HORSE FRAMEWORK

1 What it is

1.1 Overview

The European manufacturing industry needs to embrace the Industry 4.0 revolution in order to remain globally competitive. Although doing that may be relatively straightforward for big industry, the SMEs face a number of difficulties in the process, mainly the lack of expertise, highly qualified workforce and resources needed to refurbish the whole business in one step. Moreover, there is still some reluctance present, as the benefits of digitizing the business are not always immediately clear and visible. Finally, it is crucial to understand, that digitization encompasses much more than just buying and setting up an industrial robot.

The HORSE project delivers solution to those problems. The software components are designed in a way that makes them easy to customize, use and reprogram, limiting the necessity for highly knowledgeable and qualified personnel. Even more important is the fact, that the framework is easily scalable. Therefore, it is possible to start the digitization with one part of the process (a production line, or even just a single workcell) and then just expand once the framework proves its usefulness and additional resources are acquired.

Small enterprises usually cannot commit to having a single-task robot and their employees also need to be assigned to different tasks depending on the current needs. The flexible and dynamic approach of tasks monitoring and allocation in the HORSE framework is one of its more important features. As a result, the state of all the resources is continuously monitored and both human and robot agents are assigned to tasks according to their individual capabilities and the current project needs. Thus, the overall efficiency within a production shop-floor can be greatly improved.

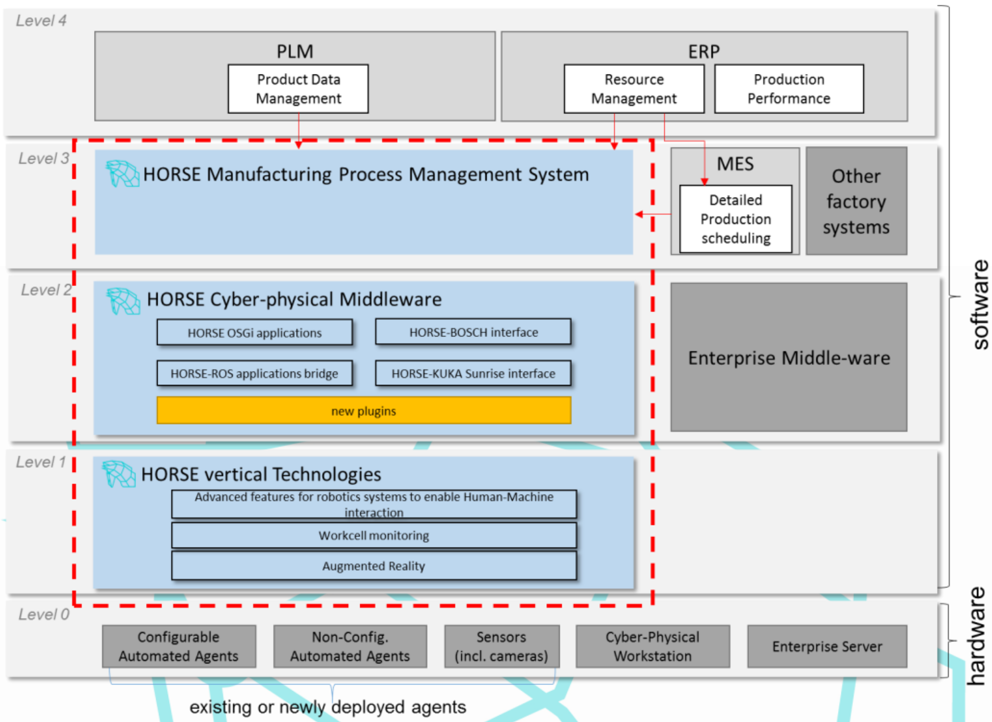


Figure 1: The HORSE Framework

All of those are possible due to the design of the framework, which consists of the basic, universal skeleton, which is used in every HORSE deployment scenario. This skeleton 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 goal of this document is to provide an overview of the whole HORSE framework. This short introduction is followed by presentation of the benefits of using the framework (both business and technical) and positioning of the framework in the wider picture of software for manufacturing control. Afterwards, the success stories – the HORSE pilot use cases – are presented. The next section presents the contents of the framework and finally, the steps to deploy it are presented in the last section.

1.2 Benefits from a Business Perspective

In manufacturing, we currently typically find situations with rather fragmented automated support for manufacturing processes and activities (if available at all). Different systems may be in place for setting up and executing manufacturing batches, allocating tasks to human workers in manufacturing, supervising product flows, and controlling robotic solutions where these are used. Where robots are not used, availability of integrated software solutions may be one of the problems. This leads to sub-optimal situations with little integration between high-level processes (on the factory level) and low-level activities (at the work cell level) on the one hand, and little integration between the activities of human workers and robots on the other hand, or even the absence of robotic solutions where these could bring big benefits. Consequently, we see a lack of flexibility in the assignment of workers and robots, unnecessary

waiting times in production, idling robots, inefficient transfer of information between the dispersed systems, and ineffective, ad-hoc handling of exceptions in the manufacturing process (such as a malfunctioning work cell).

The HORSE framework aims at addressing all these issues in an integrated way. HORSE covers both the global level of manufacturing processes (at the factory or production line level) and the local level of individual manufacturing activities (within specific work cells). It addresses both the set-up of processes and functions at these levels, and the real-time execution of processes and activities.

HORSE is a customizable and modular framework; not only it provides important tools which can be adapted to the specific needs of each SME, but new and legacy hardware and software is able to be integrated and used within the framework.

To achieve this aim, the HORSE System is built on the principles of flexibility and standardisation. The benefits of the system can be summarised in the following points:

- The availability of the complete solutions provided by the HORSE system makes the **accessibility of robotic manufacturing solutions for SMEs much easier**.
- **Adopted international standards and best practices notation (OSGi, ROS, OPC-UA, etc.)** makes the HORSE System applicable to any discrete or batch production facility.
- The **modular** design of the HORSE System makes the system **flexible and adaptable** for different situations and to address a variety of challenges which are faced by manufacturing industry SMEs – not every SME context requires the full HORSE framework; production resources can easily be added, deleted or updated.
- **Seamless integration between HORSE System modules and openness to external technology** (such as robotic platforms and sensors) makes Industry 4.0 technologies accessible to SMEs in manufacturing.
- The explicit manufacturing process management approach (at the global level of the HORSE System) allows **for high levels of flexibility in manufacturing process design**, thereby opening ways for easy re-use of manufacturing activities and underlying manufacturing infrastructure, and evolution towards mass-customization of products.
- The **dynamic allocation of production resources** (such as workers and robots) in manufacturing processes is a strong basis for improved process efficiency, leading to shorter throughput times of manufacturing processes and higher resource utilization.
- Provided **high-level overview of the status of the manufacturing process** at the production line level and manufacturing activities at the work cell level make sure that operational management of manufacturing facilities is always up-to-date of the real-time status of your business.

1.3 Benefits from a technical perspective

The HORSE Framework promotes a modular solution with clearly defined functional elements and interfaces. The key benefits for the developers, integrators and service providers are:

- **Clear interfaces** that allow replacement of modules and integration of new ones:
  - OSGi plugins to the existing OSGi nodes;
  - New ROS components;
  - Modules based on other technologies;

- A **scalable messaging middleware** that
    - is based on a widely accepted communication protocol (WebSocket)
    - is exchanging well-structured JSON formatted messages
    - permits encryption of the payload or the entire communication channel
    - offers reusable components (messaging agents) in Java (OSGi) and Python
    - features bridges to ROS and OSGi
    - supports prioritisation of the messages
- Web services standards simplifies integration to manufacturing technology, making the HORSE System suitable for factories with existing and heterogeneous robotic solutions.

1.4 Wider picture

IEC62264 presents a functional hierarchy for manufacturing control. 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. Finally, Level 0 is not a control level, but represents the actual production shop agents, humans and industrial hardware (such as robots) that execute the tasks.

Figure 2 shows the HORSE System (in blue) in the context of other systems usually found in a factory. The systems are organised according to the functional hierarchy of IEC62264. These systems may be information systems, hardware or even cyber-physical systems. Figure 2 also attempts to relate the external systems to the landscape of computer integrated manufacturing.

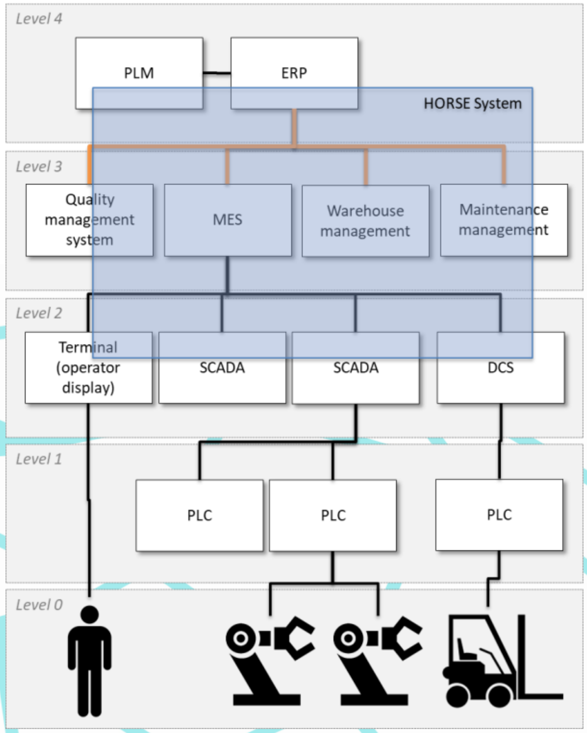


Figure 2: Typical systems found in a manufacturing system, arranged according to the functional hierarchy of IEC62264. HORSE System shown as an overlay, to convey its positioning in relation to other systems.

The HORSE System may take input from the operations manager, production planner or other information systems. Production is initiated by activating a manufacturing process (on level 3). Once the process reaches the first human or robot task, a task instruction is sent to the control system of a robot or the user interface for a human. This control system and user interface still forms part of the HORSE System, and is situated on level 3 of the functional hierarchy.

2 How is it used

2.1 TRI use case

Thomas Regout International (TRI) is one of the partners of the project that drives the HORSE requirements and is used to demonstrate flexible assembly. At the moment TRI finds itself in a worldwide leading position in producing and designing customized telescopic slides for several industrial equipment applications. The market of the company is characterized by high quality demands (zero defects), just in time delivery, high service levels, and, customized small series with short time-to-market. The production processes of shaping, punching and assembly are semi-automated.

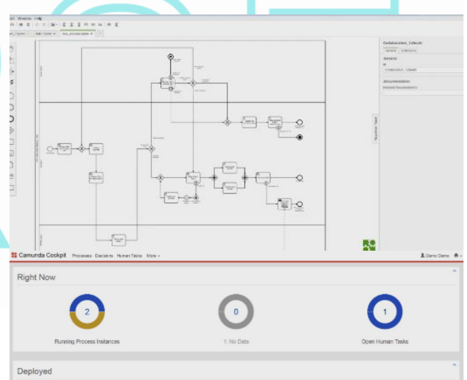


Figure 3: With HORSE, the use of resources is now optimized

The current production process consists of the following 3 steps:

1. Cold forming of steel, stamping and welding (P1).
2. Surface treatment of the steel profiles (P2).
3. Final assembly of slides (P3).

The challenge for TRI pilot case is to connect these three processes by a usable and easy to use ***Human-Robot interface*** that allows ***adaptability*** of the production processes and promotes ***interoperability*** between different systems and semi-automated procedures.

The HORSE framework was used in two separate aspects here. The first one is a robotic station performing a tedious and tiresome task of putting parts on the hangers for the chemical processing and then removing them from the hangers. This allowed to eliminate the non-ergonomic aspect of the process. The second one is providing assembly instructions in the task that still requires manual manipulation. This applications significantly reduced dependability on highly-skilled workers and improved the quality of assembled components.

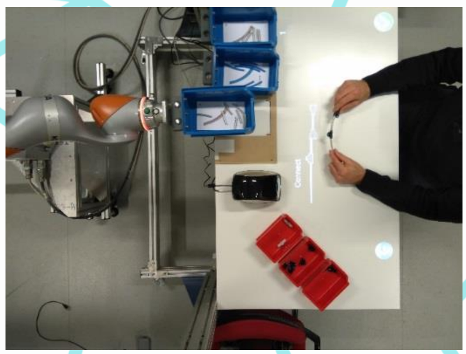


Figure 4: With HORSE, workers with no experience are now able to assemble production tools, faster and with less errors than experienced workers used to.

2.2 BOS use case

Robert Bosch España Fábrica de Castellet S.A.U. (BOS) is located in Catalunya and is part of Robert Bosch GmbH. BOS is one of the pilot partners and is used to demonstrate **human–robot co-manipulation**. BOS pilot case is implemented in front Wiper Systems Assembling (WSA) lines. Currently, the last step of the manufacturing process is manual and consists of unloading the WSA from the conveyor belt, performing the visual quality check of the part, packaging it into a box layer by layer and laying layer separators when needed. Such a process is uncomfortable physically (due to non-ergonomic movement) and mentally (as the faults are relatively rare) to the human operators.

Figure 5: With HORSE, workers can now be focused more on added-value activities, as the packaging and final check of the products is automated

The HORSE intervention focused on developing an automated visual inspection and packaging system. The robot arm picks the WSA from the conveyor belt, presents it to the automatic vision system and, if the part is correct, puts it in the box. In case a defect is detected the robot enters the collaborative mode and the assistance of the human operator is requested. Once the operator arrives, the part is held in front of him or her and the suspected faults are highlighted directly on the part using the projector-based Augmented Reality system. The final decision whether to accept the part as is, reject it or rework it on spot is made by the operator.

The HORSE Manufacturing Process Management System is used to control the flow of the process and the Hybrid Task Supervisor is used to control and synchronize the tasks of the robot, the AR system and the human operator on the local workcell level.

2.3 OPSA use case

The third of the pilot partner industries is Odlewnie Polskie Joint Stock Company (OPSA), a service foundry that is used at the HORSE project in order to demonstrate **human robot co-working with programming by demonstration.** The company produces annually about 16 thousand tons of cast iron castings within weight of 2 - 100 kg in different configuration and use (approx. 990 different types of castings). Most of the operations in manufacturing process are mechanized and automated. These are automatic moulding processing station, automatic moulding lines (horizontal for bigger parts and vertical for smaller), pouring and shake out operations, automatic core shooters.

The biggest challenge for the foundries, such as OPSA, is ***automation*** and robotization of fettling operations, due to the amount of the ***processes and product variations***. As a result, ***adaptability*** and ***flexibility*** are among the most important aspects of the system, so that auto-configuration approaches are applied and robotics can adapt easily to various castings configuration, as well as the rapid change of the demand in order to finally build an intelligent system.

The biggest problem which most of the foundries face in this area is separation of castings; meaning cutting and knocking off of castings from gating and riser systems, called ”grapes”, as these operations are done only manually. Currently, the operator takes out the full grape from the pallet using manual labour or manipulator, puts grapes on the floor, uses a heavy grinder or 15kg hammer to separate the castings from the gating system, puts the castings on a pallet and the gating system to scrap box.

Figure 6: Workers had to manually cut the castings for many product variations. With HORSE, the task is now automated, and workers can be more focused on the final product quality.

The HORSE intervention focused on providing an industrial grade robot equipped with a specialized tool that is capable of separating the castings. The number of part variants makes it unfeasible to program the robot using the traditional means, therefore a programming by demonstration approach was selected. The HORSE Manufacturing Process Management System is used to trigger the local tasks of programming and performing the cutting operation.

3 What is there

3.1 Introduction

The HORSE framework consists of several software components that can be grouped into three main groups: the generic components, the interfaces, and the case-specific ones. 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 workcell, 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.

The core, obligatory elements of each HORSE deployment are the MPMS and the middleware. The first one is necessary to define, execute and monitor the process (further described in Section 4.1). The second one is the middleware that provides communication capabilities for the heterogeneous components of the framework (details provided in Section 4.3). Depending on the realized scenario the HTS can be also used to trigger and synchronize tasks on the level of individual workcells. Section 4.2 provides information on how to use it.

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.

3.2 Generic components

3.2.1 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 7 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.

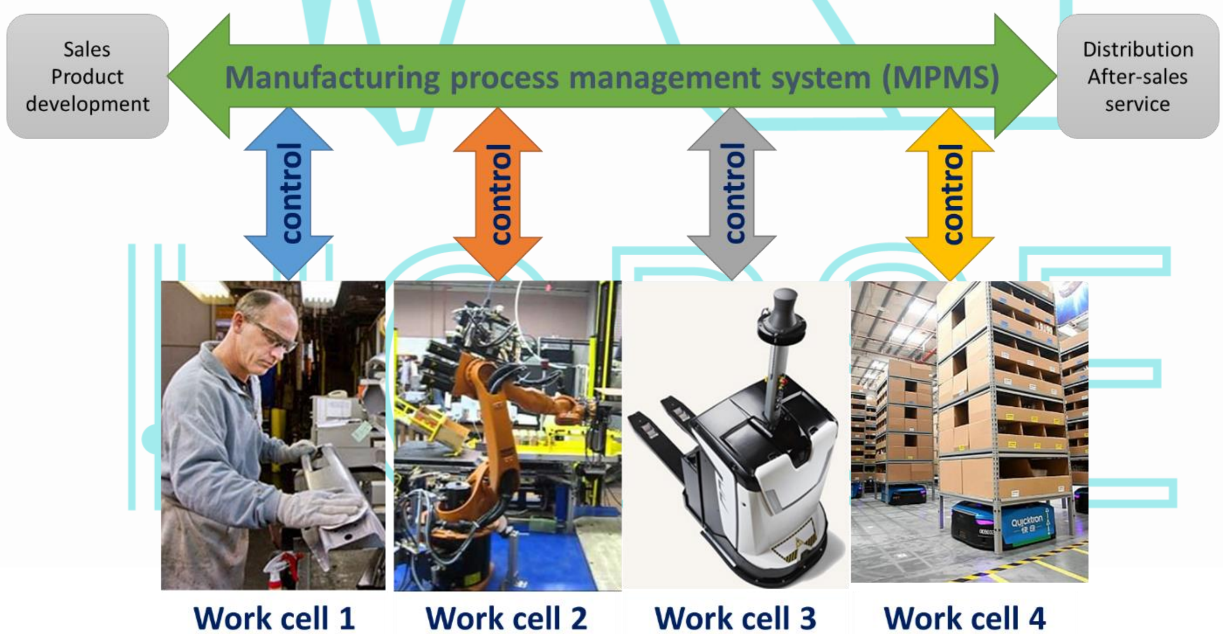


Figure 7: Conceptual illustration of the MPMS in relation to the work cells

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.

3.2.2 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.

3.3 Interfaces

3.3.1 Middleware

The HORSE 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.

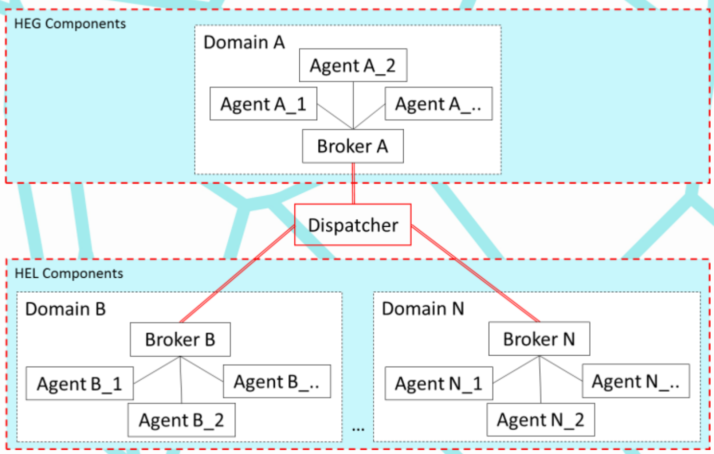


Figure 8 HORSE Messaging Middleware components

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.

3.3.2 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 HORSE middleware.

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.

3.3.3 Interface to industrial equipment: HORSE-BOSCH adapter

The HORSE-BOSCH adapter (Figure 9) was developed as a bridge between the HORSE Message Broker and the corresponding 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.

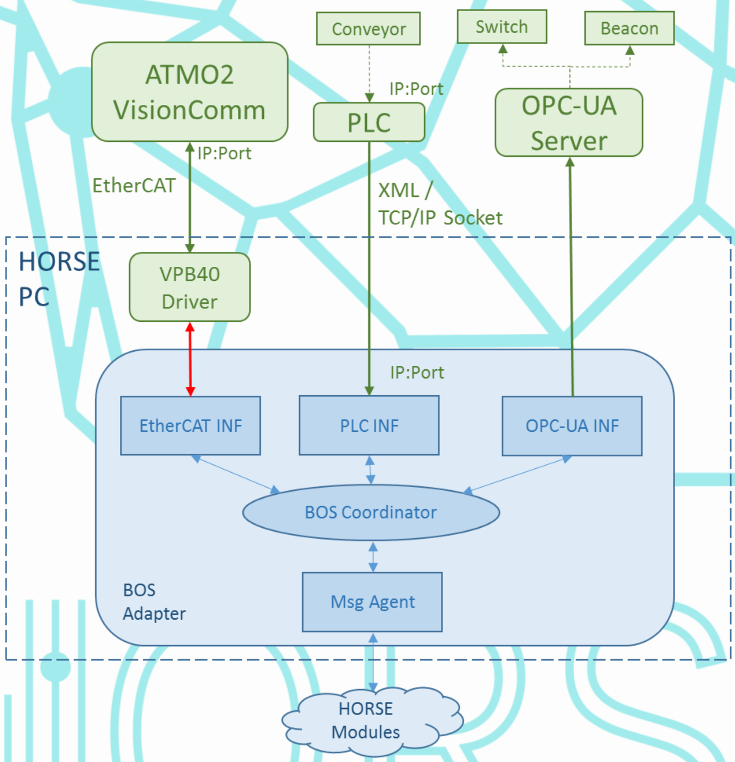


Figure 9 The Bosch adapter and Bosch machines

3.4 Specific components

3.4.1 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 workcell 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 workcell 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 10 The AR for assembly

3.4.2 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. The component is responsible for projecting additional information (e.g. highlighting the inspection points) directly on the part held by the robot or placed in a known position. The functionalities of the component are provided as a set of ROS actions triggered via the HORSE-ROS bridge (Section 0). In case the robot is used to manipulate the part the robot control and AR are synchronized by the Hybrid Task Supervisor (Section 3.2.2).

Figure 11: 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 workcell with an overhead projector, a camera and, optionally, a robot arm. This needs to be followed by an optical and spatial calibration of the elements of the workcell and setting up the overlays to be project and, again optionally, robot arm positions.

3.4.3 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.

3.4.4 Situation Awareness

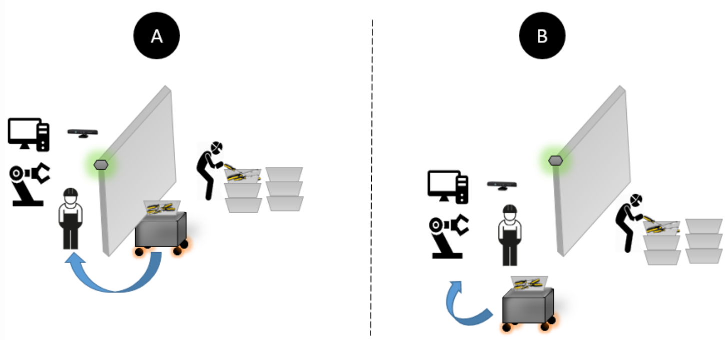
Smart factories could significantly increase production time and improve operators' working conditions in the manufacturing industry. They involve the collaboration without fences of robots and humans, whose safety needs to be ensured. Specifically, safety stops must be avoided because they may considerably 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' behaviors 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.

3.4.4.1 Example of application

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 below (on the left side) there is a situation where a collision may occur between a human agent leaving a workcell and a mobile base entering into the same workcell. 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 behavior to avoid a collision (scenario B on the right side).



3.4.4.2 How does it work?

The situation awareness module (shown in the figure below) 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.

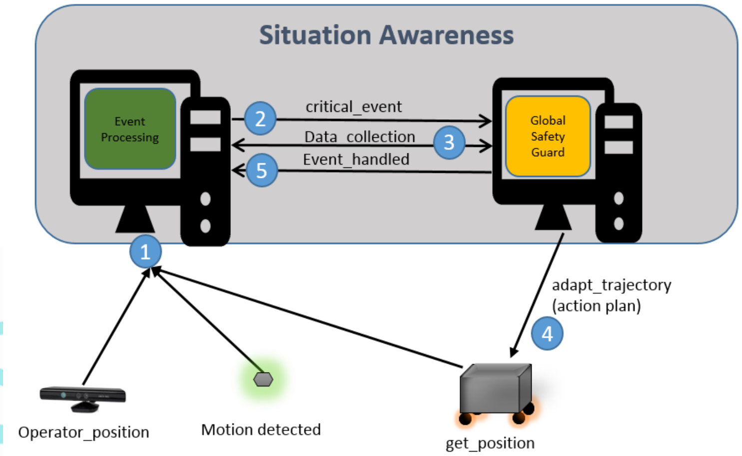


Figure 12- Situation awareness mechanism.
1) Data are gathered from the devices and the agents participating in the workcell; 2) A critical event is raised whether an anomaly may occur; 3) Relevant information from the environment is collected by the Global Safety Guard where a reasoning about the environment is done; 4) An action plan is generated to the concerned agents.

4 How to get started

4.1 Defining the scenario

The HORSE Project uses a dual-perspective approach to describe demonstration scenarios. Due to the inherent physicality of manufacturing, factories are traditionally considered from a layout perspective. The factory floor layout shows the location of work units and how work is directed from one work unit to the next. The HORSE Project views the physical position of work units and the flow of work as separate aspects of the manufacturing system.

The physical aspect is reduced to a system breakdown structure of the factory. The goal of such a depiction is to catalogue the entities present in the factory and how those entities are organised. The physical aspect is documented according to the physical hierarchy of IEC62264:2013. Figure 13 shows the physical hierarchy model recommended by IEC62264:2013.

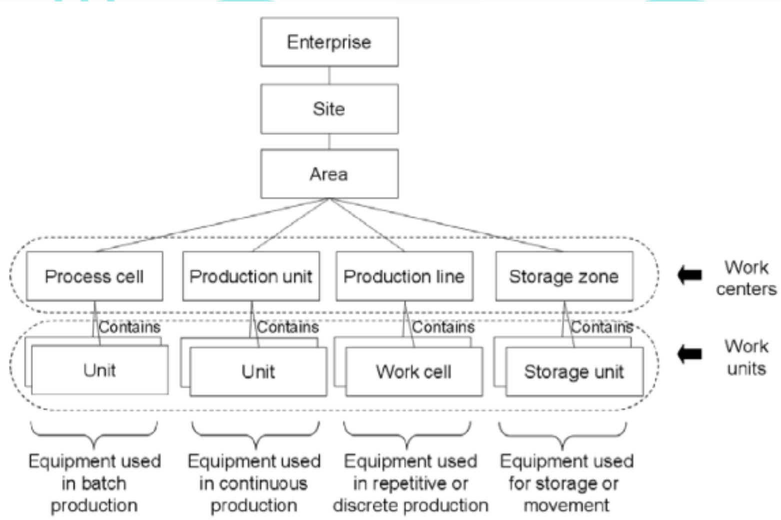


Figure 13: Physical hierarchy of a manufacturing system, as recommended by IEC62264:2013

By adopting the structure of the physical hierarchy, open call applicants can describe the composition of their factory. It is not necessary to detail every work centre and work unit of the factory, but rather to show the position of the demonstration scenario in its physical context. To serve as an example, Figure 14 shows the physical hierarchy for Thomas Regout International (TRI), one of the three pilot cases of the HORSE Project. The demonstration scenario in this factory is part of the production line labelled "PL2.1: Loading."

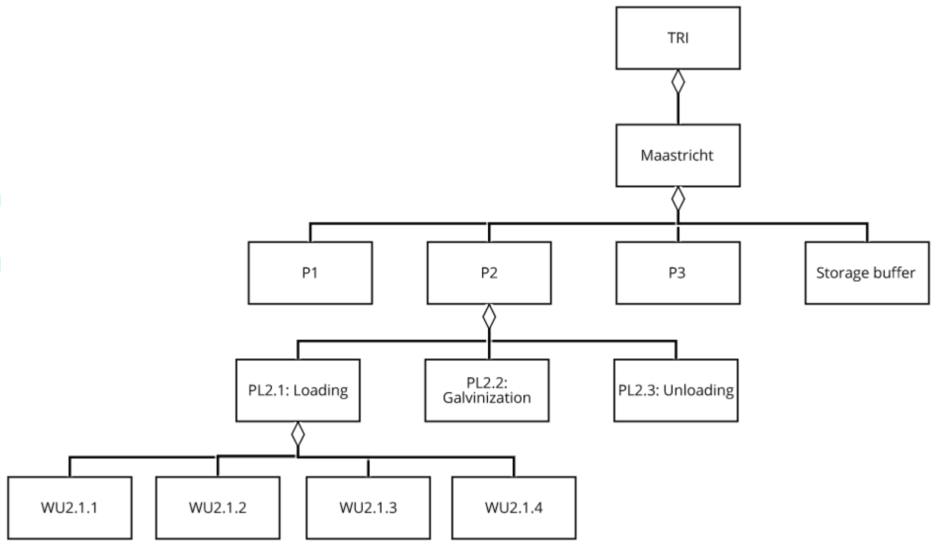


Figure 14: TRI interpretation of physical hierarchy, with demonstration scenario as part of P2

A clear understanding of the physical composition of the factory makes it easier to model the manufacturing process. The end-to-end manufacturing process model should at least cover the activities performed at all production areas. The goal is to depict the dependencies between those activities. For example, the activities performed at P1 must always happen before P2, because P1 produces the material that must be treated in P2. This may seem obvious, but it is highly informative to a person not familiar with the factory.

The HORSE Project makes use of internal standards to model processes. Specifically, the Business Process Model & Notation 2.0 (BPMN2.0) is used for the purposes of the open calls. The following four model elements are used to model processes:

Table 1: Symbols used for the four elements used in process models

Element	**Symbol**	**Description**
Activities		Actions performed by agents
Events		Things that may happen during the execution of a process

Exclusive choice gateways		A decision point that activates one of several process flows. Also used to merge alternative process flows into a single flow.
Parallel gateway		Splits the process flow into two or more active flows. Also used to merge two or more active process flows.
Connectors		Links between activities, events and gateways that enable the process flow

The physical hierarchy (see Figure 13) can be created with any software package with simple drawing tools, such as MS Powerpoint. For the process models though, we recommend that the applicant downloads the Camunda BPM modeller, which can be found here. The following two process models should be created by the applicant:

1. A contextual model of the end-to-end manufacturing process, of which the demonstration scenario is part of, and
2. A descriptive model of the process involved in the demonstration itself.

We provide illustrative process models to give some insight into such models. Figure 15 shows the contextual process model of TRI. The same three production areas are shown in three lanes. The storage area is not shown, because was deemed not important for the process view.

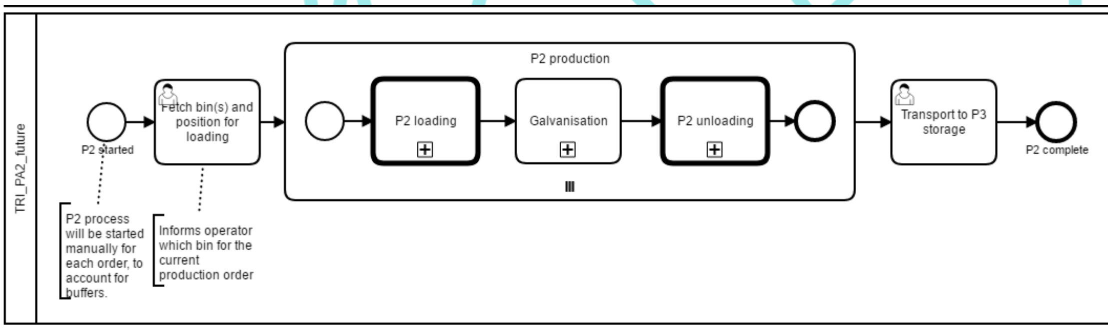


Figure 15: End-to-end process model of TRI

The demonstration scenario at the TRI factory is situated in production area P2. Figure 16 shows the process model of the demonstration scenario at TRI. For the applicant, it is not necessary to be concerned with the detailed symbols shown on the figure, but to understand that this process modelling will be required.

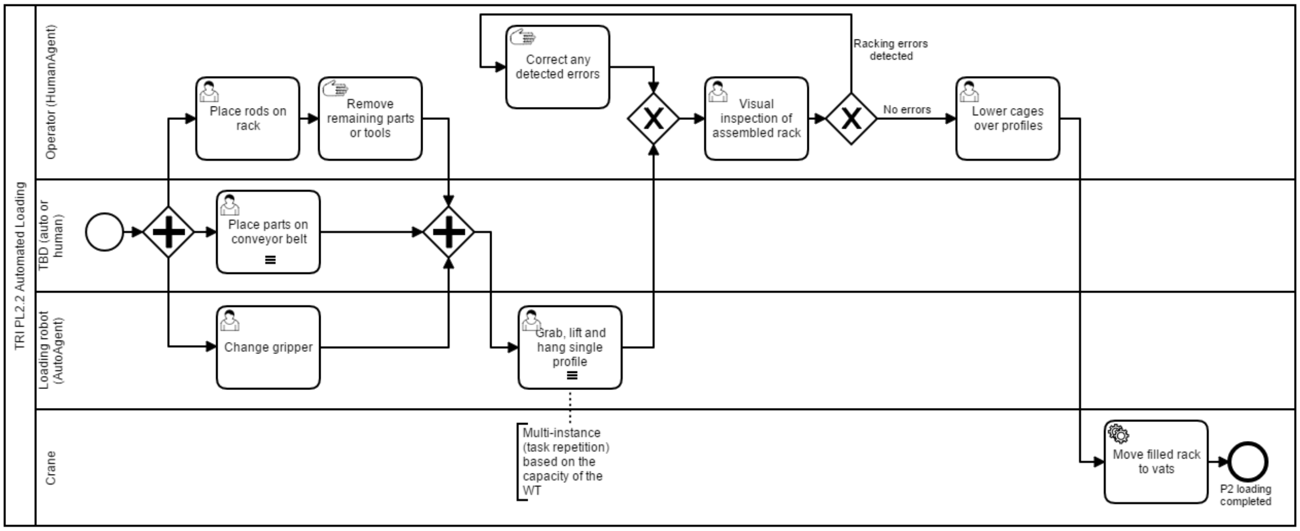


Figure 16: Process model of the demonstration scenario at TRI

Any of the models created by the applicant can be supplemented by a textual description to convey additional information about the problem or special considerations. The final step is to describe the expected change. By referring to tasks from the last process model (see Figure 16), the applicant can describe how the process and its tasks should change to demonstrate advanced manufacturing technology.

4.2 Defining the tasks

Modern and flexible industrial production requires a quick and easy operation of robot systems. When several robots or several robot components need to operate cooperatively, the effort for programming the system becomes considerable, given the difficulty of efficiently synchronizing all the working components. Therefore, there is the need to program the robot tasks in an intuitive way providing the possibility to easily change the work flow using user-friendly interfaces.

In order to define tasks in an easy way, HORSE extends the FlexBE framework, which allows the definition and testing of custom tasks through a user-friendly graphical interface.

The user can connect together building blocks that represent the steps, which together define a complex task. The steps represent for example the movement of a robotic arm or the activation of a specific tool. They can be parameterized to permit an easy specification of the desired values in the task editor and it is possible to fully customize step definitions to allow for the integration of additional hardware and software interfaces after the initial deployment of HORSE.

Once the basic steps to control a robot or a tool are defined, the user can easily connect them to fulfill the desired task workflow using the graphical interface. This allows the easy definition or modification of a task without the need of a software expert.

Through this module is also possible to define the requirements related to the execution of a task, for example the need of an autonomous agent equipped with a gripper or capable of handling heavy payload. Once these requirements are defined, it is possible to retrieve this information to allow the HORSE framework to select the suitable agents for the execution.

In conclusion, using FlexBE for the task definition allows the user to define the task execution flow, all the required parameters and the necessary components that are needed for the complete task definition, avoiding the need of specialized operators.

4.3 Using and developing components

The HORSE framework features a number of components realising specific aspects of the production process or providing the safety of the humans and machines. It communicates with the other systems as demonstrated in the figure below.

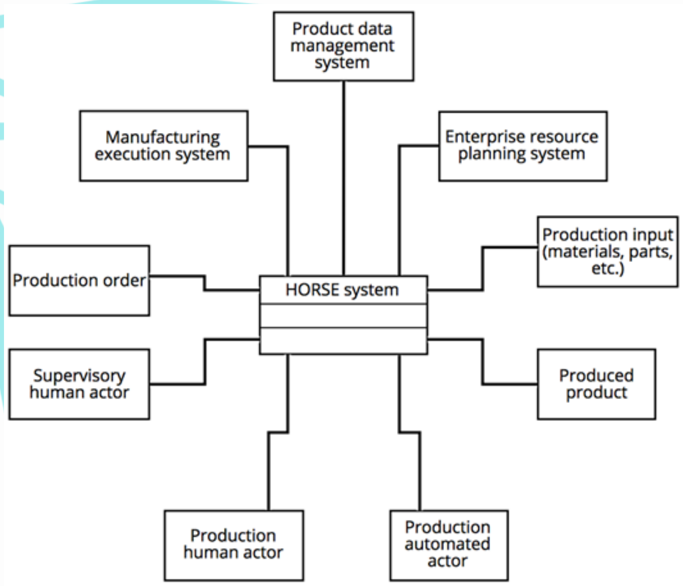


Figure 17 Context of the HORSE system

The customisation of the system according to the needs of the specific deployment requires understanding of the functions of the available components and interfaces, including the taxonomy of the alerts. These are available in details in separate documents. Few non-binding guidelines:

- Existing software systems could be integrated by:
  - direct utilisation of the middleware and HEG (HORSE Exec Global) interfaces and databases, or
  - by introducing an adapter (see HORSE-BOSCH interface) that exposes the system functions in the messaging middleware.
- Hardware equipment and other devices could be integrated through the appropriate device or agent interfaces. Some of the existing implementations (e.g. ROS bridge, industrial and device protocols) could be used or a completely new adapter.
- For an integration to the messaging middleware the new component needs to:
  - implement a local WebSocket client,
  - know the topics of the expected messages and the messages the other party is expecting;
  - know the intended recipients or send broadcasted messages;
  - know the senders of the incoming messages (or filter all incoming messages on filter and other properties)

- know the access parameters of the local broker (IP and port);
- know the semantic of the payload
- It is expected that the existing DB structures should be customised to support the specific product and agent definitions.

4.4 Testing the framework – Competence Centers

Competence Centres in HORSE are physical locations providing expertise, equipment, service, advices and support in robotics technologies and applications in manufacturing. Competence Centres will offer expert advising support on deployment and fast assessment of robotics solutions in manufacturing.

During the HORSE open call preparation and execution, Competence Centres will support the application experiments with knowledge and equipment, as they hold robotics equipment and supplies used in production lines. They will be available to support the adoption and customisation of HORSE framework for the Application Experiments selected by the Open Call. Each of the experiments will be able to use the facilities of CCs for up to two weeks during the implementation of the experiment.

Four Competence Centres are supported within the project. Three existing Competence Centres in France (Paris-Saclay, CEA), Germany (Munich, TUM), the Netherlands (Delft, TNO) will be further equipped and expanded, thus exploiting existing facilities, equipment, experience and network. The fourth one will be established by HORSE in Slovenia (Celje, TCS), and will be the seed for the future and a model for the deployment of Competence Centres in Europe. Please refer to Deliverables D7.1 for more information on the Competence Centers.

Figure 18 Location of the Competence Centers

CEA – CC Rue Noetzlin, 91190 Gif-sur-Yvette, France www.digiteo.fr	TNO – CC Bakemastraat 97K, 2628VK Delft, Netherlands
TUM – CC Schleißheimer Str. 90A, 85748 Garching bei München, Germany	ROBOFLEX SLOVENIA Kidričeva ulica 25, 3000 Celje, Slovenia

The ROBOFLEX CC is being set up during the HORSE project, therefore the bulk of the assistance to the experiments will be provided by the other three CCs. The following equipment can be accessed in their facilities:

CEA CC:

- Two highly transparent robots SYBOT usable as collaborative robots or as a telerobotic slave robots
- One HAPTION haptic device usable as a master arm for teleoperation
- One A6.15 RB3D 7 DOFs collaborative robot
- One COBOMANIP from SARAZIN technology – collaborative robot for assistance to load handling
- STAUBLI RX90L and TX90XLTX90 6-axis industrial robot for tele-operation or hybrid command (force and position control)
- One KUKA arm IIWA
- One YUMI from ABB
- One UR10 from Universal Robotics
- One Artemis AGV (automated driverless vehicles) from BA system for intra logistics
- VR platform for virtual prototyping and training for industrial applications
- One 3D TV equipped with a real time simulation environment for physical interactions.
- One high performance 7 degrees of freedom upper limb exoskeleton ABLE from HAPTION
- One KINOVA 6 DOF JACO robot equipped with a 3 fingers gripper and mounted on a ROBOSOFT ROBULAB 10 mobile platform for feasibility tries

TUM CC:

- Collaborative robot arm : KUKA LRB iiwa with a human-robot interaction safe R800 gripper
- SAPARO tactile floor
- Hokuyo UTM-30LX laser scanner
- RGB-D sensors (Kinect, RealSense)
- Logistic robot (FESTO Robotino)
- Mock-up of catory manufacturing cell with overhead, projector-based AR setup
- HORSE framework
- Access to engineering software (Matlab, ANSYS etc.)

TNO CC:

- 2 Collaborative robot arms: KUKA LRB iiwa with a human-robot interaction safe R800 gripper
- LRMate200iB
- HORSE framework
- Access to engineering software (Matlab, Python, CAD, etc.)
- Industrial (3D) cameras and Kinect camera's
- Computers for supervision and control
- Beamers
- OPS Light Guide systems
- AR Software
- Physical assembly work station
- Human movement registration systems: XSENS MVN and KINECT

- Muscle activity measurement system: EMG
- Eye tracking & emotional face reader
- Cognitive load registration system (e.g. VIG track)
- Force plate or alternatives to record forces for biomechanical analysis
- Heart rate registration devices