H2020 – FOF – 09 – 2015

Innovation Action

Smart integrated immersive and symbiotic human-robot collaboration system controlled by Internet of Things based dynamic manufacturing processes with emphasis on worker safety

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 680734

D7.2:Guidebook &recommendation for the deployment of HORSE framework for Application Experiments

Report Identifier:	D7.2		
Work-package, Task:	WP7, T7.2	**Status – Version:**	1.00
Distribution Security:	PU	**Deliverable Type:**	R
Editor:	TNO		
Contributors:	CEA, TCS, TUM, BOSCH-PROSYST, FZI, TUE, KUKA		
Reviewers:	TCS, TUE		
Quality Reviewer:	ED		

Keywords:	Competence Centre, robot, SME, use cases, scenarios, equipment, services, Framework, OSGi
Project website: www.horse-project.eu	

Table of Contents

List of Tables

List of Figures

Abbreviations

CC	Competence Centre
CEA	French Atomic Energy Commission
EC	European Commission
ERP	Enterprise Resource Planning
HFW	HORSE Framework
HMMw	HORSE Messaging Middleware
HORSE	Project acronym
HRI	Human-Robot-Interaction
I4MS	ICT innovation for Manufacturing SMEs
ICT	Information and Communication Technologies
MES	Manufacturing Execution System
MPMS	Manufacturing Process Management System Part of the HORSE framework that takes care of assigning actors to tasks according to the current production demand
PLM	Product lifecycle Management
SCADA	Supervisory Control And Data Acquisition
SME	Small and Medium Enterprise
TCS	Toolmakers Cluster Slovenia
TNO	The Netherlands Organisation for Applied Scientific Research TNO
TUM	Technical University of Munich

Executive Summary

This document provides a guidebook and recommendations for the deployment of HORSE framework for Application Experiments. This document provides clear insights on the usefulness and applicability of the HORSE Framework and as such is a useful documents for any new robotic CC or any other party who is interested to adopt HORSE for industrial applications.

This document is a public deliverable, coordinated by TNO, coming as a result of the HORSE implementations at the Competence Centres (CC) at CEA in France in Saclay, at TNO in the Netherlands in Delft and at TUM in Germany in Munich. Also experience was gathered as a result of the HORSE integration efforts at the pilot experiments at Bosch in Madrid, Odlewnie Polskie in Starachowice and Thomas Regout in Maastricht.

The purpose of this document is to share the expericience gathered at these different sites for further implementations at other sites being the open call participants and the new CCs - being set up in Celje in Slovenia and in Karlsruhe, Germany by TCS and FZI, respectively.

The HORSE framework (HFW) consists of the Manufacturing Process Management System used to describe and control the whole manufacturing process, the HORSE middleware providing standardized means of communication between the components, and the Hybrid Task Supervisor coordinating human operators and robots on the workcell level. This lightweight skeleton is then tailored to the needs of the concrete application by adding reusable, easy to develop case-specific components such as the robot control, augmented reality instructions, safety modules etc.. The description of the HORSE framework components and the guidelines for adopting and configuring these components are provided in the D4.5. This deliverable is a guideline for implementing and adapting the HORSE framework to each premises and its equipment. The guidebook comes as a result of experience gained during the deployment of the HORSE framework in the three Competence Centres. The three CC participated to the elaboration of the guidebook.

1 Overview of the document

The HFW is a technology that is aimed to facilitate the adoption of robotic technologies by manufacturing SMEs. Those are interested to invest in robotic technologies but cannot afford the early upfront costs and long time necessary for visible benefits in the production line. In addition to that, the SMEs need to have their single-task robots and their employees to be assigned to different tasks depending on the current needs. The flexible and dynamic approach to tasks monitoring and allocation in the HORSE framework is one of its more important features. The HFW can be combined with several technologies and robotic as well non-robotic agents, to implement various industrial challenges. A variety of those are being demonstrated in the HORSE CCs, during the implementation of which the experience of adoptin HFW has been gained. For this reason there are many links to CC's in this document.

The document is structured as follows:

- Chapter 1 presents the overview of the document.
- Chapter 2 introduces the basic architecture of the HORSE Framework (HFW). The scope is to show how a factory or part of a factory manufacturing process is modelled and how these processes are interconnected and later executed by the HFW are presented.
- Chapter 3 provides a guide for identifying the challenges to address within a manufacturing industry using the HFW and defining the use cases. Summary of the applications for the HORSE pilots have been provided.
- Chapter 4 presents the role and demonstration scenarios of HFW in the Competence Centers.
- Chapter 5 is a step by step guide for the HFW implementation process
- Chapter 6 presents technical details about the HFW deployment and configuration. For more details please refer to D4.5 User Handbook.
- Chapter 7 describes the HFW positioning in relation to other operational systems like ERP and MES .
- Chapter 8 gives an overview of the IPRS and licensing conditions for the HORSE components.

An example of the execution time of HFW is shown in chapter 3.4

1.1 Glossary

Terms used in the document.

(Some terms' meaning is extracted from the concepts provided on the I4MS website[1])

Advanced robotic solutions	Refers to a new generation of robot-based solutions in the fields of: • Reconfigurable Interactive Manufacturing Cells • Shop Floor Logistics and Manipulation • Plant inspection and servicing
Application Experiment	Application experiments provide first-time users with novel products and services and assist them to evaluate their application in their respective environments.

[1] www.i4ms.eu

Assessment Experiment	Assessment Experiments support suppliers of innovative high-tech equipment to assess and validate their prototypes or products in production-like environments.
Best practices	A best practice is a method or technique that has consistently shown results superior to those achieved with other means, and that is used as a benchmark. (From Wikipedia) The second phase of the I4MS initiative will be dedicated to share best practices and lessons learnt in the fields of advanced robotic solutions, Simulation, Laser-based applications and sensors.
Competence Centres	Competence Centers are one-stop-shops that help companies to become more competitive with regard to their business/production processes, products or services using digital technologies. They are based upon technology infrastructure (competence centre) and provide access to the latest knowledge, expertise and technology to support their customers with piloting, testing and experimenting with digital innovations. Competence Centers also provide business and financing support to implement these innovations, if needed across the value chain. As proximity is considered crucial, they act as a first regional point of contact, a doorway, and strengthen the innovation ecosystem. A Competence Center is a regional multi-partner cooperation (including organizations like RTOs, universities, industry associations, chambers of commerce, incubator/accelerators, regional development agencies and even governments) and can also have strong linkages with service providers outside of their region supporting companies with access to their services. Competence Centre are one of the HORSE instruments to facilitate the appropriation of new technologies in European industry.
Competence Centres demonstration scenario	Demonstration scenarios are implementation of the use cases supported by the Competence Centres illustrating the operational capacities of the Competence Centres.
Competence Centres use cases	Use cases are generic scenarios, independent from the equipment used in the Competence Centres, illustrating the capacities offered by competence centre for robotics applications in manufacturing industries. Use cases rely on the usage of the HORSE framework.
Factories of the Future	It is the European Public-Private Partnership (PPP) launched in 2008 which looks to engage the EU manufacturing industry and deliver the technologies needed for the new sustainable and competitive factories of the future.
I4MS initiative	ICT Innovation for Manufacturing SMEs of the European Commission launched in July 2013 targets to help SMEs and mid-caps in the manufacturing sector along three dimensions: • Provide access to competences that can help in assessing, planning and mastering the digital transformation. • Provide access to innovation networks of a broad spectrum of competences and best practice examples.

	• Provide financial support to SMEs and mid-caps on the demand and the supply side to master the digital transformation.
One-stop shopping	Refers to a solution that offers a multitude of services to a client or a customer in the same location, for instance a cloud-based service. The idea is to provide convenient and efficient service and also to create the opportunity for the company to sell more products to clients and customers.
Pilot experiments	They are small-scale preliminary studies conducted in order to evaluate feasibility, time, cost, adverse events, and effect size (statistical variability) in an attempt to predict performance of a full-scale research project and suggest improvements prior to a bigger-scale test.

2 HORSE Framework and components

2.1 Introduction

HORSE designs and develops a Framework for advanced, process-oriented hybrid manufacturing. Process-oriented hybrid manufacturing is an approach to manufacturing that seamlessly integrates human and robotic actors in vertical manufacturing cells that are horizontally coupled in end-to-end manufacturing processes. The HORSE system supports this approach in an advanced way as it covers dynamic actor allocation to work cells, direct robot control and human actor instruction, closed-loop local event processing and near-real-time global event processing.

To introduce HORSE framework an abstract scenario is presented that represents a wide range of application cases, including those of the three HORSE pilot cases and the open call cases. A high level use case view of the abstract HORSE scenario is presented in Figure 1. The scenario is specified in a UML use case diagram , presenting the way in which a system can be used – not specifying the internal working of a system.

This scenario includes four use cases:

- Design Manufacturing Process: functionality for the design of a manufacturing process across multiple work cells (and possibly across multiple production lines).
- Execute Manufacturing Process: functionality for the execution of a manufacturing process across multiple work cells (and possibly across multiple production lines).
- Configure Manufacturing Task: functionality for the configuration of a manufacturing task within a single work cell (possibly consisting of multiple manufacturing steps).
- Execute Manufacturing Task: functionality for the execution of a manufacturing task within a single work cell (possibly consisting of multiple manufacturing steps).

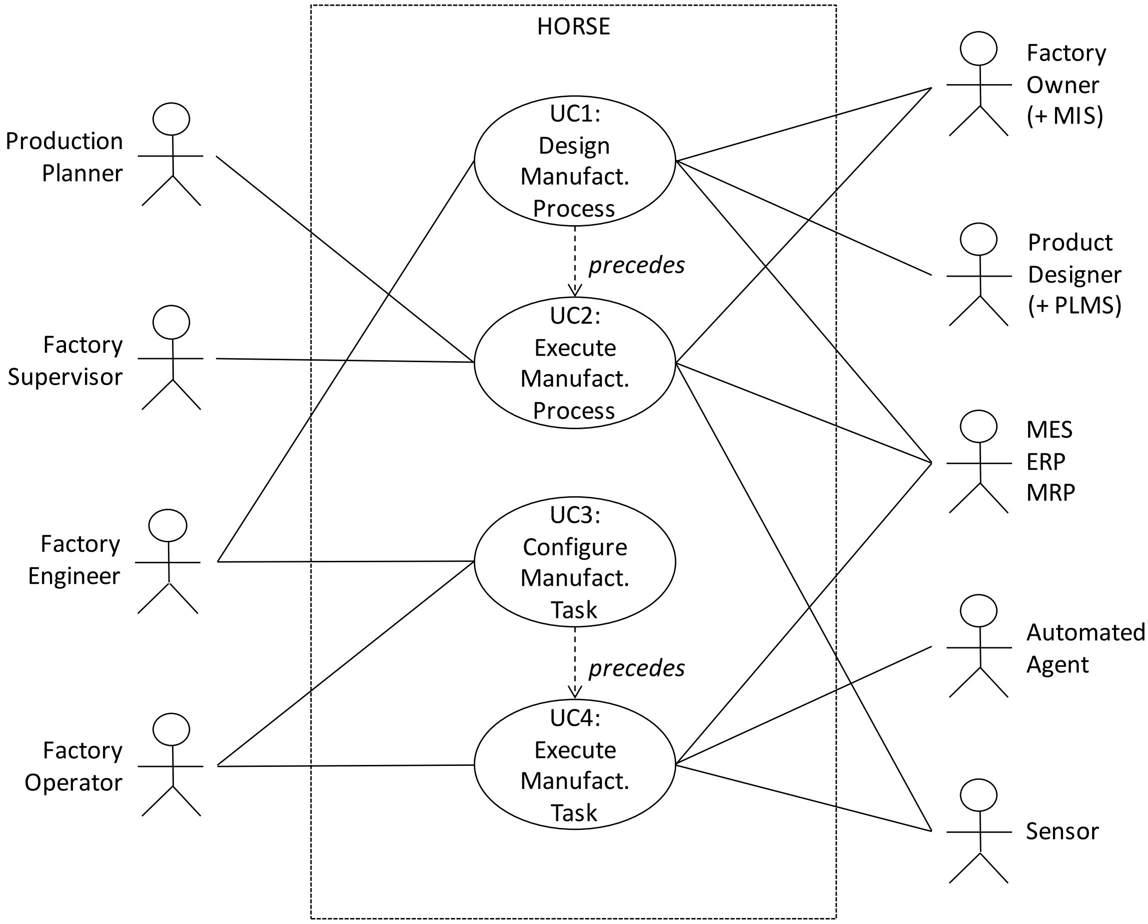


Figure 1: overall HORSE scenario

2.2 HORSE Logical software architecture

The HORSE system must support both manufacturing processes across manufacturing cells and manufacturing steps within manufacturing cells. Typically, one manufacturing process uses a number of manufacturing cells - the process coordinates, the cells perform the actual work. This means that there are manufacturing activities at two distinct levels with different characteristics. Consequently, the HORSE software architecture has two levels as well, which we call *HORSE Global* and *HORSE Local*.

Manufacturing activities need to be designed or configured (to parameterize systems for specific production) and to be executed (to actually manufacture products). This means that the HORSE architecture needs to include modules for design/configuration and modules for execution. This holds both at the HORSE Global and HORSE Local levels. This leads to a software architecture with four modules, as shown in Figure 2: two levels with each two modules.

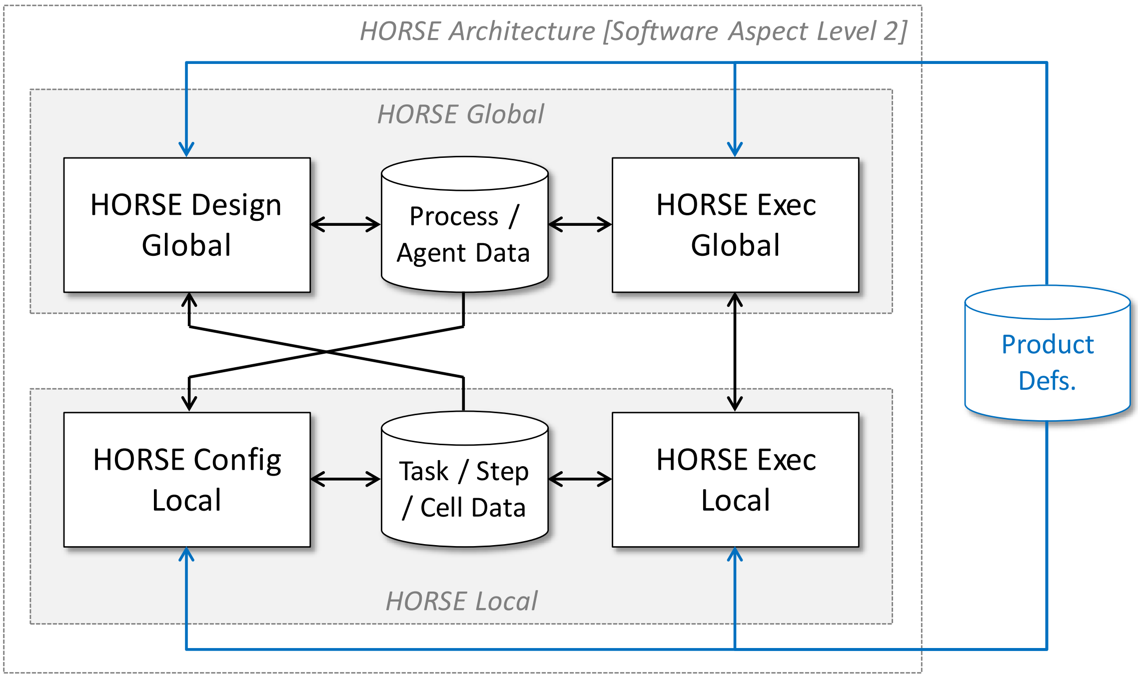


Figure 2: logical software architecture

For more details and elaboration about the HORSE logical architecture the reader is directed to deliverable D2.2: Complete System Design[2].

2.3 HORSE Physical View /Architecture

The physical view describes the mapping(s) of software onto hardware, thereby reflecting the distribution aspect. This view mainly specifies the operational deployment of a system. This is presented in Figure 3. The HORSE framework consists of several software components that can be grouped into three main groups: the core components, the interfaces, and the case-specific ones (see HFW supporting document. The first group contains the components responsible for management of the whole manufacturing process, the **MPMS – Manufacturing Process Management System** and for execution of tasks in the work cell, the **HTS - Hybrid Task Supervisor**. They can be successfully used regardless of the actual scope of the use case. The interfaces contain both the HORSE middleware, which is essential to communication between the components of the framework, and the interfaces connecting the framework to other systems e.g. the Bosch infrastructure or ROS components. Finally, the case-specific components provide functionalities required by concrete applications. Those may involve robot control, trajectory planning, augmented reality etc. Their development is usually driven by a specific use-case; however, they can be adapted to similar scenarios with minimal effort.

Therefore, each HORSE deployment includes the MPMS (level 3) and the middleware (level 2). The first one is necessary to define, execute and monitor the process. The second one, the middleware provides communication capabilities for the heterogeneous components of the framework. Depending on the realized scenario the HTS can be also used to trigger and synchronize tasks on the level of individual work cells.

[2] http://horse-project.eu/sites/default/files/publications/HORSE-D2.2%20(Public%20Version).pdf

The case specific components may be adapted and used in different scenarios if they fit the requirements. Although reusing the existing software is strongly recommended, new components can be integrated as well, as long as they are connected to the messaging middleware of HORSE.

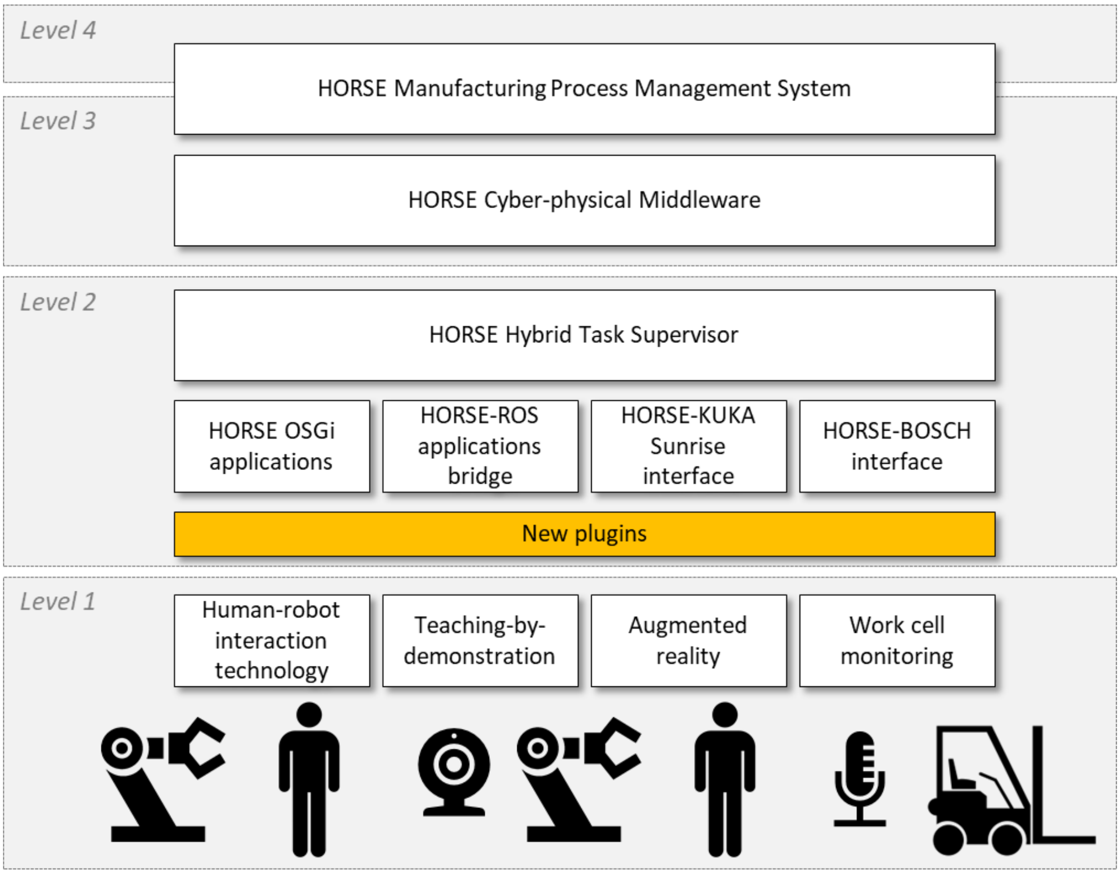


Figure 3: HORSE Framework

2.4 Core components

This section presents the overall function of the HORSE components to support the decisions about reusing them for new challenges. The details about the way to access and configure these components are provided in D4.5:User Handbook which is available in the project website (www.horse-project.eu).

2.4.1 Manufacturing Process Management System (MPMS)

The MPMS is the collection of subsystems responsible to orchestrate the tasks of agents in the manufacturing processes. Orchestration is dependent on the design of the processes and agents. The MPMS includes the functionality to design processes and describe agents, and execute the processes by assigning activities to agents. Figure 4 shows the process management layer, embodied by the MPMS, as a function of horizontal and vertical integration. Horizontal integration refers to the inter-operability between the manufacturing processes and other management or support processes in the enterprise. Vertical integration refers to the link between the process management and resources located on the factory floor.

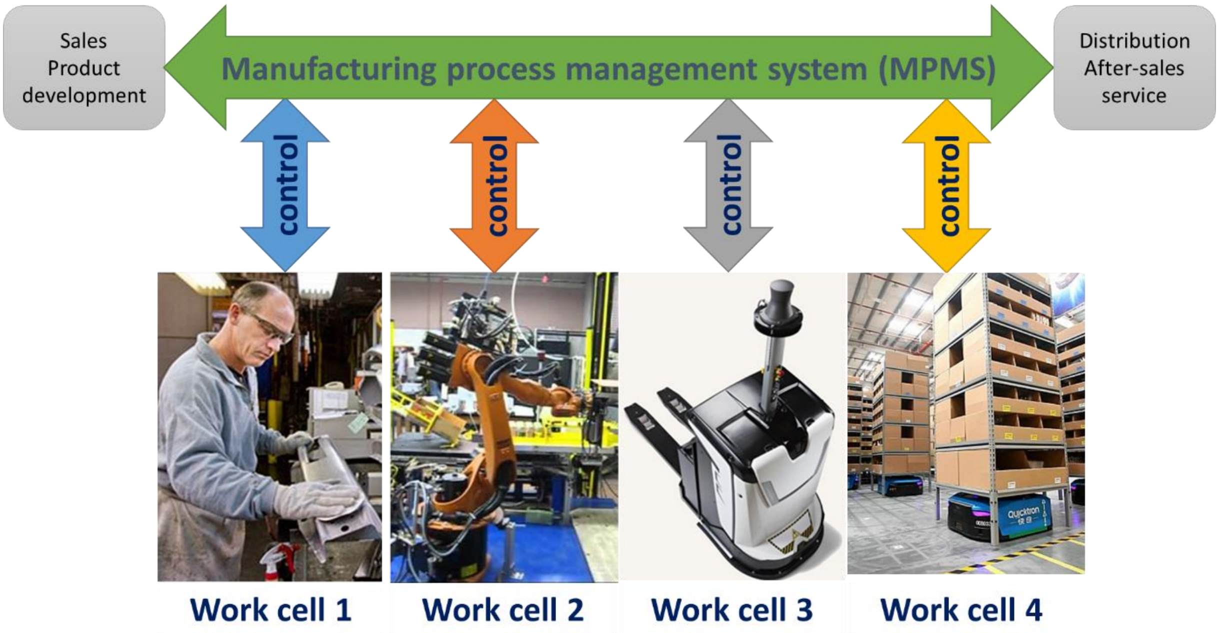


Figure 4: Conceptual illustration of the MPMS in relation to the work cells

The MPMS consists of three system modules and a single data store within the larger HORSE System. The MPMS elements can be described as the following:

- **Process design**: The modules which enable visual modelling of a manufacturing process, comprising of tasks, events, gateways and connectors.
- **Agent design**: The modules which enable creating or editing profiles of production agents, including their competences, authorisations and performance indicators.
- **Global execution**: The run-time process engine which enacts the designed process and assigns agents to perform the tasks. This subsystem also provides an overview of active processes and their status.
- **Process / agent data**: This represents a collection of data stores in which process and agent definition and execution data is stored. These are logical data stores which may or may not be realised in a single database.

An important scoping dimension to mention is the distinction between **global and local functions**. The software aspect of the architecture clearly contains layers corresponding to notions of global and local. Local includes all activities and objects within a single work cell, while anything that crosses work cells is considered global. This is used as a starting point to establish a scoping statement that is not dependent on the physical hierarchy of the manufacturing system.

A manufacturing process consists of activities, events, gateways and connectors. Activities may be sub-processes or tasks. A single process may contain multiple tasks, located and performed in multiple work cells. A task is assigned to and performed by a team of one or more agents. This team may be a virtual team that only exists for the duration of the task execution. A single task is entirely contained within a single work cell, for the duration of the task. For this reason, task is considered the smallest unit of work that appears in the global layer. The case of vehicles is more complicated, but still conforms to this definition. The transport task performed by a vehicle is located in the work cell that is defined by the route of the vehicle.

Finally, it is prudent to state functions which are explicitly excluded from the system. The HORSE Project does not aim to develop detailed planning and scheduling technology. Such technology is widely available and advanced. Instead, the MPMS aims for run-time control which orchestrates all agents in an efficient manner. Planning optimisation and detailed scheduling are assumed to be done and available as input to the HORSE System. The MPMS then executes the process according to control flow, in support of the production plan.

2.4.2 Hybrid Task Supervisor (HTS)

The Hybrid Task Supervisor is the component related to the local execution of a task in a work-cell by both the human operators and the robots. It receives the task execution requests from the MPMS and it keeps track of the progress of the task execution. Tasks are defined through the user-friendly graphical interface available in the HORSE framework.

When a request is received, the Hybrid Task Supervisor retrieves the information related to the considered/matching task in order to activate the autonomous agents in the work-cell.

Furthermore, after the processing of a request, this component sends a message to the MPMS global level to notify the start time of the execution of the task involved. A similar notification is sent after the completion of the task, allowing the work-flow of the entire process to continue.

In addition, the Hybrid Task Supervisor allows to keep track of the progress of the task during the execution, receiving also information about anomalies, like obstacles or unexpected humans that block some robot trajectories. In this case the component is responsible to send an alert to the global level.

2.5 Interfaces

Interfaces allow easy interoperability of the Framework, by enabling easy replacement of modules and integration of new ones. Interfaces developed to support the specific needs of the pilot uses cases include:

- OSGi plugins to the existing OSGi nodes;
- New ROS components;
- Modules based on other technologies;
- ROS integration of KUKA LBR iiwa (providing a ROS connector running on a KUKA Sunrise Cabinet)
- Interface to industrial equipment: HORSE-BOSCH Adapter

2.5.1 HORSE Messaging Middleware (HMMw)

The HORSE Messaging Middleware is a software solution supporting HORSE to overcome the heterogeneity of the HORSE software components adopting widely adopted standards. It is realised through a messaging infrastructure with star topology in which the individual components (nodes) communicate with each other through a local broker. The components could be organised in functional domains, each represented by a broker and all brokers communicating with each other through a dispatcher. The JSON formatted messages are exchanged over the WebSockets low-level communication protocol. This allows the implementation of the HORSE Message Node specification as part of every HORSE module, with no additional constraints for the programming language or the execution environment.

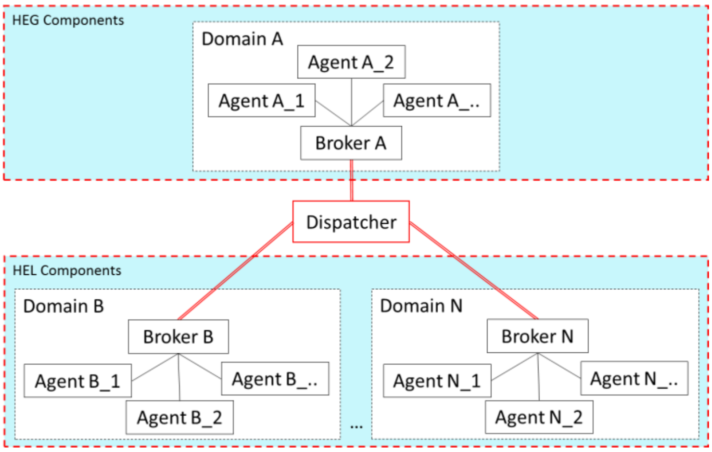


Figure 5: HORSE Messaging Middleware components

The message-driven collaboration between the major HORSE components permits the detachment of their implementations from the agreed interfaces. This in turn promotes the continuous development and testing of all components with increasing maturity of the implemented functionality.

The biggest benefit of such an approach is that integration of new components in the framework requires only development of WebSockets-based communication client and processing the messages exchanged between the new component and the rest of the HORSE framework.

2.5.2 HORSE-ROS Bridge

The HORSE-ROS Bridge interface allows the easy communication between native ROS nodes (the Open-Source framework "Robotic Operating System") and nodes using the HMMw.

This interface permits middleware clients to use the full ROS functionality available to native ROS nodes. The forwarding of HORSE events originating at native ROS nodes to middleware nodes is supported as well and it offers a ROS service interface to forwards arbitrarily complex messages.

The HORSE-ROS Bridge is a useful interface to connect ROS based components to nodes using the HORSE middleware. For example, the user is allowed to use ROS hardware interfaces to communicate with the other HORSE components. It can be easily used to connect software and hardware components already integrated with ROS to the HORSE framework.

2.5.3 Interface to industrial equipment: HORSE-BOSCH Adapter

The HORSE-BOSCH Adapter (Figure 6), has been developed for the BOSCH use case and features a set of interfaces for communication with the Bosch industrial equipment: the Visual Control system, the conveyor belt and a beacon. The module provides support of etherCAT, PLC and OPC-UA. Additional protocols could be easily integrated.

The Bosch Adapter is a set of OSGi components deployable on a networked PC equipped with an EtherCAT Master Card and Java (for the OSGi framework).

Although the component is not necessarily applicable in every use case, it is a working example of integration of the HORSE framework with an existing infrastructure and control software of a factory. Thus, it can be used as a base for development of similar interfaces for different applications.

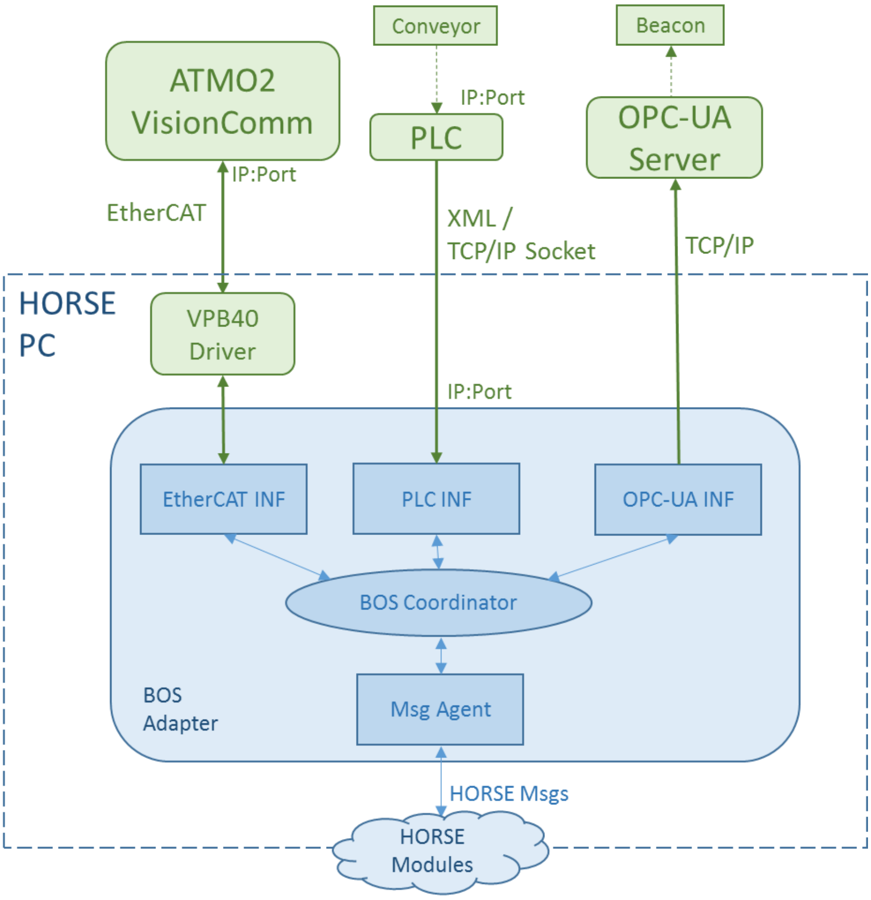


Figure 6: The Bosch adapter and Bosch machines

2.5.4 Specific components

2.5.4.1 Augmented Reality for assembly

The Augmented Reality (AR) for assembly component was initially developed for the TRI use-case. However, it has been already successfully transferred to other applications. The main purpose of the component is to display information which aims to improve on the one hand the efficiency and quality of work (e.g. assembly instructions) and on the other the safety and working conditions (e.g. safety zones). This is applied directly on the assembly table where parts are worked on and supports processing the input from the user (e.g. his or her gestures) and displaying this information on the table.

Using the component requires setting up a work cell consisting of an overhead projector and an RGB-D sensor (e.g. Kinect) used to track the motion of the operator. The proper operation of the component requires calibration of the work cell components relative positions and defining the overlays to be displayed as well as the reactions to user actions (e.g. using virtual buttons displayed on the assembly table). The component is fully integrated with the HORSE middleware messaging system.

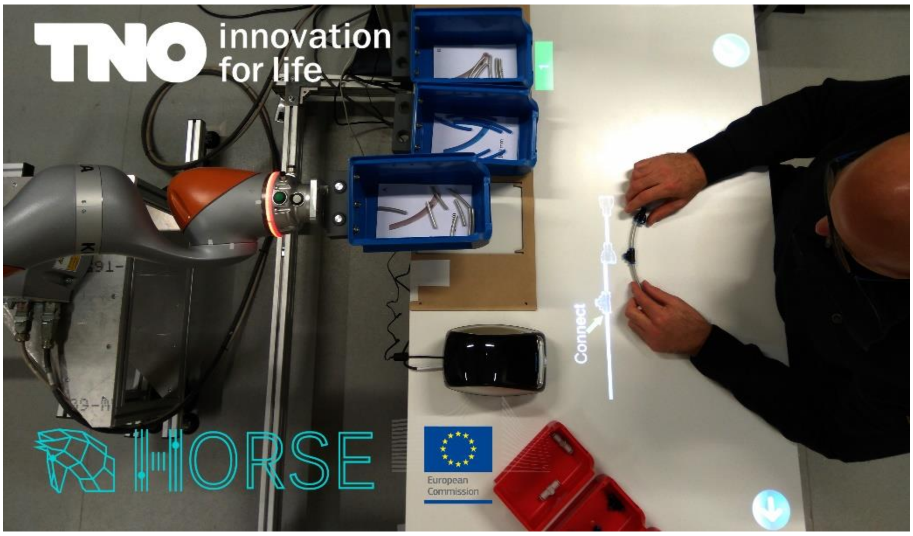


Figure 7: The AR for assembly

2.5.4.2 Augmented Reality for quality inspection

The Augmented Reality for quality inspection component was also developed for assisting the human operators of the Bosch factory in efficient visual quality check of the handled part. During configuration stage a number of projection "slides" can be defined, consting of the 3D position information of the target points and the shapes to be projected on them. During rendering stage the tartge locations are then calculated via TF and back-projected to the projectors optical frame to find the right pixel position and size for highlighting the defined POIs.

As inspection tasks in the BOS pilot case are executed in an unstructured environment (no table below the part) we also provide an additional touch-screen interface to display instructions to worker (e.g. an iamge on how the inspected POI is supposed to look like), and giving him the opportinity to make inputs.. The functionalities of the component are provided as a set of ROS nodes triggered via ROS actions from FlexBe. In case the robot is used to manipulate the part the robot control and AR are synchronized by the Hybrid Task Supervisor (Section 2.4.2).

Figure 8: An exemplary part with a control point (label) highlighted

In order to use the component in a different use case it is necessary to set up a work cell with an overhead projector, a camera and a robot arm in case the task includes human robot collaboration. This needs to be followed by an optical and spatial calibration of the elements of the work cell and setting up the overlays to be project and, again optionally, robot arm positions.

2.5.4.3 Collision detection and avoidance

The Collision Detection and Prevention ensures safety during any human-robot collaboration in a shared workspace.

This component can be used in every use-case that involves the need of a human operator into the robot workspace, in order to identify and avoid upcoming collisions and guarantee better efficiency fostering the robot to work in areas away from obstacles.

Factory automation has revolutionized manufacturing over the last years, but there is still a large set of manufacturing tasks that are tedious or strenuous for humans to perform. Some of these tasks, such as electronics or aircraft assembly, are difficult to automate because they require workers to collaborate in close proximity and adapt to each other's decisions and motions, which robots cannot currently do. Rather than automating such tasks fully (which may not be possible and/or cost-effective), HORSE consortium believes that human-robot collaboration enables safe and effective task execution while reducing tedium and strain of the human.

For example, mobile manipulators can supply different work stations with parts and perform standard assembly tasks, while human workers perform more complex tasks in the same workspace.

To allow for such shared human-robot workspaces in cluttered environments, robots have to be able to avoid collisions with static and dynamic obstacles while they are executing their original tasks. This involves both the monitoring of the robot environment to detect obstacles and the motion control that has to be able to avoid collisions while moving the robot along reference trajectories determined in a high level planning layer in order to fulfil the robot task.

At the basis of the HORSE Collision Detection and Prevention component is the GPU-Voxels framework that can be used for monitoring and planning applications in 3D and performs all computationally expensive calculations on the GPU. GPU-Voxels is a novel approach to live environment representations, in fact most similar approaches are not voxel-based and not capable of offering similar level of detail and response times.

This component allows the robot to automatically switch from its currently executed plan to a new one, when dynamic changes in the environment prohibit further progress towards the current goal, avoiding idle waits for the clearance recovery.

2.5.4.4 Situation Awareness

Smart factories could significantly increase production time and improve operators' working conditions in the manufacturing industry. They involve the collaboration without fences of robots and humans, whose safety needs to be ensured. Specifically, safety stops must be avoided because they may considerably slow down the production (safety protocol verification, re-launching the production line, etc.).

HORSE project provides a solution through a situation awareness mechanism to prevent from safety stops and adapt the agents' behaviours when a critical situation is detected.

The situation awareness mechanism of HORSE framework takes into account all the data related to the agents to predict a hazard, warn the operator and revise the robot's task accordingly. This module

is hardware independent and is configured with the agents and the sensors participating to the process.

2.5.4.4.1 Example use case

In a use case of deployment of a mobile base (AGV), one essential issue is to guaranty the safety of the operators who are in the same space of the robot. As shown in Figure 9 (on the left side) there is a situation where a collision may occur between a human agent leaving a work cell and a mobile base entering into the same work cell. The mobile base is able to detect collisions but this will lead to an emergency stop which will slow down the task. The situation awareness gathers all the data in the environment including the operator and the robot positions. The situation awareness mechanism adapts the robot behaviour to avoid a collision (scenario B on the right side).

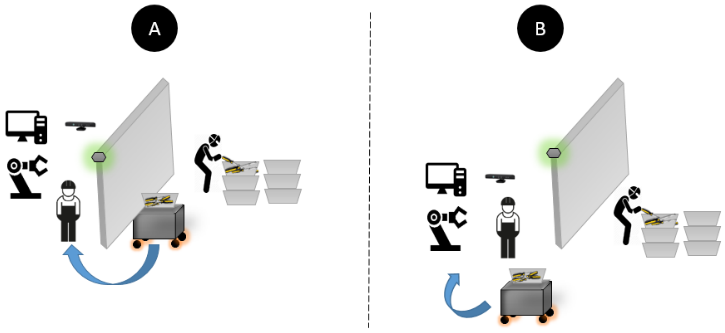


Figure 9: Example application of situation awareness

2.5.4.4.2 How does it work?

The situation awareness module (shown in Figure 10) is decomposed into two HORSE components: Event Processing and Global safety guard. The Event processing is able to detect critical events and the global safety guard relies on a reasoning system and a planner in order to generate a new action plan for the appropriate agents.

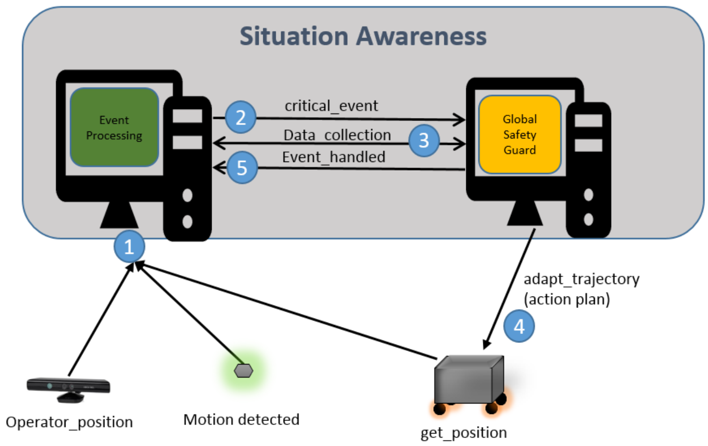


Figure 10: Situation awareness mechanism

1) Data are gathered from the devices and the agents participating in the work cell; 2) A critical event is raised whether an anomaly may occur; 3) Relevant information from the

environment is collected by the Global Safety Guard where a reasoning about the environment is done; 4) An action plan is generated to the concerned agents.

3 Creating a HORSE use case

3.1 *The approach to adopt HFW in manufacturing SMEs*

When an SME wants to adopt the HORSE framework the following approach should be followed: First , the technological scenario is discussed and the SME manufacturing challenges are analysed in detail. At the end of this stage, a scenario description is finalized as illustrated in Figure 11. This technological scenario is used to build a prototype to evaluate the value to the factory in real life conditions and based on that to create a business proposal, which includes the deployment of this technological scenario at a large scale with the help of an integrator. As the HFW is now demonstrated and promoted in the HORSE targeted regions by the five HORSE CCs, this entire process can take place there.

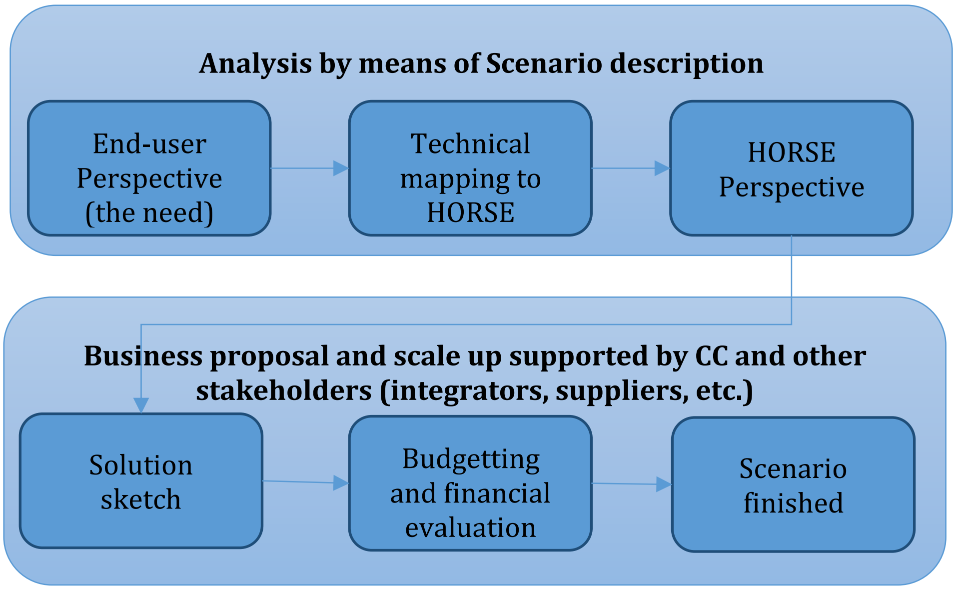


Figure 11: HORSE FWK adoption process for SME

3.2 *Template to analyse the use case*

HORSE has developed a template to support the analysis of use cases that is basically aimed to support SMEs to identify their challenges to be automated focusing on the cooperation between humans and robots with no fences in manufacturing applications.

In this template, the SMEs express the context of their scenarios and the expected challenges and outcome from HORSE. The technical mapping is then performed in order to select the relevant features that may answer to the identified problem. Finally, the HORSE framework perspective shows which are the modules from the framework that can be used to provide a solution.

In more detail the process is described below:

1. The first challenge in creating value with the HORSE framework is to write down your manufacturing process in a schematic manner and define which processes are not yet automated. Figure 12 shows an example of such a schematic, with two processes indicated as completely manual and one as mostly manual. In the example, the potential for improvement are located in those processes.

2. The second step is defining if these steps that are not yet automated are a problem or not. Returning to the example shown in Figure 12, the 'Distribute to customer' is completely manual, but this is not a problem because the company is satisfied with this as a manual process. However, the mostly manual 'Surface treatment' process can be improved with more automation. Problems can for example be in
   a. Inefficiency due to manual labour that is not 24/7, whilst automated machinery is
   b. High physical or mental loads on workers due to heavy lifting, uncomfortable body positions and stress due to the need to keep up with machinery
   c. Quality loss due to human errors

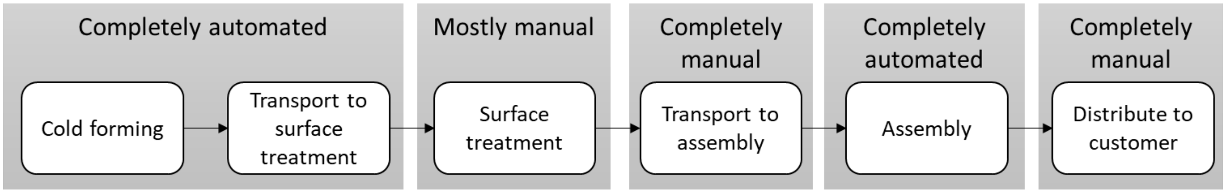


Figure 12: Example of a process schematic indicating which parts are automated or manual

Typically, in the use cases so far seen in HORSE, the problematic tasks have been found in the process flow between 2 automated tasks. The human is used to modify and transport the half-fabricates from process step A to process step C. Ideally this step would not exist, however it needs to exist to make the output of process A compatible with the input of process C. In the use cases so far discussed in HORSE this intermediate process B is generally highly complex, part specific, handling. This high complexity is difficult to automate and makes a human, with its flexibility, the ideal actor. This leads us to the next steps.

3. Define the inputs and outputs of this process step. (raw material, information from management systems, half fabricates, tooling, etc.)
4. Make a detailed analysis why this process was not automated yet (technical challenges, economic challenges, social challenges, …)
5. The next step is to see if the HORSE framework now has features to help automate this process step. This can for example be in:
   a. Easy information exchange and task allocation between the global (factory) level information system and the local (work cell) information system
   b. Allocating tasks to automated actors and human actors at the same time
   c. Allocating tasks to automated actors and human actors based on availability
   d. Smart robotic solutions
   e. Solutions to aid human actors in their task (to relief stress/strain)
6. Define scenario

If there are features of the HORSE framework that can relieve (part of) the problem, a scenario can be defined. Since most not yet automated processes are difficult ones to automate, the question may arise if partial automation can alleviate part of the problem and still can be of added value for your company. As seen in the use cases in HORSE, the processes that needed automation will not reach full automation at this stage. A solution is found in partial automation, where the human actor is just used for highly complex tasks and/or exception handling. But he is no longer under heavy time pressure and can focus more on quality. Most repetitive tasks of the original process step have been moved to automated actors. The trick in this step is to think outside the box and not "just replace the human

actor by a robot". The competence centres can help in defining such solutions. Inspiration can also be found at other manufacturing companies, fairs and literature[3, 4].

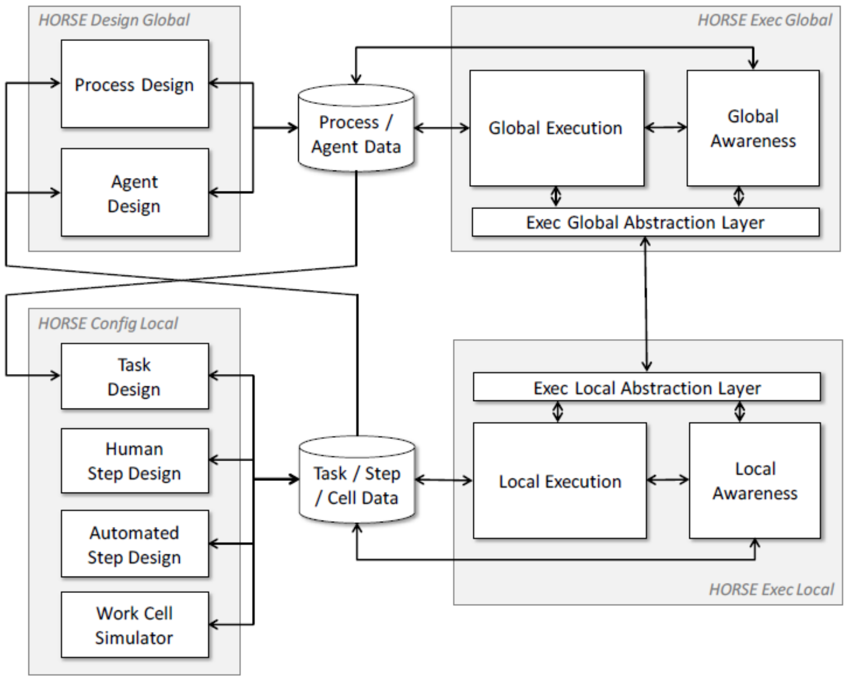
7. With the updated scenario of process B, the process flow made in step 1 should be updated. This process flow is the basis for the high level factory or work cell control in the MPMS
8. Assess the added value of the solution by a business model canvas
9. Re-evaluate the new scenario and identify what the differences are between the now sketched scenario and the "ideal" scenario. Repeat the process from step 2 until a satisfactory result has been achieved.

Table 1: Scenario Demonstration Template

<table>
<tr><td colspan="4">End-use perspective:
In the cells below, the end-user can describe his view on the problem at hand. This can be completed before going to the CC. The content is evaluated in the CC.</td></tr>
<tr><td>Context</td><td colspan="3"></td></tr>
<tr><td>Expected outcome/ Challenges</td><td colspan="3"></td></tr>
<tr><td>Expected benefits/ Impact
(From D2.1)</td><td colspan="2">☐ Safety
☐Production Monitoring
☐Comfortable working conditions
☐ Gain of productivity</td><td>☐Flexibility
☐ Cost Efficiency
☐Quality
☐ Cycle Time</td></tr>
<tr><td colspan="4">Technical mapping to HORSE framework:
In this section, the CC experts together with SME make the translation from a problem to a solution.</td></tr>
<tr><td>Task / User Story</td><td colspan="3"></td></tr>
<tr><td>Benefit</td><td>HORSE component</td><td>Description</td><td>Hardware</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">HORSE framework perspective:
The section below illustrates the use of the HORSE framework.</td></tr>
<tr><td>Component from the HORSE Framework
(Aggregation level 2)</td><td colspan="2">☐Process Design
☐ Human Step Design
☐ Agent Design
☐Automated Step Design
☐Global Execution</td><td>☐ Work cell Design
☐ Global Awareness
☐Local Execution
☐Task Design
☐ Local Awareness</td></tr>
</table>

[3] TNO. (2014). Increasing productivity and flexibility in manufacturing. Retrieved from TNO flexible manufacturing & assembly cases.pdf

[4] Operator-oriented Product and Production Process Design for Manufacturing, Mainte-nance and Upgrading. Retrieved from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjoiN2Y0fXaAhULmbQKHc6kCZsQFggqMAA&url=http%3A%2F%2Fpublications.tno.nl%2Fpublication%2F34625064%2FvFQsjF%2Frhijn-2017-operator.pdf&usg=AOvVaw1Q9u04F0Tgr-5E_4APQeHW

<table>
<tr>
<td>Architecture mapping
(explanation below)</td>
<td>
</td>
</tr>
</table>

Description of the features:

- **Process Design:** Design of the manufacturing process, i.e., what needs to happen in which order and with what requirements to the agents involved (role specifications).
- **Agent Design:** Design of manufacturing agents, i.e., the humans and machines (robots and other relevant automated machines) with their characteristics. In this scenario the module is used to define the roles and responsibilities to the human operator and to the robot.
- **Global Execution:** Supporting execution of manufacturing processes, i.e., making things happen; execution module developed within HORSE: MPMS.
- **Global Awareness:** supporting awareness about the global state of execution, i.e., observe things that happen and processing this into relevant signals for controlling execution.
- **Task Design:** Configuration of manufacturing tasks, i.e., the high-level activity spanning a work cell; note that this may require multiple agents of different kinds that each execute manufacturing steps. This module is responsible for the design of the coordinating the actions of the actors in a task.
- **Human Step Design:** Configuration of manufacturing steps, i.e., the low-level procedures performed by the human. This module is responsible for defining the human operator actions: providing additional parts and rearranging the rack.
- **Automated Step Design:** Configuration of manufacturing steps, i.e., the low-level procedures performed by the robot. This module is responsible for defining the robot actions: picking up the part and putting it on the rack.
- **Local Execution:** Supporting execution of manufacturing tasks and steps.
- **Local Awareness:** Supporting awareness about the state of execution, i.e., observe things that happen and processing this into relevant signals for controlling execution

3.3 Example of HFW application for the HORSE Pilots

3.3.1 OPSA challenges and benefits

The most important challenge for OPSA was considered the improvement of working conditions for their employees. This piece of process which OPSA recognised as important challenge to tackle is very hard: a manual labour intensive task in unnatural positions. This hard work within a difficult working conditions like dust, heavy loads makes it even harder. This results in big rotation of employees, as not so many people want to work there and while not being really well paid. The objective is to make workers' lives easier. HORSE application automatizes the cutting of metal casts, thus contributing to making this process faster, more efficient and most importantly improving working conditions of the workers. Another challenge is to organize the production in the automated manner covering the great variety of the over 900 products that the factory produces. Flexibility, re-configurability and safety are also important. The HORSE application will focus on the production organization for the various products, its direct interface (via OSGI and ROS) to the production agents (human worker and robotic equipment) and the orchestration of their collaboration. In parallel it is monitoring the safety for the operation during cutting metal castings by handling critical situations.

OPSA benefits from this use case by avoiding production stops for changing over products. Another benefit is precision. Current work is performed mainly manually so precision is not great. Parts are often rejected due to overgrinding or damages. In addition, robotizing these tasks is expected to improve product quality. Finally, the optimisation of production costs can be achieved by reducing cycle times and this is a dual benefit.

Figure 13: OPSA cutting task (before HORSE)

Challenges addressed:

- Automation of the casts cutting process, to increase the quality of the final part and flexibility by adopting learning by demonstration and productions processes which allow to re-use the relevant programmed trajectory for each product.
- Reduced level of defective parts
- Improved working conditions and reduce the risk of injury

3.3.2 BOS Challenges and benefits

Robert Bosch España, Fábrica de Castellet produces Wiper System assemblies (WSA). This pilot experiment is dedicated to the automotive industry and will be implemented in this assembling line (Front Wiper Systems - WSA). Depending on each car design, windshields come in different sizes and require appropriately sized front wiper systems to ensure efficient, reliable windshield clean. A single production line for WSA manufacturing is made by several working stations: about seven which are fully automatic, one semi-automatic and finally five manual working stations. The production line made by these workstations work in a sequence for assembling the different components making the WSA. The proposed pilot is focused on the last workstation – the final verification, which is one of the bottlenecks of the WSA production line. This process remain manual because of the worker flexibility, to not only check various parts in different places, but also because he can easily adapt to a huge quantity of combinations of WSA's, control features and containers layers.

This type of work puts workers in occupational hazards and health issues hurting their back, wrist, producing arm harm or hitting their fingers.

HORSE project provides a Robotic integrated system without fences, designed by the experts of Bosch, KUKA, TUE, TUM and TNO which picks up the product, automatically checks its quality and places it in the right way of every layer of the predefined box. The human and robotic agent collaboration is controlled and monitored by a Manufacturing Process Management System to ensure cycle time. In addition the safety is enhanced by the cell monitoring system of FZI.

The initial benefit offered by HORSE was the improvement of some of the occupational hazard and occupational health issues of the workers; avoiding wrist, arm harm or hitting their fingers thanks to the support of the robot manipulating a heavy product. This resulted in avoiding to exchange the workers every two hours in order to let them rest. Even the operator's complains due to repetitive risky movements to achieve the cycle time will be reduced. All of these improvements will increase the worker satisfaction which will have positive impact on the productivity.

In addition, the accuracy and the quality is improved because the artificial vision detect and record defects and every incident, during the automated visual control. This process assures the traceability of the inspection which is needed when any defect is detected by the quality department after delivery to the customer. The digitalization of the data will improve the quality of the information included in the factory's quality reports. Another benefit is the integration and orchestration of the four agents involved at the final work cell (conveyor, vision system, AR and Robot). As the HORSE architecture allows new agents to be added in the future, this increases flexibility of the factory. Finally, the digitalization of the process as a whole, will allow to program in advance the parameters and tooling when another product has to be produced, reducing the time to prepare the production line and as a result the production stop.

Challenges addressed:

- Automated packaging of Wiper Systems
- Operator's Safety and product quality with the required cycle time
- Flexibility for several product dimensions and weights as well for customer packaging diversity and new product variations.
- Solution robustness regarding: compatibility among systems and standard equipment

3.3.3 TRI Challenges and benefits

Thomas Regout International BV is a Dutch manufacturer specialized in the production and design of customized telescopic slides (see picture bellow) for several industrial equipment applications. The production processes of shaping, punching and assembly are semi-automated. Because of small batches sizes, operators have to change machines and tools over between different product types and operating tasks very frequently: this will take time and causes risks of making errors. Furthermore, they have to solve problems in a very short time and provide feedback on product and process design to engineers and management. This requires the right technology (flexible automation, overall production management), optimal human machine interaction as well as the right expertise, competences, team collaboration, insights in the production process and (instruction) tools.

The profiles of these slides are made out of steel coils. To prevent the profiles from corrosion they get a surface treatment. To get this treatment, the profiles have to be racked up one by one (see picture bellow).

Figure 14: Manually positioned products into frames for surface treament

The racking-up is done manually because of the high variety of the dimensions of the profiles. This type of work is labor intensive and has a high physical impact on the workers. HORSE intervention in the factory develops a demonstrator adopting the HORSE framework and a custom robotic application, which will rack up the profiles in an environment where humans and robots work in the same area. A robotic hand of KUKA has been adopted in a solution which have been designed by partner TNO and TRI experts. A manufacturing process model has been designed to monitor the production and allocate tasks in turn to worker and robot. The cell is monitored by sensors which alert the worker and the safety engineer when the risk is identified.

Business strategy of TRI is based on: Customer specific products/small order quantity/Quick response delivery/High Quality.

TRI is interested in adopting novel concepts as these enhance humans' own abilities and can add maximum value to the processes of TRI in a sustainable way. More specifically, HORSE supported the people of TRI in ergonomics, in understanding complex processes, in controlling complex processes, in more flexibility and also in lowering production costs by increasing the ability to do tasks first time right.

This in practice benefited TRI by addressing pilot projects in the areas of information technology and robotics to a network of experts. TRI would not have the knowledge nor the innovative power to run these projects by its own.

Second, by being member of HORSE the pilot projects are worked out by the consortium to the level of real built demonstrators. With the aid of these demonstrators, TRI can test in real production environment if these new techniques will work in the end. This is extremely important to get funding from investors to scale up to production level.

Third benefit for TRI is the fact that the pilots, resulting in the validation testing of the demonstrators are funded by the EU and do not claim TRI resources in terms of money.

Finally, being part of the consortium gives TRI a boost in terms of increase of knowledge and minimizes the financial risk of technological innovations which contributes to the long term continuity of the organisation.

Challenges addressed:

Support the business strategy of TRI by increasing flexibility on the shop floor and reducing production costs:

- Deployment of HORSE in TRI: visual management shop floor, MES is not enough
- P1: replacing Pick & Place unit by a Smart Robot without fences
- P2: replacing human labor by robotic agents to eliminate heavy load on human beings,
- Reduce high dependency on experience of tool preparation by introducing Augmented Reality

3.4 Demonstration of an end-to-end procedures in HORSE

An example of the execution time of the MPMS, message bus and components can be seen in this video titled "Manufacturing Process Management with Robot Task Synchronization":

https://www.youtube.com/watch?v=hD1vqzykLkU

4 HORSE HFW demonstrations in CC

4.1 Competence Centres in HORSE

Five Competence Centres are supported within the project: in France (Paris-Saclay, CEA), Germany (Munich, TUM and Karlsruhe, FZI), the Netherlands (Delft, TNO). The HORSE CCs demonstrate HFW coupled with their own technical capabilities and infrastructures. In this way, they promote HFW and their technologies and services to regional SMEs. The following presents how HFW has been integrated in the three initial HORSE CCs in specific scenarios, integrated with their components and infrastructures. The two new CCs – TCS and FZI, will follow similar process to establish their demonstrators.

4.2 HORSE demonstration *in Paris-Saclay Competence Centre at CEA*

CEA Competence Centre proposes several services and technologies to SMEs and provides them with access to HORSE framework components illustrated on innovative robotics use cases. In the following, the technologies and the components available in Paris-Saclay CC for HORSE and the use cases developed are presented .

4.2.1 HORSE framework in Paris-Saclay CC

HORSE framework is made available through different levels in Paris-Saclay CC as shown in the blue frame of Figure 15.

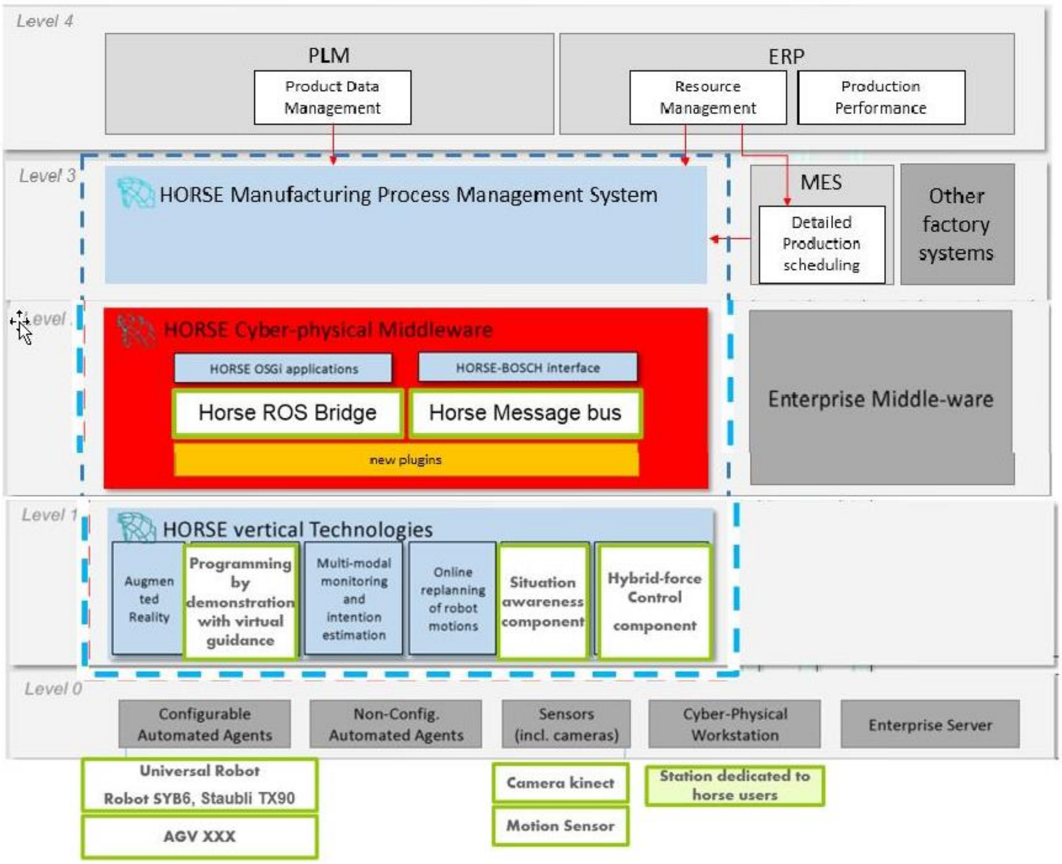


Figure 15: HORSE Framework in Paris-Saclay CC

The level 3 is related to the Manufacturing Process Management System (MPMS) which allows the visualization of the process, task assignment and tracking of the production.

The level 2 is dedicated to cyber-physical systems communication in order to standardize the communication format between heterogeneous automated agents (several robots, sensors, etc.) regardless their implementation language. HORSE ROS-Bridge and HORSE MessageBus have been developed by HORSE partners and deployed in Paris-Saclay CC for this purpose.

The level 1 proposes various HORSE components. In Paris-Saclay CC, the following components are already available and demonstrated in different scenarios at Paris-Saclay CC premises:

- Programming by demonstration component enriched with virtual guide and hybrid-force control component (both components are described in the deliverable D3.1). Those components are demonstrated on a use case for cutting foundry parts. The goal of using those components is 1) to ease the programming of a task using a collaborative robot. 2) to reduce vibration and friction during the execution of the task.
- Situation Awareness (described in both deliverables D3.6 and D3.7) is also illustrated on a scenario for parts transportation between two work-cells.

Finally, Paris-Saclay CC's equipment is compliant to HORSE framework. HORSE components available in Paris-Saclay CC have been tested and integrated with the UR10 arm, the Isybot SYB6 collaborative arm and the Staubli TX90. Regarding the sensors, the Kinect camera and the motion sensor are able

to send data and to communicate with other HORSE components through HORSE MessageBus protocol.

Paris-Saclay CC has also made available a station dedicated to HORSE users that will be mainly used for open calls experiments.

4.2.2 HORSE scenarios in Paris-Saclay CC

The following scenarios are deployed in CEA premises and illustrate HORSE components described in the last section.

4.2.2.1 Cutting foundry parts

In this scenario, the operator teaches the robot the trajectory to follow for cutting a foundry part (as required by OPSA challenges (see chapter 3.3.3)Then, the robot executes the cutting autonomously. The HORSE level 0 components used in this scenario are programming by demonstration with virtual guide and hybrid-force control. A full description of those components can be found in the deliverable D3.1: Hybrid position/force control of articulated flexible structures used in contact with the environment (see at the www.horse-project.eu).

For the teaching, the operator shows the robot the following steps as shown in Figure 16:

1. The starting point
2. The cutting trajectory shifted.
3. The target point.

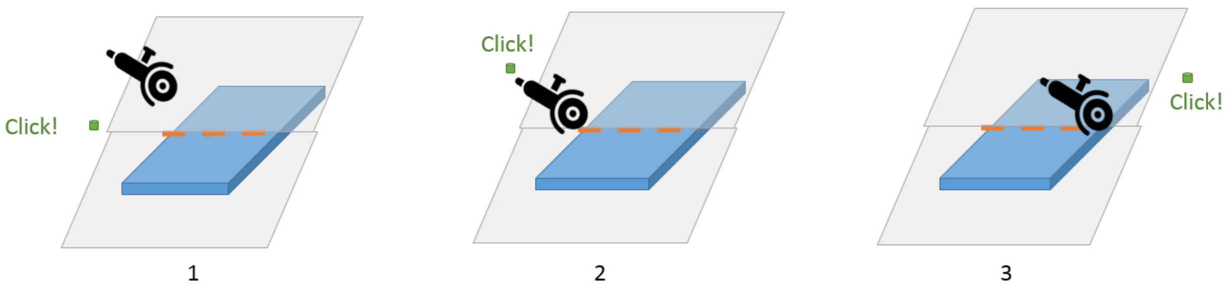


Figure 16: Programming by demonstration steps for cutting

Figure 17: Cutting use case in Paris-Saclay CC

Then, in the execution phase of the task, all the parameters from the teaching phase are recorded into the planner: the robot, the tool orientation and the positions. The trajectory generator is responsible for generating set points while taking into account those parameters. Hybrid-Force control component is also used in this phase for friction reduction and vibration dumping.

Figure 17 shows this scenario demonstration in Paris-Saclay CC with the StaubliTX90 robot.

4.2.2.2 Transportation between several work-cells

The following scenario describes a use case which involves different robots and operators sharing together the same workspace in advanced-manufacturing situations. In order to ensure the operator's safety and avoid emergency stops, the situation awareness component was used, which relies on contextual data and adapts the robot behaviour accordingly. The situation awareness components are fully described in the deliverables D3.6 and D3.7.

The Figure 18.A shows a scenario where the mobile base is transporting parts from one work-cell to another. Two operators are doing different tasks. One operator is working with a manipulator and another operator is doing a manual task. The situation awareness component gathers all the data in the environment including the operators' and the robots' positions and is able to adapt the robot behaviour to avoid a collision. Figure 18.B shows that the mobile base was informed that a possible collision may occur because the operator is moving from his work-cell to another one. The mobile base waits then in a safe position (Figure 18.C) until the operator is far enough. The mobile base continues then its way to the target position.

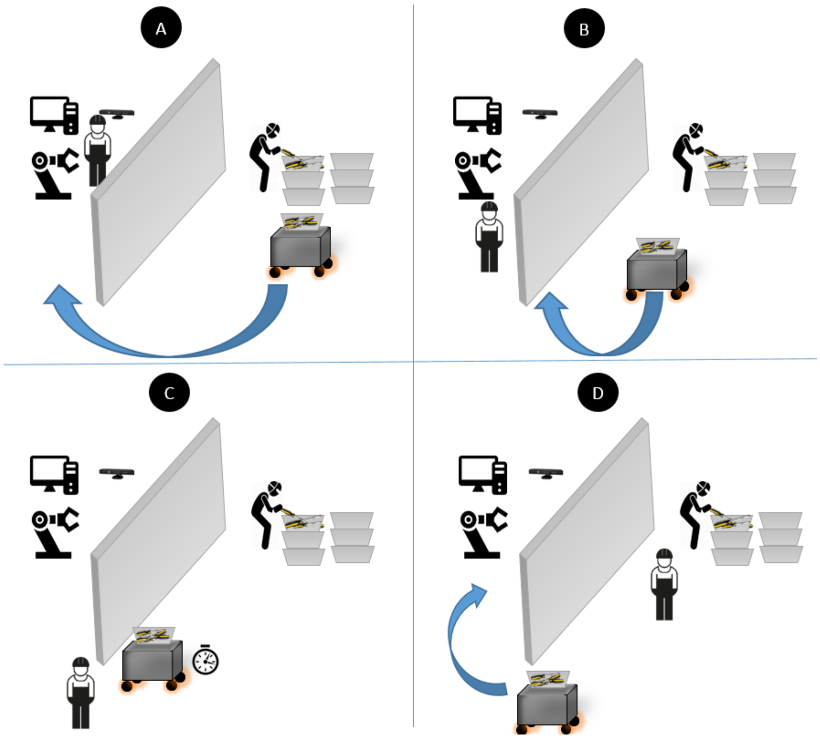


Figure 18: Schematic representation of transportation use case

(This scenario has been demonstrated in CC premises as shown below)

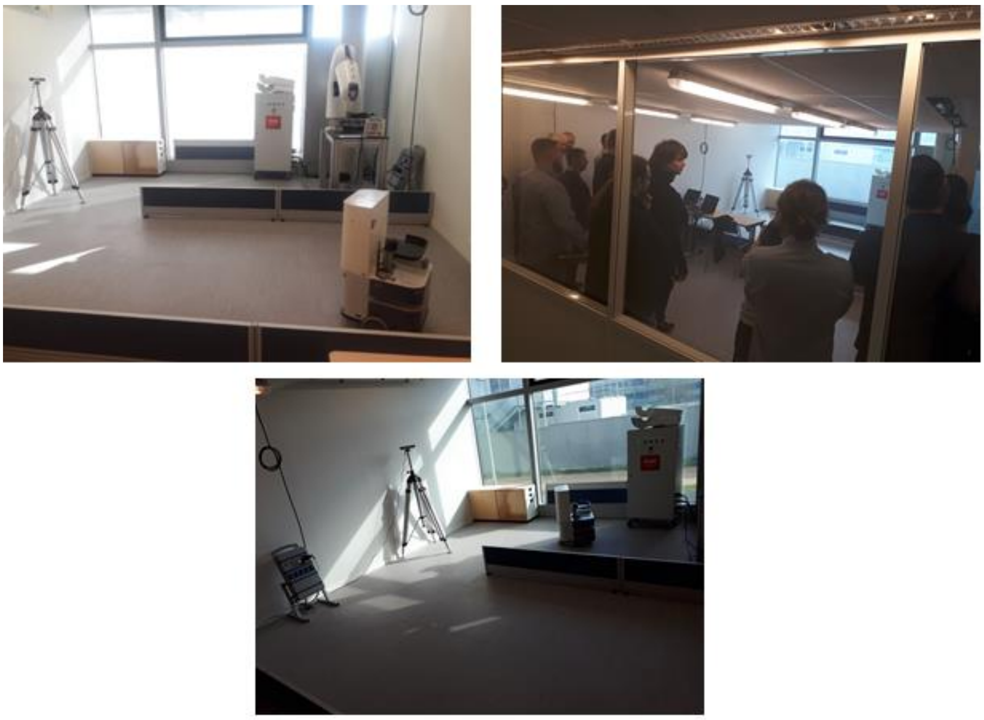


Figure 19: Transportation use case in Paris-Saclay CC

4.2.3 HORSE framework in Munich TUM Competence Center

4.2.3.1 Description

The Collaborative Work Cell (CWC) offers opportunity to quickly develop and test of human-safe object manipulation tasks. The user can program the trajectories of the collaborative arm using a simple, intuitive interface. The trajectories and actions are then integrated into the task descriptions using FlexBe. Additionally, the user can set-up the AR overlays which can be displayed in the robot's environment or on the objects held by the robot which facilitates development of collaborative task such as visual inspection or assembly. Finally, CWC is integrated with the SAPARO tactile floor, overhead RGB-D camera and a laser scanner which are connected to the local situation awareness module and can be used to trigger appropriate reactions from the system.

4.2.3.2 Equipment

The heart of TUM CWC is formed by a KUKA LBR iiwa 14 R820 robot arm. It equipped with torque force sensors on all of its seven axis and capable of lifting up to 14kg of payloads. It is connected to a KUKA R800 parallel gripper.

Our sensor system is composed of a Kinect RGB-D camera, a 2D laser rangefinder and pressure sensitive floor. The used laser scanner is a Hokuyp UTM-30LX-EW having an angular resolution of 0.25° and a reach of up to 30 meters, making it suitable for real time data collection. Moreover, the floor is a unique prototype developed by Fraunhofer IFF in Magdeburg. It offers a resolution of 40 by 40 tiles (12.5cm x 12.5cm each), that can measure 1,024 different pressure levels. The technology has been taken over by Pilz GmbH and will be certified and publicly available soon.On top we have other optical sensors available for tryout including the whole Intel Realsense family.

For demonstrating the HORSE AR components we are using an EPSON EB-1980WU full HD beamer, mounted right above the robot arm.

Beside of our workcell TUM's Competence Centre offers a lot of other state-of-the-art hardware, featuring:

- Stäubli TX-30 robot arm
- Säubli TX-90 robot arm
- 4x ABB IRB 100 robot arm
- A car testbed for electronic cars

4.2.3.3 Risk analysis

When setting up the CWC we always had in mind that to allow visitors of our Competence Centre to explore the potential of our robot without being limited by additional safety constraints. Thus choose a robot arm and gripper that are both intrinsically safe and certified for collaborative use cases were used.

On top a wide range of sensors allow to test what level of safety is actually possible with modern technologies, while always having safe hardware as a secure bottom layer.

4.2.3.4 HORSE Framework mapping and validation

TUM CWC is focussed but not limited to demonstrating the local layer of the HORSE framework. All hardware is completely integrated and controllable via HORSE Local Execution and HORSE Local Safety. Moreover all necessary components have been developed to allow easy teaching of robot trajectories and AR guided human tasks from HORSE Local Configuration.

4.2.4 HORSE framework in Delft, TNO Competence Centre

4.2.4.1 Description

The work cell for augmented-reality-and-robot-assisted-assembly (derived from the TNO use case) shows the potential of relieving stress and strain form the human actor by assisting him in two ways. On the one hand the strain on the worker is reduced because repetitive work of gathering parts is taken over by an automated system. On the other hand the stress on the worker is reduced because he is guided in the complex assembly task by an augmented reality system, helping the worker with doing the steps in the right order and assuring quality. The demonstrator was developed in the competence centre in Delft. After development it was shipped to end user TRI and is currently operational in their factory.

A schematic representation of the work cell is shown below. The following were the main factors that contributed to the complexity of this use case:

- Large number of part in the bin storage
- Selection of bins in the assembly station
- Robot moves bins between the storage and the assembly station
- AR work instructions
- AR projection safety zones

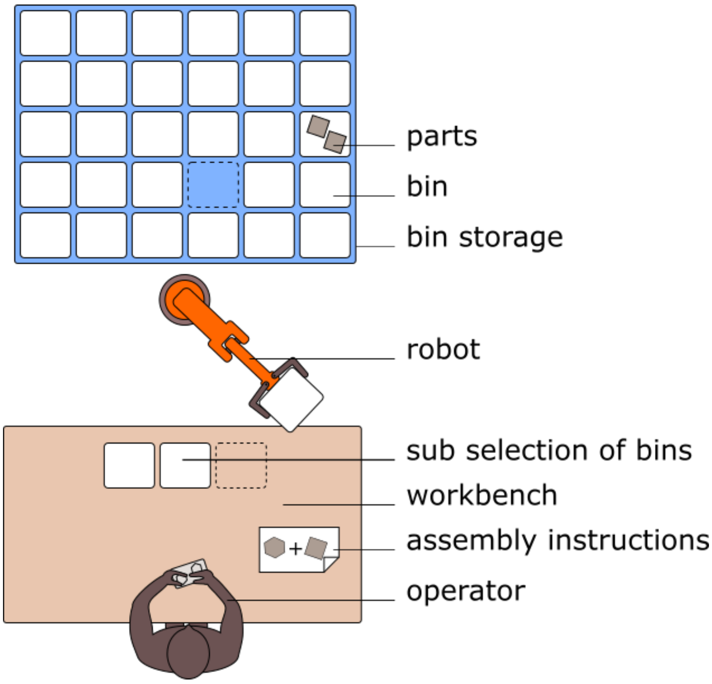


Figure 20: Schematic representation of the robot and augmented reality assisted work cell

The following components of the work cell get tasks via the HORSE MPMS and communicate with each other via the HORSE message bus:

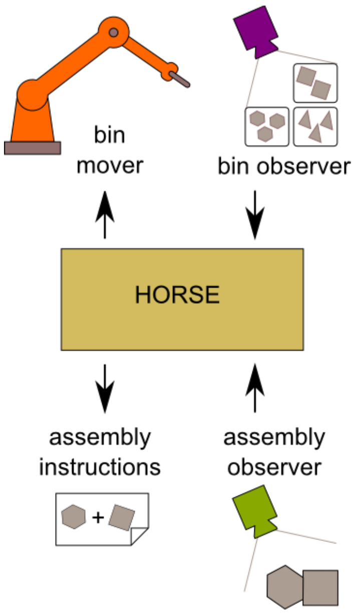


Figure 21: Components and position of HORSE in this demonstrator

- HORSE Framework
- Bin mover: moves bins between desired locations
- Bin observer: observes the current locations of the bins
- Assembly & safety instructions: instruction for operator (AR)
- Assembly observer: observes the progress of the assembly
- Horse software synchronizes robot actions to operator demands

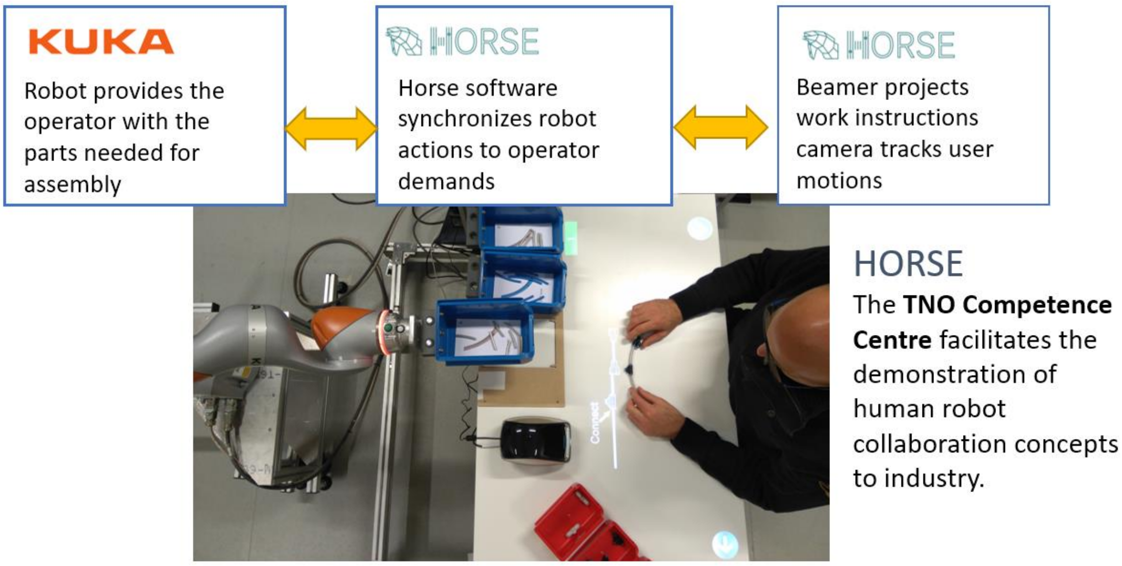


Figure 22: Realized demonstrator

4.2.4.2 Equipment

The main equipment consists of 3 parts.

1. Augmented reality system to guide the worker including beamer, camera and 3D camera and PC
2. Robotic arm (Kuka LBR iiwa)
3. PC for the MPMS system and message distribution to the worker, AR system and robotic actor

4.2.4.3 Risk analysis

For the current use case intrinsically safe hardware has been used to allow the robotic actor, human and AR system to use the same physical area without limitations. If needed this use case can be

performed without intrinsically safe hardware as well by adding a safety system on top to mainly ensure the human safety in the robotic area. To allow for this the human working area and robotic working area have, only where needed, minimal overlap.

4.2.4.4 HORSE framework mapping and validation

The demonstration scenario in the CC includes the global level MPMS in runtime for task distribution to the local level actors. On the local level the focus is again mainly on the runtime part for the human, robot and AR system. The local level design time of the AR is included as well as is local level design time of the robotic actor to allow quick reconfiguration if the work cell layout changes.

5 Step-by-step implementation process

Starting with the outcome of chapter 3.3 (Example of) being the process how you want your process to look like. The implementation of the HORSE framework can start.

The implementation covers several steps.

1. Start with the process model
2. Defining work cell agents and the
3. Defining interfaces among agents.
4. Developing interfaces and continuous integration
5. Testing the entire system
6. Component optimization
7. Addition or removal of work cells, agents/modules

5.1 Start with the process model

Having a high-level overview of the process is valuable for the integration process to keep focussed on the overall system. Component optimization itself is the last step.

Design the process in the MPMS as envisioned in chapter 3. This may take a few iterations between the engineers and the factory The result is a logical and rather simple process without many dependencies. This is a tough task which can be supported by the HORSE experts and Competence Centres.

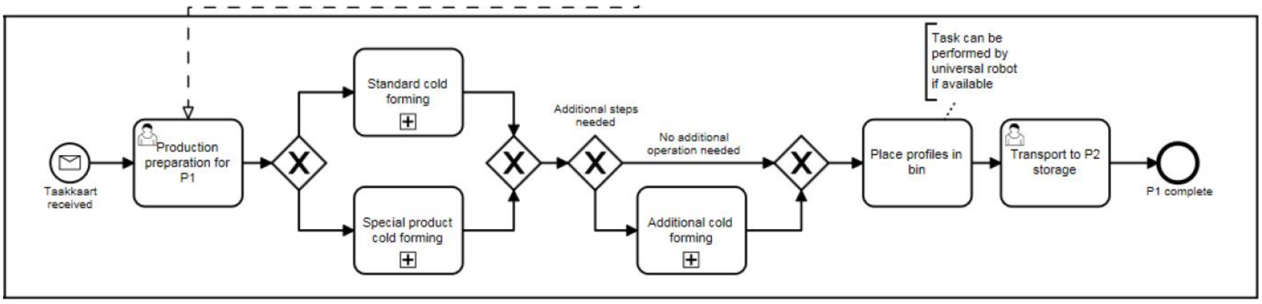


Figure 23: Example of a MPMS flow used to depict part of the production process

5.2 Work cells and HMMw component

Now the process is clear. Work cells can be defined with their corresponding agents. This might cause some redesign of the MPMS process to simplify the work cells and optimize the synchronisation. At the end of this step it is clear what actors are in a work cell, which process(es) take place and what is their execution sequence. For proper reference, each actor should receive a unique name. From this point onward deployment diagrams can be drawn including the HMMw component for task distribution and communication of components in a work cell (to be presented in the following step)

In case the number of actors is limited and they are operating close to each other (e.g. within only one work cell), it is enough to have just one HORSE Broker and no HORSE Dispatcher. An example of such deployment is the Bosch pilot case at Castellet, Spain (Figure 24).

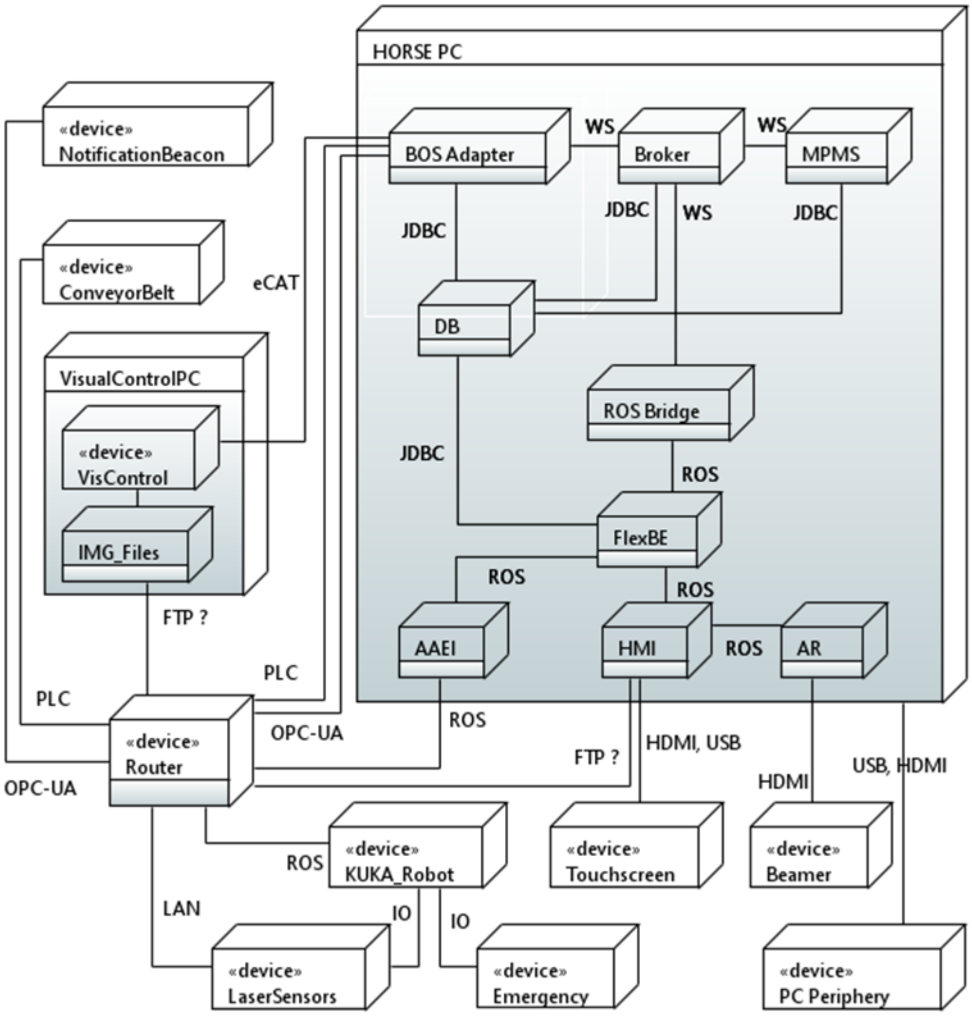


Figure 24: HORSE Deployment at Bosch Castellet, Spain

In this case we have all HORSE software modules (from HORSE Exec Global and HORSE Exec Local domains) installed on one HW platform – the HORSE PC. The industrial equipment is connected either over Ethernet or special wiring.

In case of a bigger setup we will have a distributed deployment in with multiple instances of HORSE Exec Local in each work cell and one instance of HORSE Exec Global. Each of these instances are equipped with their own HORSE Broker as introduced in Section 2.5.1and shown in Figure 5). The Broker could be installed on the same computer with other HORSE components of the given work cell or on a dedicated one (e.g. Raspberry Pi). All brokers should be mediated by a HORSE Dispatcher.

An example of such deployment is realized at TRI pilot site (Figure 25). In this case the MPMS, the DB and one Broker are constituting the HEG (HORSE Exec Global) PC. For simplification, the HORSE Dispatcher is installed on the same PC, but this is not necessary. There are three work cells (P1 Tooling, P1 Stacking and P2 Hanging). The HORSE components of these work cells constitute three instances of the HORSE Exec Local. Each work cell is equipped with a Raspberry Pi featuring an installation of HORSE Broker and enabling the communication of the actors of this work cell (e.g. AR and Robot) with the HEG and other work cells. Because of physical separation of the work cells, additional routers are deployed in order to provide the low level networking. The diagram shows

details of HEG PC and one of the work cells (HEL P1 Tooling) with placeholders for the other two work cells (HEL P1 Stacking and HEL P2 Hanging).

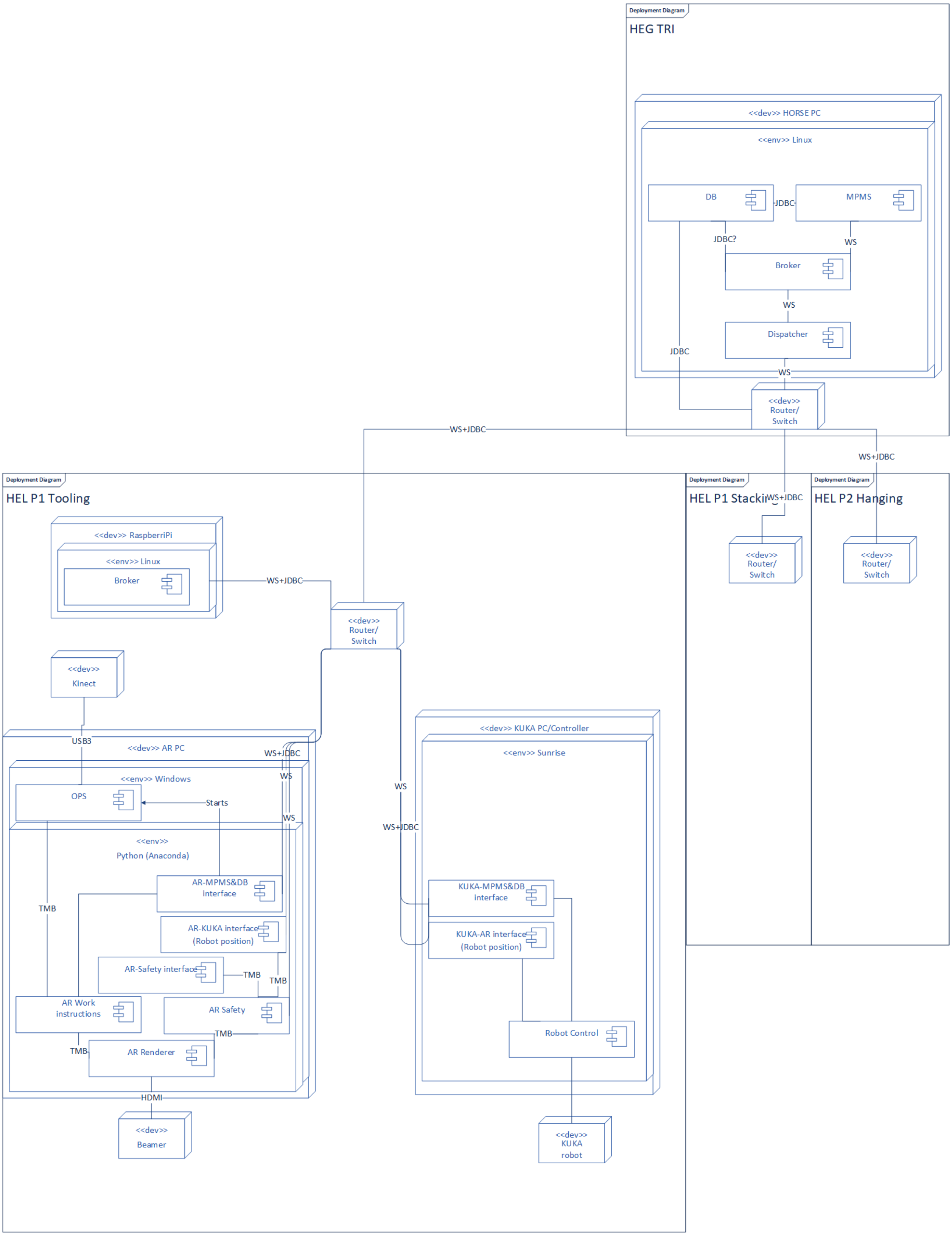


Figure 25: Multiple work cells deployment (TRI pilot site)

5.3 Defining interfaces among agents

Having defined the actor and the work cells, their physical and communication interfaces can be elaborated as well as their deployment within the production environment. For each agent the inputs and outputs should be specified. These in- and outputs can be used within the work cell, to communication with another work cell or with the global/MPMS process.

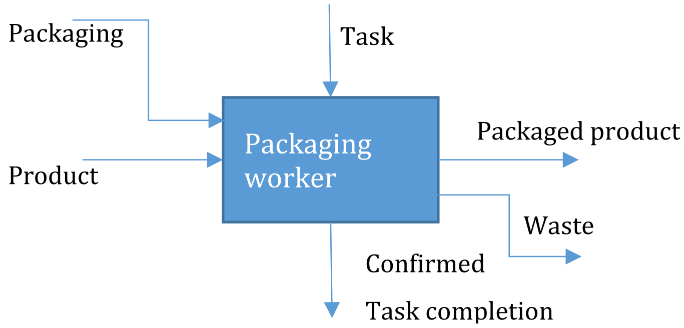


Figure 26: Example of a simple input-output design of a packaging agent's task

5.4 Developing interfaces, continuous integration and integration testing of the software

The interfaces should be developed in an iterative cycle in which components are integrated one by one and tested for their communication between them. Dummy components,which do not include their functions already implemented but only example input and output values, can be used for the not yet developed modules and agents to facilitate that the high level MPMS process can be executed properly, for testing purposes . This stepwise approach ensures that unforeseen issues and or other dependencies are identified early and can be solved before integrating the whole system.

5.5 Testing the entire system

When the integration is completed the entire system can be tested. The MPMS shows a clear overview of which agent/ module is/should be working on which task. The clear foreseen separation between the Local/Global level and the Design/RunTime aspects aids in debugging the right parts and keeping a maintainable system that is future proof (e.g. extensions can be added later and actors/robots can be replaced by other actors/robots).

5.6 Component optimization

When the entire system is operational, there is always room for optimization. The parameters related to the exact product being produced are abstract throughout the HORSE framework. This leads to the situation that machines operations can be optimized without having to adopt the overall process. Machine timings and speeds as well as parameters relevant to the quality of the production can be adjusted. It is recommendable to add these parameters to the interface definition of the component when identified such that they can be adjusted from MPMS level. This aids in fast and good change overs between different products.

Production lines are always evolving for more performance and more flexibility and different products. By having as much as possible data like "calibration data", "machine settings" and "product

data" stored in linked data bases, production likes can be reconfigured easier and are less prone to error during reconfiguration.

5.7 Addition or removal of work cells, agents/modules

Adding actors to a work cell or even adding work cells can follow the same approach. First, the work cell or actor is added on MPMS level and its components step-wise integrated.

Removal of an actor or work cell is just as easy. It is recommended to make a dummy actor and let that run in the process whilst the physical actor is removed from production. At any stage convenient to the user, the MPMS program can be updated such that the work cell or actor is removed completely.

6 Deployment and configuration of the HFW

The next sections describe the way to adopt the HFW towards the specific challenges that will be targeted. The following shows how the HORSE FWK has been implemented in the pilots and Competence Centres. This illustrate how the guidebook comes as a result of the CCs scenarios deployment, the D7.1.2: Scenarios for Competence Centers.

6.1 Installation of the HFW

The installation of the HFM is addressed in D4.5. An illustration of the MPMS and HMMw installation is given in the next chapters.

6.2 MPMS installation

The MPMS is the collection of subsystems responsible to orchestrate the tasks of agents in the manufacturing processes. Orchestration is dependent on the design of the processes and agents. The MPMS includes the functionality to design processes and agents on one hand and execute the processes by assigning activities to agents on the other hand. It covers the modules of HORSE Design Global and the Global Execution module of HORSE Exec Global of HFW software Architecture.

The technology realization of these modules (MPMS) is achieved with an open-source Business Process Management System called Camunda BPN[5]. Camunda is highly extensible and includes native support to model and execute international standards by Object Management Group, namely Business Process Model & Notation (BPMN2.0), Decision Model & Notation (DMN1.1) and Case Management Model & Notation (CMMN1.1).

Detailed presentation of Camunda and how its software architecture maps into the HORSE software Architecture can be found in HORSE Deliverable D3.5. Here we describe the main steps to install MPMS as part of HFW (in a high-level way, while full technical details can be found in HORSE Deliverable D4.5).

6.2.1 Camunda software installation

The main components to install are the Modeler, in which users can model their processes, and the Process Engine, which is responsible to enact the modeled processes with task allocation to agents.

Modeler

The standalone application for modelling processes is available on Camunda Modeler.

- Download the appropriate for your OS .zip file
- Unzip in a folder in your pc
- Run camunda-modeler.exe

When you create a process in the Modeler (a quick guide can be found on Modeling a BPMN 2.0 Process), the "Executable" property should be checked (see Figure 27).

[5] https://camunda.com/

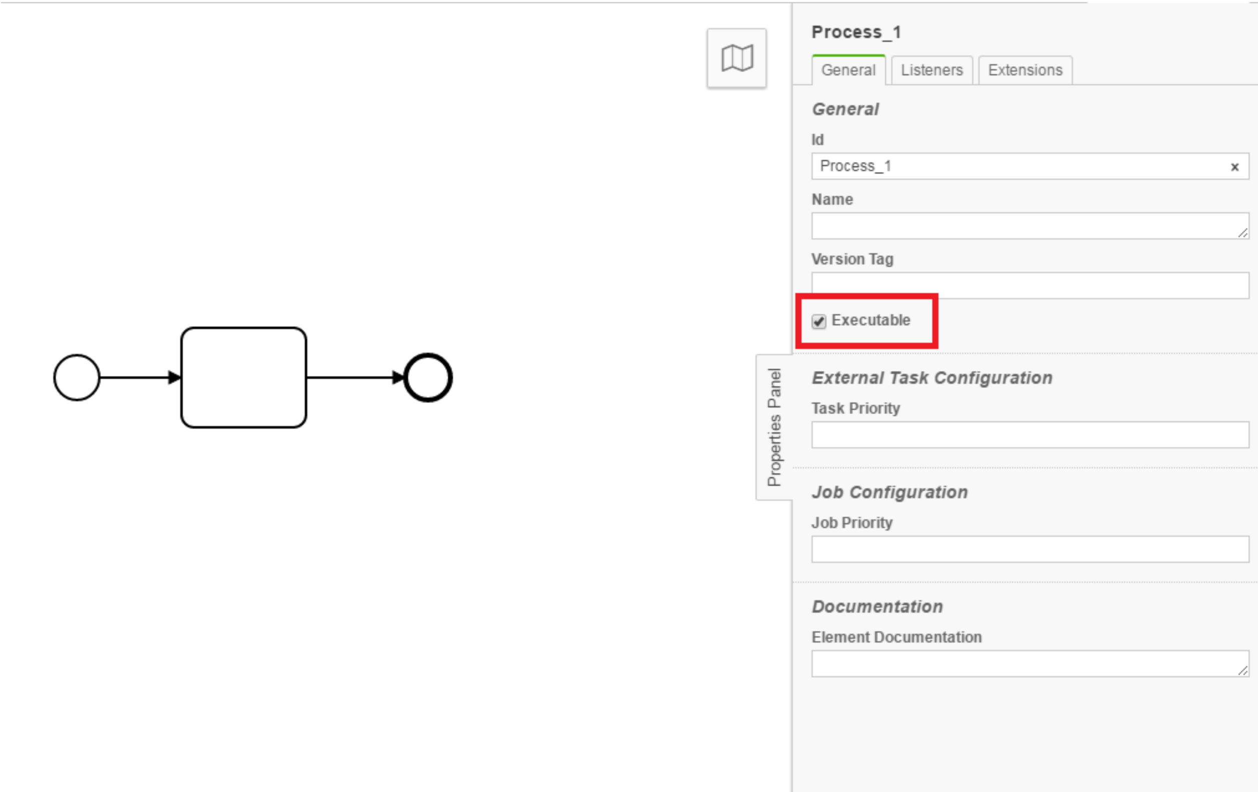


Figure 27: Modelling a process in the Modeler

Process Engine

You can download one of the three full distributions, available on Camunda Download.

A full distribution of Camunda bundles:

- Process Engine configured as shared process engine
- Runtime Web Applications (Tasklist, Cockpit, Admin)
- Rest API
- Container / Application Server itself

For HFW we use Camunda BPM 7.7.0 – WildFly Distribution. After downloading, simply unzip the file inside a directory of your choice. We will call that directory $CAMUNDA_HOME. Start the engine by executing the script file "start-camunda.bat" (for Windows users), respectively "start-camunda.sh" (for Unix users). This script will start the application server and open a welcome screen in our web browser (see Figure 28). If the page does not open, go to http://[host_ip_address]:8080/camunda-welcome/index.html. You can login with demo/demo.

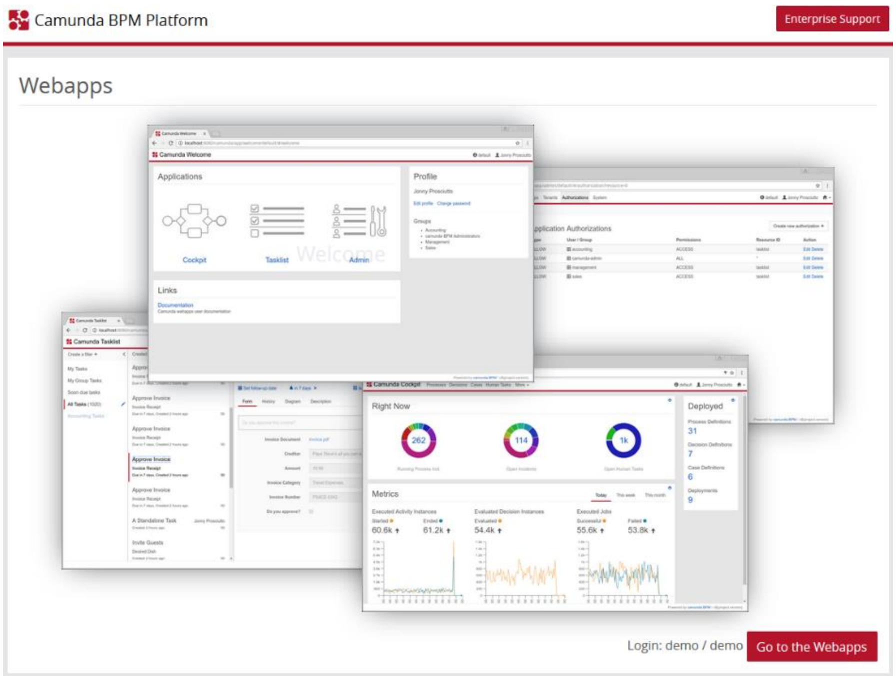


Figure 28: Camunda welcome screen

6.2.2 MPMS integration to other HORSE modules

Installing Camunda as described above does not add much value without integration to the other modules of HFW. The main interfaces are the connection to the middleware (so communication to modules of the local level is enabled) and to databases.

Integration to the abstraction layers

MPMS is considered as a websocket client, as shown in Figure 29.

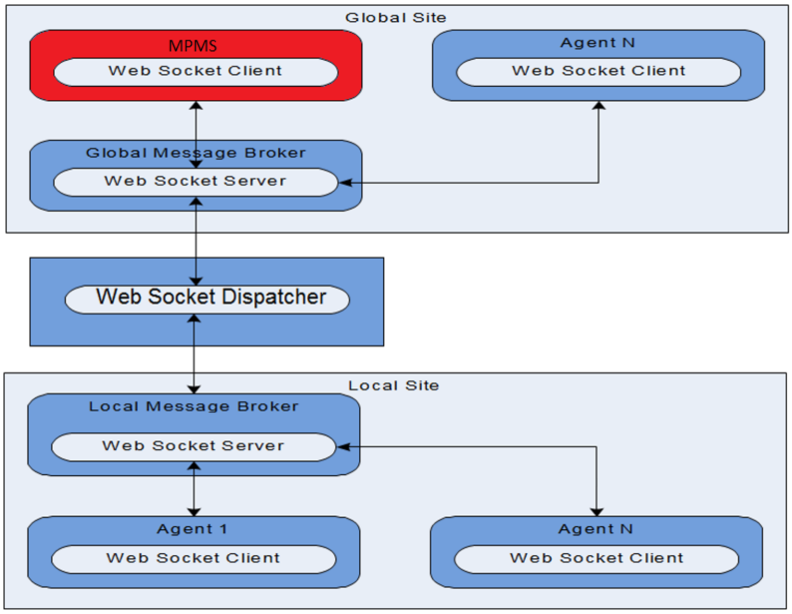


Figure 29: MPMS websocket client node

From the MPMS side, when modelling a process, we need to have a BPMN "Service Task" to initiate the opening of the websocket connection to the Message Bus which is kept active until the whole process is completed (just before the end of the process, another "Service Task" triggers the closing of the websocket connection with a delegate code). For that, we use the Java API for WebSocket JSR 356, which defines a standard API for creating websocket applications. We implement a @ClientEndpoint POJO, which is responsible to open/close a websocket connection (we only need the IP address and the port number of the Message Bus server to register MPMS as a client node) and send/receive messages. Messages are represented by JSON messages, the complete syntax of which is described in HORSE deliverable D3.12.

Integration to DBs

In the default configuration of the Camunda distribution, the database schema and all required tables for the process engine are automatically created in an H2 database when the engine starts up for the first time. You may also use this internal DB to setup table definitions for your application/business purposes, like for instance Product Def. DB, but the information stored there will not be easily accessible and shared by other modules of HFW architecture. Since Camunda supports various DB products you just need to configure the database that the Camunda engine will use. This can be done either by defining the JDBC (Java Database Connectivity) properties of the database or use a javax.sql.DataSource implementation (e.g. DBCP (Database Connection Pool) from Apache Commons).

For our HORSE MPMS developments we use a PostgreSQL JDBC driver to connect to a PostgreSQL DB Server. In that server we can setup our "HORSE-application/business" databases, like for instance the Product Def. DB or Agent Def. DB.

The JDBC driver shall be put under *"$CAMUNDA_HOME\camunda-bpm-wildfly10-7.7.0\server\wildfly-10.1.0.Final\modules\system\layers\base\org\postgresql\main"* (if folders don't exist, create them). Also, configuring the process engine ("standalone.xml" file of the Camunda distribution found under *$CAMUNDA_HOME\camunda-bpm-wildfly10-7.7.0\server\wildfly-10.1.0.Final\standalone\configuration*) is required for the DB integration (full details in HORSE Deliverable D4.5).

6.3 MPMS configuration

The steps described in the above section are the default steps to install MPMS as part of HFW. A running Process Engine with integration to the middleware and databases is essential to use MPMS but process applications developments and configurations based on user's situation and requirements should be done to exploit the benefit of it.

First of all, you start with modelling the processes. This is something that the end user should do since he has the knowledge of its factory workflows. These process models represent the sequence of activities of a manufacturing process and the allocation of tasks to specific agents. All process and task information (e.g. procerss_name, task_name) is stored in Process and Task DBs.

Agents can be designed in the Agent DB of HFW architecture. Currently, this DB has a core schema of describing agents (e.g. agent_name, agent_abilities, etc.) but more attributes can be added by the end user.

The next step after modelling the process and designing the agents is to make the models executable. A process model itself has some business logic behind but the execution of it requires a process application in which delegation code will execute this business logic. A process application is implemented in any Integrated Development Environment (IDE) where you can set-up a project and

then build it. Building refers to the creation of .war files which are then deployed by the process engine of Camunda.

Eclipse is an option, you can download Eclipse IDE for Java EE Developers from Eclipse Downloads. Instructions on how to setup an Eclipse project can be found on Create a new Maven Project in Eclipse and as a video on Camunda's Development Environment Tutorial.

Camunda provides several project templates for Maven, which are also called Archetypes. They enable a quickstart for developing process applications using the Camunda BPM platform. We use such an archetype for developing our HFW MPMS process applications (the ones of the three pilot cases).

The developed process application should feature the following (bare minimum):

- Service Tasks (or Script Tasks) to retrieve Task information from the DB (or any other info from other DBs)
- Service Tasks to open a web socket connection, register MPMS to the Message Bus (middleware) and to close the connection at the end of the process
- Business Rule Task (or Script Tasks) to select the best agent for a task
- Exception event in case of a task failure
- Delegation code attached to AutoAgent tasks (ExecutionListeners) to send messages to the local level of HFW through the message bus

Such a process including the above would look like Figure 30:

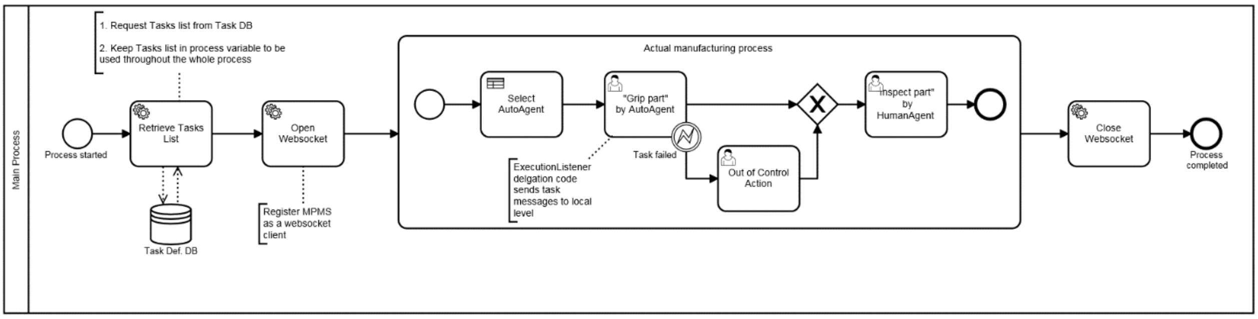


Figure 30: Executable Process Model with all MPMS functionality

This process model, with the right code written in IDE, is a typical process application which the end user should develop. The deployment of the generated .war file from IDE is done by simply copying the file from *\the target* folder of the (Maven) process application project to the *$CAMUNDA_HOME\camunda-bpm-wildfly10-7.7.0\server\wildfly-10.1.0.Final\standalone\deployments* folder. The running process engine will pick this .war file up and make it "executable".

6.4 HMMw adjustment

The installation of the HMMw components (one or more Brokers and a Dispatcher, if multiple Brokers) is described in HORSE deliverable D4.5.

The Brokers should be configured to communicate with the History DB for logging the incoming messages as explained in Section 2f of Appendix of D4.5. The IP addresses of the brokers should be made known to the components of the respective work cell or functional group. The Dispatcher should be connected to each Broker by creation of service configuration as explained in Section 2g of Appendix of D4.5. The name of the Broker should be used in the addressing of the HORSE messages.

Then, each actor should be registered to the Broker of the actor's work cell and start exchanging payload messages with the other actors.

Adding a new work cell requires the deployment of a new HORSE Broker instance and creation of new configuration in the Dispatcher settings. As a next step the individual actors of the new work cell can be registered to the Broker and start exchanging messages.

The removal of individual actor is done by unregistering it from the local Broker. Then the physical removal could occur.

The removal of a work cell with its actor is done by removing the configuration of the needed Broker from the Dispatcher settings.

The HMMw allows listing of the registered actors or looking up the status and registration parameters of a given actor.

6.5 AR adjustment

After installation of the AR environment, the AR communicates via task ID's with the MPMS. The database in the framework contains the translation of task ID to AR instructions. These AR instructions are programmed in the design time part of the AR on a specific user interface. For the demonstrator at TRI and the TNO competence centre the OPS program is used for programming the instructions in a graphical and user-friendly manner. For the BOS demonstrator and the TUM competence centre a custom interface was developed. For both, care should be taken to make modular programs to aid in reusability. Again Horse experts can help with this.

6.6 FlexBe adjustment

FlexBE provides a library of basic steps that allow the definition of a task. In order to communicate with new robots and tools, the definition of additional steps is required. This involves the use of python code to adapt the interfaces of the pre-defined basic steps. After this initial phase, the user can use the graphical interface to define a task. The basic steps, represented as building blocks, needs to be connected together in order to achieve the desired task workflow. The HORSE experts can help in the definition and adjustment of the required steps needed to communicate with the needed hardware.

6.7 Collision detection and avoidance adjustment

The Collision detection and avoidance component requires a URDF model of the robot in order to compute the robot trajectories swept volumes used for the detection of collisions. An important adjustment is the calibration of the 3D cameras used to capture the live environment. Then, the trajectories that the robot has to execute have to be defined using the provided GUI, which allows to save the trajectory waypoints moving the robot manually. The HORSE experts can help for a correct definition of the robot model and providing useful tools for the calibration of the cameras.

6.8 Presence Detection adjustment

After installation of the presence detection environment, the sensor parameters should be configured. For this, all available sensors should be enumerated. For each one, the global position and orientation inside the workcell are required, and an individual launch file with these parameters must be created. A separate node automatically gathers all generated sensor output and fuses the information. This allows for quick modular combinations of different sensor configurations. For the TUM workcell, an

example setup was created. It involved a Saparo pressure sensitive floor, a LIDAR scanner, and a Kinect camera.

6.9 Situation Awareness adjustment

In order to configure the situation awareness module, it is first required to install Eclipse IDE: Eclipse Neon Milestone 6 (4.6.0M6) with the following plugins: *JavaSE-1.8, JRE1.8.0_65, Maven, Message Pack and MessagePack-RPC libraries*[6]*, Protégé*[7] *ontology editor, OWL files contain the ontologies including environment and task-specific information, and SWRL files definining rules for the reasoner.*

Once the development platform is ready, we can start configuring the ontology with geometric data corresponding to the environment (world) settings. The robot actions has to be configured as well depending on the used robot capabilities. In addition to the environment geometric information and to the robot actions, we can also define a set of parameters like _Threshold, _goalPosition, etc.
Those parameters are then used while defining the rules on the ontology features. For instance, depending on the distance between the robot and an obstacle compared to the threshold, different levels of warnings are generated.

The situation awareness component has been demonstrated in CEA competence centre on a transportation scenario with automated and human agents to avoid emergency stops when undesired situations occur.

6.10 Programming by demonstration adjustment

The programming by demonstration functionality is implemented inside the robot controller and triggers the different control modes depending on the current state of the robot. The latter is defined with a state machine which describes the behaviour of the task and the different steps that must be followed by the operator in order to teach the task to the robot. Based on the taught information gathered through programming by demonstration module, the recorded information are processed into action parameters for the execution sequence of the robot automatic mode.

Thus, programming by demonstration component behaviour has to be configured in a state machine which triggers the different control modes of the robot. This component can be mainly used with collaborative robots which are compliant to operators' movements.

The adjustment of the programming-by-demonstration component to an industrial robot is a new challenge for the development, because industrial robots do not have the force sensor at the actuator level (end of robotic arm) where the operator could potentially give the move commands by applying the hand force. Thanks to the flexibility of the programming by demonstration architecture, it is possible to apply this functionality to an industrial robot arm by adding an external force/torque sensor at the end of the robotic industrial arm to make the mechanical interface between the tool and the robot. The signal from the sensor is then deviated by the program-by-demonstration component, which actually takes the control of what the robot controller force control option receives as signal.

This component has been demonstration on OPSA pilot with a KUKA industrial robot and with different robots (collaborative and industrial) in Paris-Saclay Competence Centre.

[6] https://msgpack.org/

[7] https://protege.stanford.edu/

7 HFW in relation to other SW systems

IEC62264 (IEC, 2013) presents a functional hierarchy for manufacturing control (Chen, 2005), as shown in Figure 31. This hierarchy acts as a reference framework to classify the various types of control found in modern factories, ranging from control of complete enterprises to control of individual components of a production machine. At the top of the hierarchy, Level 4 is concerned with the broader business management, including the resource, financial and supply chain management functions. Level 3 is responsible for the planning, directing, coordinating and monitoring of operations in the factory. Level 2 includes the functions used to coordinate and synchronise a grouping of manufacturing resources, to support process execution. Level 1 is the direct control of a single resource (such as a production robot) to execute tasks for the actual handling, processing and sensing of products and material in the factory. Finally, Level 0 is not a control level, but represents the actual humans and industrial hardware (such as robots) that execute the tasks and is responsible for the flow of product and material through the factory.

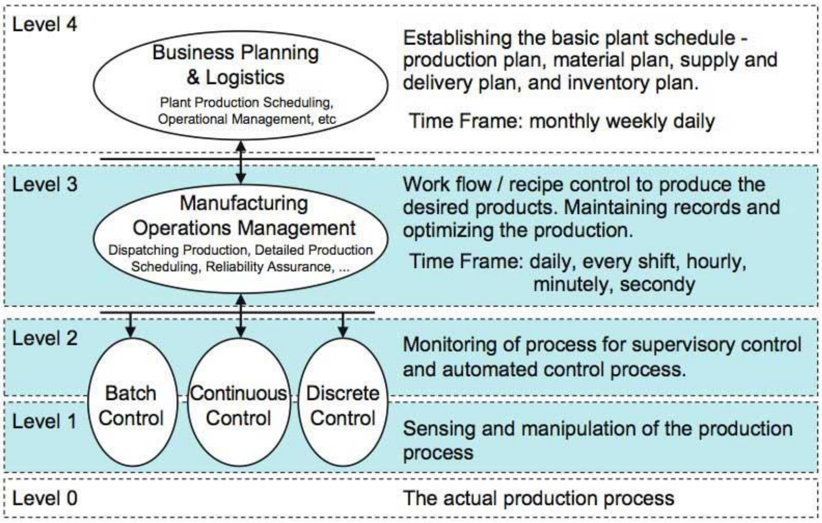


Figure 31: Reference functional hierarchy of manufacturing control (IEC, 2013)

The four levels of control are used to place information system types in context, giving structure to the landscape of computer integrated manufacturing. The information system types typically associated with each of the levels in the functional hierarchy are graphically depicted in Figure 32. For the purposes of the HORSE Project, the most important information system types are Enterprise (or Manufacturing) Resource Planning systems (ERP or MRP) at Level 4 and Manufacturing Execution Systems (MES) at Level 3 of the functional hierarchy.

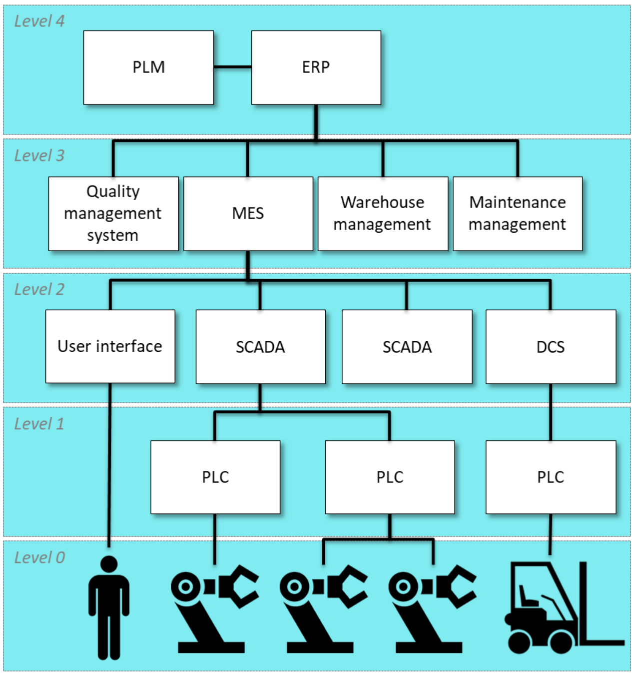


Figure 32: Software and hardware systems positioned on the functional hierarchy

HORSE-D2.2 provides a top-level view of the software aspect of a typical manufacturing enterprise. The diagram is shown in Figure 33 and provides a useful guide regarding the context of the HORSE System. The HORSE System, as viewed in this context, fills a gap between the ERP and MES (corresponding to Levels 4 and 3 of Figure 34, respectively).

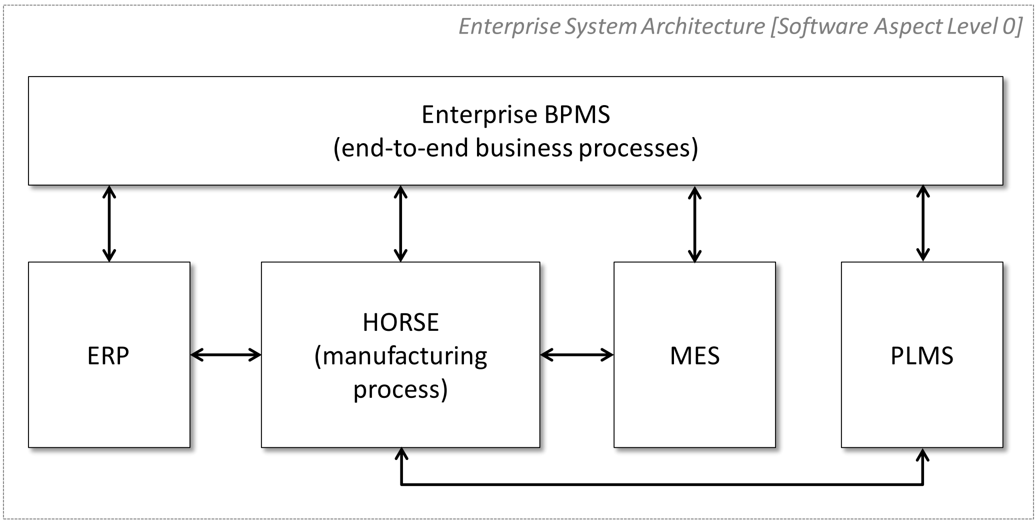


Figure 33: Software aspect at aggregation level 0 (taken from D2.2)

As a very simplified description, the HORSE System will use information from the ERP and PLM systems to enable design of manufacturing processes. When a manufacturing process enters execution state, the HORSE System will use production scheduling and factory floor status information from the MES to determine when to initiate the tasks in the process. A task will be executed by composing a team of agents and providing them with the necessary information to perform a task. The team will then execute the task and provide feedback to the HORSE System.

Instead of attempting to establish integration between the process management components of different systems (i.e., a bottom-up interoperability attempt), we propose a unified, standardised and extensible system that provides process management functionality between and to other information systems (i.e., a top-down choreography approach). The process management functionality is realized in the HORSE Exec Global subsystem of the HORSE System. Individual tasks are supervised by instances of the HORSE Exec Local, which directly connects to the control systems of individual robots and automated vehicles. HORSE Exec Global and HORSE Exec Local are shown in the context of the functional hierarchy in Figure 34.

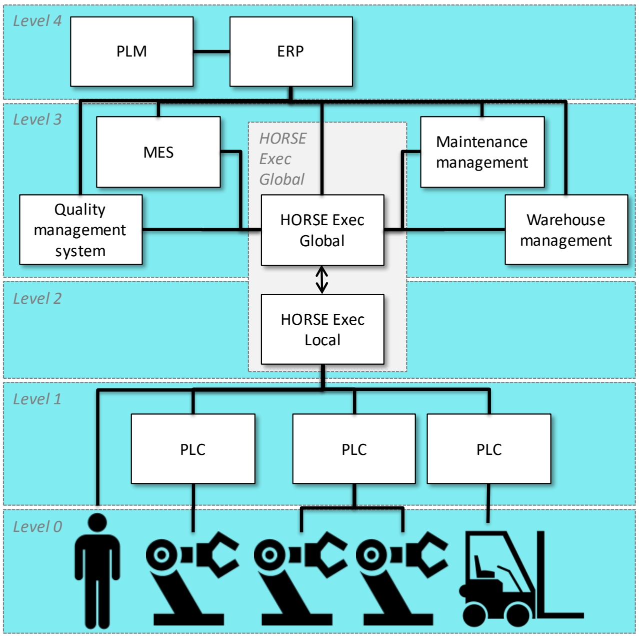


Figure 34: The HORSE System positioned on the functional hierarchy

8 IPR, fees and conditions of use

The chapter presents the conditions of use for the various HORSE components. Some of those components are based on open source tools which have been adopted and redirected to satisfy the HORSE functions. The following table indicates the components and partners who provide and give access to these components. results in terms of system components have been identified and summarised below:

- **C1**: Manufacturing Process Management System
- **C2**: Cyber-physical middleware
- **C3**: HORSE-BOSCH interface
- **C4**: HORSE-ROS bridge
- **C5**: HORSE-KUKA Sunrise interface
- **C6**: Augmented Reality (BOS deployment)
- **C7**: Augmented Reality (TRI deployment)
- **C8**: Learning mechanism for task definition
- **C9**: Multi-modal monitoring
- **C10**: Online replanning
- **C11**: Situation awareness and signal deviation
- **C12**: Hybrid position-force control
- **C13**: Intuitive task programming

Claims for these exploitable results have been made and summarised in Table 2.

The legend for the table is:

- **Rights of use**
  - Licensing (**L**): selling of the software license (integrated platform or stand-alone modules). License schemes are further broken down into:
    - Free license (**FL**): Open-source or other zero-cost license
    - Paid license (**PL**): License with cost
  - Making (**M**): Making the products, manufacturing and selling or directly implementing it through own facilities and skills
  - Use (**U**): internal use of the result

Table 2: IPR Registry table

Partner	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
ED													
CEA											M	M	
FZI				M, FL*				M, U*	M, FL*	M, U*			FL*
PROS		M, PL**	M, PL**		F, M								
TUE	F, M												
SER													
HUA													
OPSA												U	
KUKA					F, M								
BOS	U	U	F, MU, PL**			U							
TUM						F, MU, FL					F, U, FL		
TCS				UL	UL			UL	UL		UL	UL	
TNO							M, PL***						
CET													
TRI	U	U					U						

* FZI contributions to C4, C9 and C13 are under open source licenses and marked as FL. The FZI contributions to components C8 and C10 are proprietary to FZI and currently only available to HORSE partners.

** Regarding the Bosch IoT Gateway Software and Bosch IoT Remote Manager (PROS), these are proprietary products from Bosch Software Innovations. Both products can be either purchased or provided as managed services to potential users of the HORSE platform. The exact pricing structure is confidential, but depends parameters like volume (e.g. number of connected devices). Both products can be made available free of charge for time-limited evaluation period.

*** The AR developments are based on TNO software libraries which are customized for an application. Together with for example an end user or integrator a license agreement can be substantiated. Demonstration of the technologies is available in the TNO CC.