6640-180 IRC5 Tool changer: ABB, RSP 3HACO21652-001	PLC: Simatic S7-300 CPU 317-2 PN/DP

The intention of WP6 is to capture all test results, even though some testing activities might be conducted within pre-test beds (e.g. TUBS, Emden) or with the actual production environment (e.g. at the use case sites). It should be noted that duplication of testing efforts should be avoided. Testing activities will be based on the assets available at each test labs, as well as the supporting infrastructure. A list of potential hardware assets that are available at the test labs (MTC and SmartFactory), as well as pre-test beds (UNINOVA, TUBS and Loccioni) can be seen in Table 7. It is to be noted that this list may be extended or reduced based on the availability of the assets at a specific time. The next deliverable for this work package (D6.4) will provide detailed description of the set-up and the infrastructure involved in the demonstrator. Deliverable D6.4: "*Self-Adaptive Machines Demonstrator Documentation and Results*" which will be released on June 2017, will also present details of the test cases.

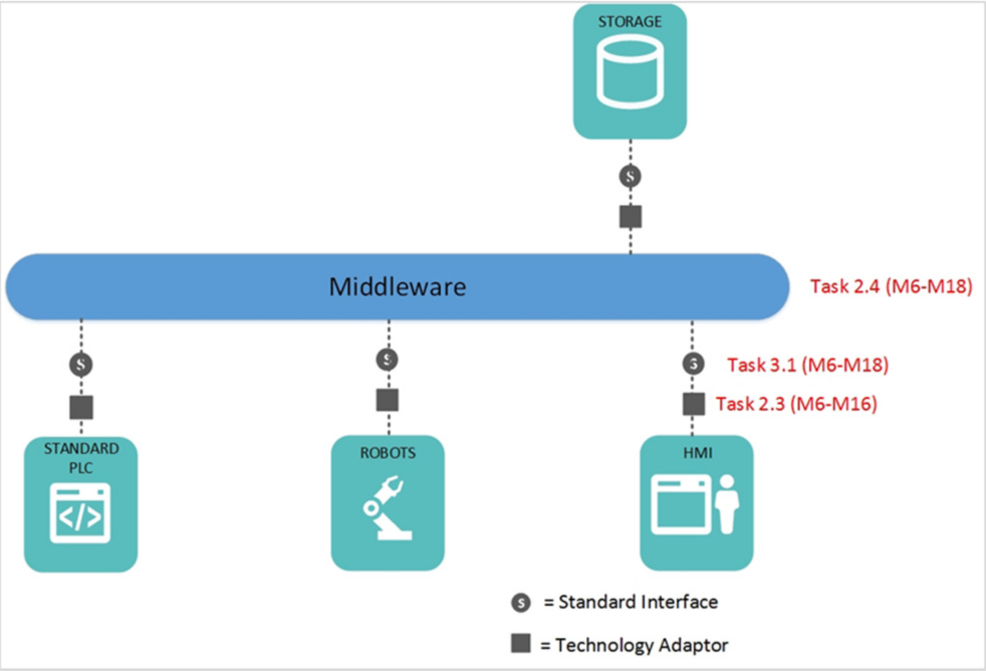
Table 7. Potential assets available at test labs (MTC and SmartFactory) and pre-test beds

Partner Name	Databases	Robots	Controllers	Machines
MTC	Microsoft SQL, Hadoop Platform in HPC	ABB 140, UR5 and ABB 1200	PLCs: Siemens S7, Siemens ET200, Siemens ET200 S, Siemens S7-1200, Beckhoff re-deployable kit, Omron NJ3	CNC: DMG DMU50 Milling Machine, DMG DMC1035 V Ecoline

			Motion Control: Siemens, Beckhoff	Milling Machine, DMG DMU210 FD Milling Machine, DMG DMU 340 Milling Machine
SmartFactory	MySQL	UR5, Kuka KR3, Mitsubishi MovemasterEx	PLCs: Siemens S7-300, Siemens S7-400, Siemens S7-1516+Periphery (ET 200), Beckhoff CX 9001, PhoenixContact ICL 350PN Motion Control: LinMot E1130	CNC: emco Concept Mill 105
TUBS	MySQL Server/DEBIAN	Kuka KR6-2	PLCs: Keka CP 265/W, Heidenhain ITNC530, Fanuc 16i-TB, Fanuc 16i-T, Siemens 840D Solution Line, Traub systems TX-8-D, Sinumerik 810 G (880), Festo CECC-LK, Kuka KRC4	DMG DMU 100 monoBLOCK, Studer S 120 Ernst, Studer S 40, Spiller TC 600, Traub TNS 60 CNC, Blohm Profimat 307 CNC,
Loccioni		Schunk Powerball Lightweight Arm LWA 4P, SDH servo-electric 3-finger gripping hand, Robosoft robuLAB-80	PLCs: Siemens S7	
IPB			PLCs: Modicon M340, Omron C200HG, Omron CPM1, Siemens S7 Motion Control: ABB – ACS 355-03E-07A3-4	CNC Machine Deckel Maho – DMC 63V

6. Test Scenario for Use cases

As indicated in Section 1, the scope of Task 6.1 is restricted to validating the concepts of the PERFoRM architecture using standards machines and robots. The scope of Task 6.1 has been illustrated in Figure 35. Though D6.1 is due in M12 (September 2016), the development effort of the middleware (Task 2.4), standard interfaces (Task 2.3) and technology adaptors (Task 3.1) run from M6 to M18, M6 to M16 and M6 to M18, respectively. Consequently, this deliverable is limited to presenting the test scenarios only. Detailed test cases for the test scenarios will be detailed in the next deliverable for Task 6.1 i.e. Deliverable 6.4: "*Self-Adaptive Machines Demonstrator Documentation and Results*".



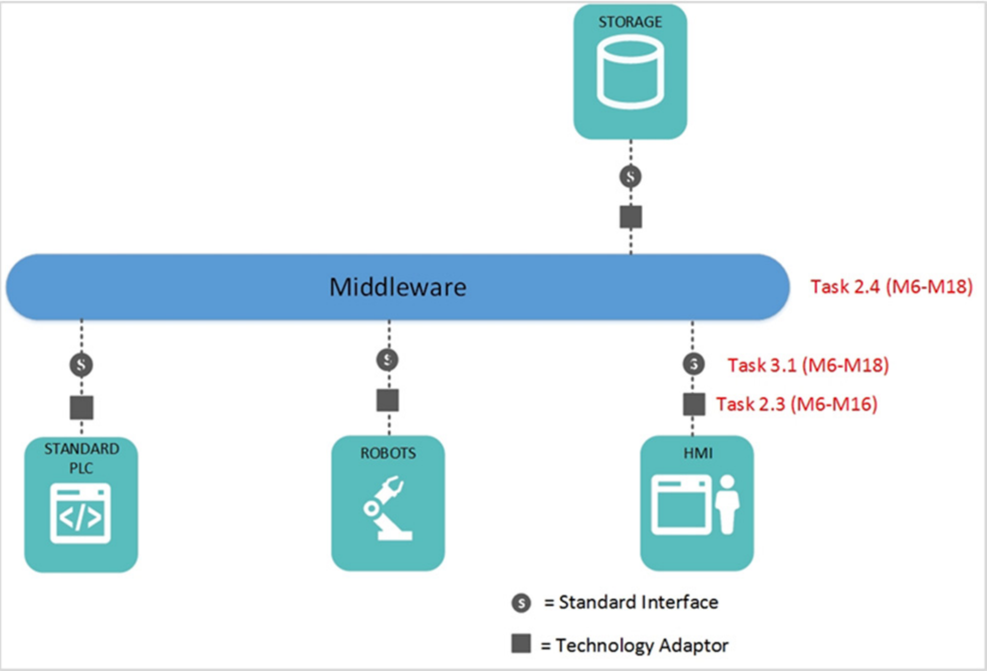


Figure 35. Scope of Task 6.1 and the time lines for the development of the individual architectural elements

The systems that have been described before (see Section 5) consists of different elements, which include standard and legacy systems, but also new concepts and technologies. To de-risk these solutions before they are implemented in real factories, the correct behaviour of each system and its components must be validated in a safe environment. The correct behaviour of the systems has been described by the requirements collected in D 1.2: "*Requirements for Innovative Production System – Functional requirement analysis and definition of strategic objectives and KPIs*". As Task 6.1 is focused on the asset level of each system, the generation of test scenarios for the Siemens's use case will remain out of scope. Test scenarios for Siemens will be described in later deliverables of this work package.

6.1. Whirlpool Test Scenarios

Whirlpool has two use case scenarios to demonstrate the flexibility of PERFoRM on different architecture levels. The first one is focusing on the development of a digital twin for the microwave factory and using the system to reconfigure the factory by changing and mapping KPIs to Key Performance Factors (KBFs). The second use case has its focus on the asset level, aiming at developing of an automatic or semi-automatic robotic reconfiguration system. In particular the robot should perform an electromagnetic compliance (EMC) leakage test based on special markers in the CAD file of the microwave. The markers in the CAD file should be translated into a Universal Robot path file and transferred to the robot in an automatic or semi-automatic way.

Because the scope of Task 6.1 is to de-risk and demonstrate the technologies on the asset level, the test scenario developed in this deliverable has its focus on the first use case, in particular the verification of the correct behaviour of the robot and the interaction with the robot path repository.

6.1.1 Robotic Cell Reconfiguration

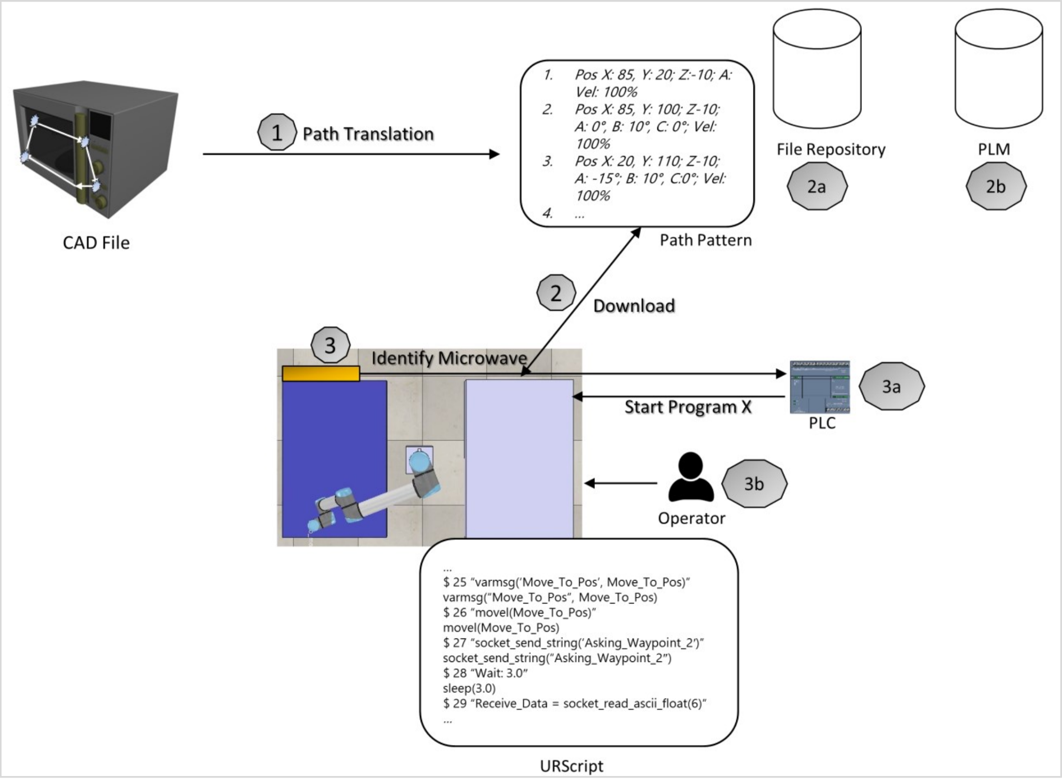
The Robotic Cell Reconfiguration System is aimed at the automatic path programming of an EMC leakage test. Figure 36 depicts a conceptual view of the system, which consists of the system steps "Path Translation", "File Download", "Microwave Identification", "Program Start", "Leakage Test". Currently there are different solutions for these steps under consideration.

In the first step, the CAD path defined during the design time of the microwave must be translated to a robot understandable path file, which is basically a text file. In this particular case, the path file pattern must be in a format that can be processed by a Universal Robot (UR). The path translation step can be done in a very sophisticated way, by identify critical points on the microwave (e.g. welding points or the door) or in a simpler way, by adding specific markers to the CAD file, such as start and stop points in the path of the robot (waypoints) (refer deliverable D9.1: "*Description of requirements and Architecture Design*" (Whirlpool, 2016)).

For the storage and the file transport to the robot, there are currently two options (refer to Figure 36, File Repository: Point 2a and PLM: 2b). In the first one, the path file is stored in the factory's PLM system. This would include the development and test of an adapter to the PLM system, to be able to transfer the file directly to the robot or over middleware. The other solution is bypassing the PLM and storing the file in a standard file repository, which can be directly accessed by the robot or the middleware without an additional adapter.

For the demonstration purpose in the Whirlpool factory, the second solution is currently preferred, as it is not in the scope of PERFoRM to handle the interaction between a robot and a PLM (Whirlpool, 2016).

The PLC can look up the corresponding path file ID and give the download and start instructions to the robot. Another option is to manually identify the microwave and to start the program by the operator. The last step is the execution of the leakage test, which is basically the execution of the UR 10 program with the path from the CAD file.



```
1.    Pos X: 85, Y: 20; Z:-10; A:
      Vel: 100%
2.    Pos X: 85, Y: 100; Z-10;
      A: 0°, B: 10°, C: 0°; Vel:
      100%
3.    Pos X: 20, Y: 110; Z-10;
      A: -15°; B: 10°, C:0°; Vel:
      100%
4.    ...
```

```
...
$ 25 "varmsg('Move_To_Pos', Move_To_Pos)"
varmsg("Move_To_Pos", Move_To_Pos)
$ 26 "movel(Move_To_Pos)"
movel(Move_To_Pos)
$ 27 "socket_send_string('Asking_Waypoint_2')"
socket_send_string("Asking_Waypoint_2")
$ 28 "Wait: 3.0"
sleep(3.0)
$ 29 "Receive_Data = socket_read_ascii_float(6)"
...
```

Figure 36. Robot Reconfiguration System

6.1.2 Test Scenarios Description

Table 8 gives an overview of the test scenarios related to the use case “Robotic Cell Reconfiguration”. A more detailed description can be found after the table. Each test scenario is described by an ID and a name, and reflects a requirement identified by its requirement ID. The overview includes also the priority of a requirement.

Table 8. Test Scenarios- Whirlpool

Test Scenario ID	Test Scenario	Priority
TS-WHP-F-1	Verify that robot can access the correct path pattern file with an adapter	high
TS-WHP-F-1.1	Verify that the robot can access the file repository with an adapter	high
TS-WHP-F-1.2	Verify that the robot can access the PLM system with an adapter	low
TS-WHP-F-2	Verify that the robot follows the path described in the CAD model	high

TS-WHP-F-2.1	Verify that the robot follows the path with the correct behaviour (velocity, acceleration)	high
TS-WHP-F-2.2	Verify that the measuring heads orientation is correct over the whole path	high
TS-WHP-F-3	Verify that the robot can access the correct path pattern file via the middleware	low
TS-WHP-F-3.1	Verify that the robot can access the middleware	low
TS-WHP-F-3.2	Verify that the middleware can access the file repository	low
TS-WHP-F-3.3	Verify that the middleware can access the PLM	low
TS-WHP-F-3.4	Verify that the middleware can route text files between two systems	high
TS-WHP-F-4	Verify that the system can identify the correct product	low
TS-WHP-F-4.1	Verify that the system recognises when a new product occurs	low
TS-WHP-F-4.2	Verify that the correct product is identified	low
TS-WHP-F-5	Verify that the PLC correlates the product ID with the correct robot path pattern	low
TS-WHP-F-5.1	Verify that the PLC gets the correct product ID	low
TS-WHP-F-6	Verify that the leakage test starts when the path download is completed and the product is in position	high
TS-WHP-F-6.1	Verify that the PLC starts the process properly (automatic)	low
TS-WHP-F-6.2	Verify that the operator can start the leakage test (manual)	high

TS-WHP-F-1
The robot must have access to the path pattern file which is either stored in the PLM system (*TS-WHP-F-1.2*) or in a file repository (*TS-WHP-F-1.1*). For this, an adapter acts as a meta-layer between the storage system and the robot.

TS-WHP-F-2
This test scenario covers the correct behaviour of the robot. The robot must follow the path as it is described in the CAD file. This includes also the orientation of the measuring head, (*TS-WHP-F-2.2*) and the robots motion with the proper velocity and acceleration (*TS-WHP-F-2.1*).

TS-WHP-F-3

The transfer to the robot can either be done by done with the middleware solution or with a standard file transfer protocol over TCP/IP. In the current stage of the project, it is not clarify which solution will be used in this use case. By using the middleware system, it must be verified that the robot can access the middleware (*TS-WHP-F-3.1*) and that the middleware can access the file repository (*TS-WHP-F-3.2*) or the PLM (*TS-WHP-F-3.3*), and that the middleware is able to route text files between systems, in this case the file repository or the PLM and the robot (*TS-WHP-F-3.4*). By passing the middleware by using a standard file transfer protocol over TCP/IP would lead to different test scenarios. For this it is necessary to test if a TCP/IP file transfer connection between the robot and the file repository (*TS-WHP-F-2.1*) or to the PLM (*TS-WHP-F-2.2*) system is possible, by using the developed adapter (see TS-WHP-F-2)

TS-WHP-F-4

The robot reconfiguration system must be able to recognise when a new product occurs (*TS-WHP-F-4.1*) and it must be able to identify the product (*TS-WHP-F-4.2*).

TS-WHP-F-5

When the PLC is used to identify the product and to start the download, it must be verified that the PLC gets the correct product ID from the identification system used (*TS-WHP-F-5.1*) and that the PLC correlates the product ID to the right robot path pattern.

TS-WHP-F-6

The start of the leakage test can either be done automatically by the PLC (*TS-WHP-F-6.1*) or manually by the operator (*TS-WHP-F-6.2*). In both cases, it must not start before the product is in position and the path download is completed.

6.1.3 Test Items

The following components are tested in the scenarios:

- The **Universal Robots Script,** which downloads the path and processes the leakage test by following the given path.
- The **Adapter,** which acts as a meta-layer between the robot and the repository/PLM or the robot and the middleware. The adapter is implementing the mechanism to access the path pattern file.
- The **Middleware**, which is used to transfer (text) files between two systems, if used.
- The **PLC,** which identifies the robot and starts the download process and the leakage test.

6.1.4 Test Scope

This section gives an overview of the features of the system that will be tested in this test step (in scope) and the features that will not be tested in this step (out of scope).

Table 9. Test Scope- Whirlpool

ID	Feature	Scope
F-WHP-1	Translation of the path described in the CAD file to a path pattern file	No
F-WHP-2	Download or file transfer of the path pattern file to the robot	Yes
F-WHP-3	Identification of the microwave by the identification system	Yes
F-WHP-4	The automatically or manual start of the leakage test	Yes
F-WHP-5	The Path Following	Yes

6.2. GKN Test Scenarios

GKN has one main use case scenario to demonstrate a new industrial structure in the current production plant. The main focus of this use case is to construct a Reconfigurable Robotic Cell that helps to decrease the change over time. The cell should also be able to recognise different products, which enables flexibility and re-configurability in the whole industrial plant. The plant consists of a common Micro-flow Cell, two processes A and B around the main cell. GKN's overall architecture can be seen in Figure 36.

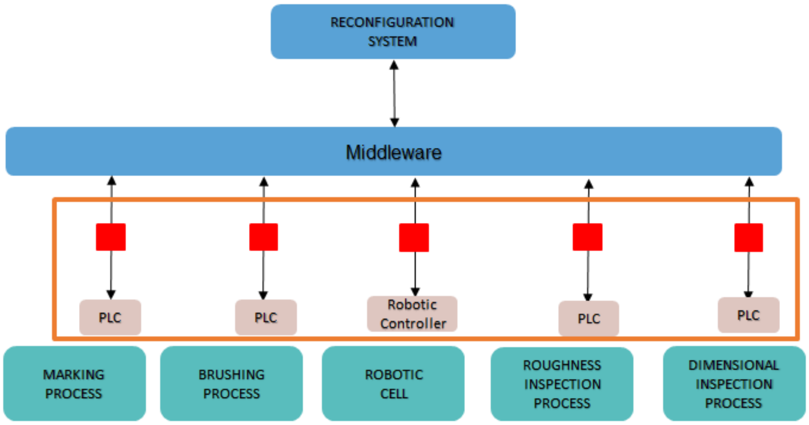


Figure 37. GKN's architecture – red blocks represent the technology adaptors

6.2.1 Test Scenarios Description

As the IT architecture for GKN is currently under development, not all its details are known. The test scenarios described below include testing of all the devices connected to the cell middleware. However, at the moment it is not completely clear which of these devices will require a custom data model translator to communicate with the middleware. Therefore the scenarios presented in Table 10 are still generic and include all the possible scenarios that may require testing.

Table 10. Test Scenarios- GKN

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

The middleware is one of the core components in the PERFoRM architecture. The test scenarios that need interaction with the middleware within the GKN context includes tests to validate that the: (1) robot can receive data and commands from the middleware (*TS-GKN-F-1*), (2) user can investigate the status of the cell and the manufacturing process through the HMI and provide input on the same interface (*TS-GKN-F-2*), (3) middleware can receive information about cell security (*TS-GKN-F-3*), (4) the PLC of each process is able to communicate with the middleware (*TS-GKN-F-4*) and (5) the devices in each process are able to communicate with the process PLC using the required (wired or wireless) adaptor (*TS-GKN-F-5*).

All the test scenarios presented here require that a mechanism is already in place to allow the cell to download the process instruction from the Hi-Fit repository, even though the full test of the functionalities of the architecture and the production process will be performed respectively in Tasks 6.2 and 6.3.

The *TS-GKN-F-5* test scenario requires the testing of each device which may be used during the manufacturing process. The operator on receiving instructions needs to change the tools/fixtures as required. A suitable robot program should be downloaded to the robot from the DNC such that the robot is able to function on the new product.

6.2.2 Test Items

The components that will be used in testing can be found below:

- The **DNC / Robot Programs,** which are stored on the HI-FIT repository,
- The **Middleware** to transfer (text) files between the two systems, if used,
- The **PLC-** Cell to identify the process and to start the download process of the programs,
- The **Robot controller programs** to control the actions of the robot.

6.3. IFEVS Test Scenarios

E-district's objective is the creation of a Cyber Physical factory environment, based on a decentralised cloud architecture that connects the physical systems to all the actors involved. High quality Electric Vehicles will be produced in low budget and flexible assembly lines. The PERFoRM system will increase the automation level and contribute to the improvement of the most critical operations, in order to improve the quality and the productiveness. E-district's test scenarios are focused on the steps that a product will follow, through the different stages of production. The main systems that will be used on E-district's production system can be seen in Figure 38.

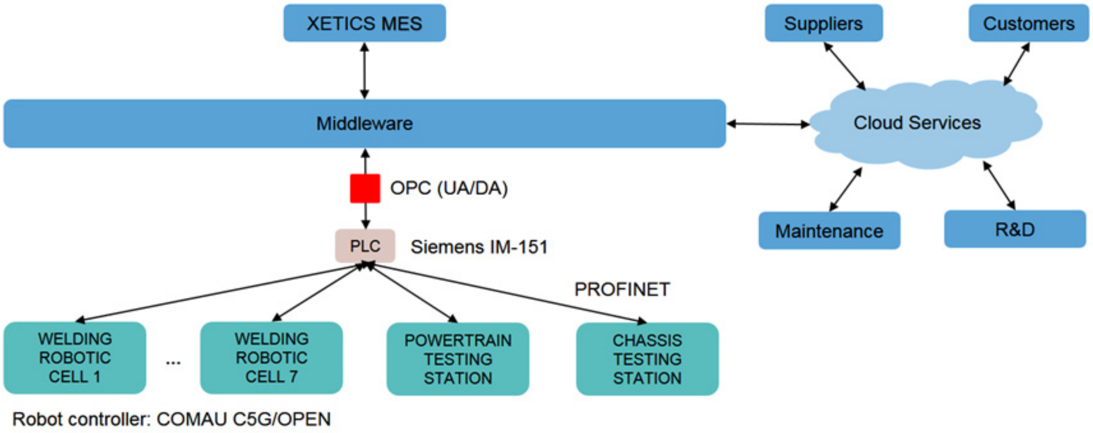


Figure 38. E-district's architecture – red block represents the technology adaptors

6.3.1 Test Scenario Description

Table 11 illustrates the test scenarios related to the E-district's use case.

Table 11. Test Scenarios- E-district

Test Scenario ID	Test Scenario	Priority
TS-IFEVS-F-1	Verify that the middleware can access the scheduling system through the adaptor.	high
TS-IFEVS-F-2	Verify that the PLC starts the process of controlling the sensorised templates first.	high
TS-IFEVS-F-3	Verify that the control of the sensorised templates is valid.	low
TS-IFEVS-F-4	Verify that the middleware can access the PLC through the adaptor.	high
TS-IFEVS-F-5	Verify that the PLC starts the welding process properly, after finishing the control of the sensorised templates.	high
TS-IFEVS-F-6	Verify that the HMI can access the PLC through the adaptor.	low
TS-IFEVS -F-7	Verify that the robot will access the PLC.	high

A description of each of the test scenarios included on Table 11, can be found below.

TS-IFEVS-F-1

Middleware is one of the most significant components that will be implemented in the system, in order to build the PERFoRM architecture. As the middleware has to interact with other legacy systems, special adaptors will be developed and used in order to allow the communication between those entities. This test scenario will check whether the middleware can access the XETICS MES system and get the data that are related to the schedule of the production.

TS-IFEVS-F-2

PLC is responsible for the control of the sensorised templates. In E-district's production line, special templates will be equipped with part presence sensors. Their task will be to ensure that all the welding parts have been put in the right place. The welding phase will start only after the control of the templates is successful. This test scenario specifies that the PLC will first control whether all parts have been placed in the right place.

TS-IFEVS-F-3

This test scenario will examine whether the control of the sensorised templates is valid.

TS-IFEVS-F-4

Middleware should be able to communicate with the PLC through the adaptor.

TS-IFEVS-F-5

The welding phase will start automatically, only after the control of the sensorised templates is finished.

TS-IFEVS-F-6

The HMI will be used to visualise information that is related to the control of the sensorised templates. HMI will communicate with the PLC through an adaptor.

TS-IFEVS-F-7

The robot should be able to communicate with the PLC in order to start the related tasks.

6.3.2 Test Items

In the test scenarios that will be created for E-district, the following components will be needed:

- The **Siemens IM- 151 PLC**.
- The **COMAU C5G/ OPEN** Robotic Controller.
- The **Middleware**, which will transfer data between different entities.
- The **adaptors**, which will allow the communication between the Middleware and other entities.
- The sensorised templates.

7. Conclusion

The objective of the PERFoRM architecture is to support a new generation of agile manufacturing systems based on the plug-and-produce concept, thus enabling the production of smaller lot sizes, more customised products, shorter lead times and shorter time-to-market. WP6 in particular is responsible for validating the technologies developed for supporting the proposed architecture.

The scope of the three tasks for WP6 has been clarified in this deliverable. The deliverable has also specified the foundations of the PERFoRM architecture and has analysed the proposed architecture of each use case in detail. An inventory of assets present at the pre-test beds that may be utilised for the demonstrator for Task 6.1 has been presented. An initial list of test scenarios for the use cases that can be verified within the realm of Task 6.1 (i.e. machines and robots) has been presented. The next deliverable of Task 6.1 will be presenting details of the set-up of the demonstrator, test cases and the results observed whilst executing the test cases.

8. References

Advanced Strategies, I., 2008. *www.advstr.com.* [Online]
Available at: http://www.advstr.com/web/resources/downloads/Process_Modeling-Why_DFDs-When_Swimlanes_are_Not_Enough.pdf

Aleryani, A. Y., March 2016. *Comparative Study between Data Flow Diagram and Use Case Diagram,* s.l.: International Jouranl os Scientific and Research Publications, Volume 6, Issue 3.

Department Of Defense, M. T. P., 2011. *Manufacturing Readiness Level (MRL) Deskbook.* [Online]
Available at: http://www.dodmrl.com/MRL_Deskbook_V2.pdf
[Accessed 8 August 2016].

Donald S. and Le Vie, J., 2000. *Understanding Data Flow DIagram,* Texas: Integrated Concepts, Inc.

FhG-IPA, 2016. *Requirements Review, evaluation and selection of best available Technologies and Tools,* s.l.: PERFoRM project.

GKN, 2016. *Use Case goals/ KPIs and Requirements Defined,* s.l.: PERFoRM project.

H2020, 2014. *HORIZON 2020- Work Programme, G. Technology readiness levels (TRL).* [Online]
Available at:
http://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf
[Accessed 23 September 2016].

IFEVS, 2016. *MIcro Electric Vehicles Description and Requirements of Architectures in View of Flexible Manufacturing,* s.l.: PERFoRM project.

IPB, 2016. *Definition of the System architecture,* s.l.: PERFoRM project.

POLIMI, 2016. *Guidelines for Seamless Integration of Humans as Flexibility Driver in Flexible Production Systems,* s.l.: PERFoRM project.

R. S. Peres, M. P.-. R. A. D. R. J. B., 2016. *Selection of a Data Exhange Format for Industry 4.0 Manufacturing Systems,* s.l.: s.n.

R.S Peres, M. P.-R. A. D. R. J. B., 2016. *Selection of a Data Exchange Format for Industry 4.0 Manufacturing Systems,* s.l.: IEEE.

Siemens, 2016. *Report on Decentralized COntrol & Distributed Manufacturing Operation Systems for Flexible and Reconfigurable Production Environments,* s.l.: PERFoRM project.

Siemens, 2016. *Siemens Description and Requirements of Architectures for Retrofitting Production Equipment,* s.l.: PERFoRM project.

Whirlpool, 2016. *Description of Requirements and Architecture Design,* s.l.: PERFoRM project.

Xu, D., 2011. *Using Adaptive Agents to automatically generate test cases from the UML activity diagrams,* s.l.: s.n.

Yen, R. I. a. S. Y., 2011. *A Formal Model for Data Flow Diagram Rules,* s.l.: ARPN Journal of Systems and Software.

Yonglei.T., 1991Formal definitation and verification of data flow diagrams. s.l.: s.n.

Yonglei, T., 1991. *Formal definition and verification of data flow diagrams,* s.l.: s.n.

Appendix

I. Acronyms

Abbreviation	Explanation
CPPS	Cyber-Physical Production Systems
DB	Database
DFD	Data flow diagrams
ERP	Enterprise Resourcing Planning
HMI	Human Machine Interface
ICT	Information and communication technologies
KBF	Key Business Factors
KPI	Key Performance Indicators
MES	Manufacturing Execution Systems
MRL	Manufacturing Readiness Level
PLC	Programmable Logic Controller
RE	Requirements Engineering
SCADA	System Control and Data Acquisition
SOA	Service Oriented Architecture
TRL	Technology Readiness Level
XML	Extensible Markup Language

II. Technical questionnaire

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

Whirlpool's answers to technical questionnaire

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

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

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

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

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

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

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

Type of data model	Yes / No
Proprietary data model	Y
Ontology	Y
Unspecified	
Standardized data model	
OPC-UA	N
BatchML (IEC 61512)	N
B2MML (IEC 62264)	N
XMPLant (ISO 15926)	N
CAEX (IEC 62424)	N
AutomationML (IEC 62714)	N
Other(s):	

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

   YES

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

   Virtual Server running VMware.

Whirlpool specific questionnaire

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

   SQL 2000

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

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

   UR10

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

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

E-district's answers to technical questionnaire

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

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

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

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

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

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

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

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

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

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

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

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

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

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

E-district specific questionnaire

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

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

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

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

GKN's answers to technical questionnaire

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

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

Comments:

- Product related data will need to be downloaded before processing as well as uploaded after processing to have full traceability for each individual part produced.
- Similar for process data; e.g. process control programs and parameters downloaded and some uploaded afterwards as part of the logging/documentation
- Some resource monitoring may be of interest, or e.g. measure tool time/remaining tool life.
- Process monitoring or measurements data from sensors

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

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

Comments:

- Product related data will not change very often (weeks/months), but some data needs to be downloaded for each individual part that is produced.
- Similar for process data; e.g. process control programs and parameters downloaded and some uploaded afterwards as part of the logging/documentation

- Some resource monitoring may be of interest, or e.g. measure tool time/remaining tool life, as well as process monitoring, this will need to be updated "continuously" but may not be time critical.

- Measurements data and data from sensors can be large but can perhaps be stored locally in real time and uploaded as batch after the operation has finished.

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

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

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

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

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

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

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

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

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

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

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

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

GKN specific questionnaire

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

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

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

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

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

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

Siemens' answers to technical questionnaire

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

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

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

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

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

   Not decided yet. Depends on system implementation

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

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

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

Type of data model	Yes / No
Proprietary data model	Yes
Ontology	No
Unspecified	Yes
Standardised data model	No
OPC-UA	No
BatchML (IEC 61512)	No
B2MML (IEC 62264)	No
XMPLant (ISO 15926)	No
CAEX (IEC 62424)	No
AutomationML (IEC 62714)	No
Other(s):	

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

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

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

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

Siemens specific questionnaire

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

   Oracle, maybe also accessible via OPC???
   SAP

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

   SQL, SAP

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

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

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

   No PLC → SINUMERIK 840D → Firmware not known

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

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