Mobile dual arm robotic workers with embedded cognition for hybrid and dynamically reconfigurable manufacturing systems

Grant Agreement No	**:**	**723616**
Project Acronym	**:**	**THOMAS**
Project Start Date	**:**	**1st October, 2016**
Consortium	**:**	**UNIVERSITY OF PATRAS (LMS)** **PEUGEOT CITROEN AUTOMOBILES S.A. (PSA)** **SICK AG (SICK)** **FUNDATION TECHNALIA RESEARCH & INNOVATION (TECNALIA)** **ROBOCEPTION GMBH (ROBOCEPTION)** **DGH ROBOTICA, AUTOMATIZACION Y MANTENIMIENTO INDUSTRIAL SA (DGH)** **AERNNOVA AEROSPACE S.A.U. (AERNNOVA)** **INTRASOFT INTERNATIONAL SA (INTRASOFT)**

Title	**:**	THOMAS Station Controller – Final Version
Reference	**:**	D5.5
Availability	**:**	Public
Date	**:**	30/09/2019
Author/s	**:**	INTRASOFT, LMS, TECNALIA, PSA
Circulation	**:**	EU, consortium

Summary:

This document provides the final version of THOMAS Station Controller.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. EXECUTIVE SUMMARY

THOMAS core objective is to increase re-configurability of production systems by introducing mobile dual arm workers able to collaborate with human operators in dynamic changing environments. To this direction, cognition and decision making is a core aspect [1] for meeting the projects' milestones. In THOMAS project cognition and decision making are included in different layers of the system. In a high level the components that are responsible for integration, cognition and decision making are the Task Planner and Station Controller. Task Planner is responsible for the assignment of Tasks to Resources and Station Controller is responsible to orchestrate the execution of this task. However, the two components interact frequently since the assignment of tasks to resources is dynamic and can be changed during the manufacturing process in order to address emerging needs. In another level the other Thomas modules and sensors are responsible for functionalities that be classified into three different layers: a) Data Storage and representation, b) Data processing and b) Cognition

This deliverable presents the final prototype of the THOMAS Station Controller and the Task Planner. The final prototype is based on the initial prototypes presented in D5.2 "Dynamic work reorganization module – Initial prototype" and D5.3 "THOMAS Service Oriented Network of Resources – Initial Prototype". The design of the final prototype as well as the initial prototypes is presented in deliverable D5.1 "Methods for dynamic work balancing of human robot collaborative environments - Design". The API of the final prototype of the Station controller and the Network of services that enable the THOMAS Process reconfiguration through Adaptive skills is documented on D5.4 "THOMAS service-based integration and communication network – Final Version".

This document focuses in the description of the final version of the Station Controller and the HRC Task Planner.

The final prototype for the Station Controller for task can execute and coordinate tasks using the THOMAS Network of Services so that the robot's cognition systems can adjust them for real world execution. The final prototype is an updated, improved version of the initial prototype.

HRC Dynamic Task Planner is the core decision making software that facilitates the cognitive aspects of the line level work re-organization. The Task Planner can be triggered to re-plan and re-allocate the required tasks to the existing resources. The Task Planner triggering may be outcome either of human input / request or any other unexpected event such as resource breakdown. THOMAS Digital world model latest version is presented in this document too.

The D5.5 THOMAS Station Controller – Final Version" will be used, connected and integrated with the developments of the other THOMAS work-packages in the demonstrators of THOMAS and be part of the THOMAS Open Production Station that will be the output of WP6. Furthermore, within WP7 "THOMAS Demonstrators and Assessment" the capabilities of the final prototype of the THOMAS Station Controller together with the other THOMAS developments will be used to implement the pilot cases. The pilot cases of the automotive and aeronautics sectors will validate the effectiveness of each result based on the metrics and acceptance criteria that are defined in T1.1.

2. INTRODUCTION

THOMAS core objective is to increase re-configurability of production systems by introducing mobile dual arm workers able to collaborate with human operators in dynamic changing environments. To this direction, cognition and decision making is a core aspect [1] for meeting the projects' milestones. In THOMAS project cognition and decision making are included in different layers of the system. In a high level the components that are responsible for integration, cognition and decision making are the Task Planner and Station Controller. Task Planner is responsible for the assignment of Tasks to Resources and Station Controller is responsible to orchestrate the execution of this task. However, the two components interact frequently since the assignment of tasks to resources is dynamic and can be changed during the manufacturing process in order to address emerging needs. In another level the other Thomas modules and sensors are responsible for functionalities that be classified into three different layers: a) Data Storage and representation, b) Data processing and b) Cognition

The latest version of THOMAS Digital world model as previously introduced in deliverable D5.3 is presented in section 3 of this document.

This deliverable presents the final prototype of the THOMAS Station Controller and the Task Planner. The final prototype is based on the initial prototypes presented in D5.2 "Dynamic work reorganization module – Initial prototype" and D5.3 "THOMAS Service Oriented Network of Resources – Initial Prototype". The design of the final prototype as well as the initial prototypes is presented in deliverable D5.1 "Methods for dynamic work balancing of human robot collaborative environments - Design" . The API of the final prototype of the Station controller and the Network of services that enable the THOMAS Process reconfiguration through Adaptive skills is documented on D5.4 "THOMAS service-based integration and communication network – Final Version".

The Task Planner implements THOMAS Line Level Work re-organization functionality and can be shown together with the Station Controller in the context of the context of THOMAS overall application component diagram in Figure 1.

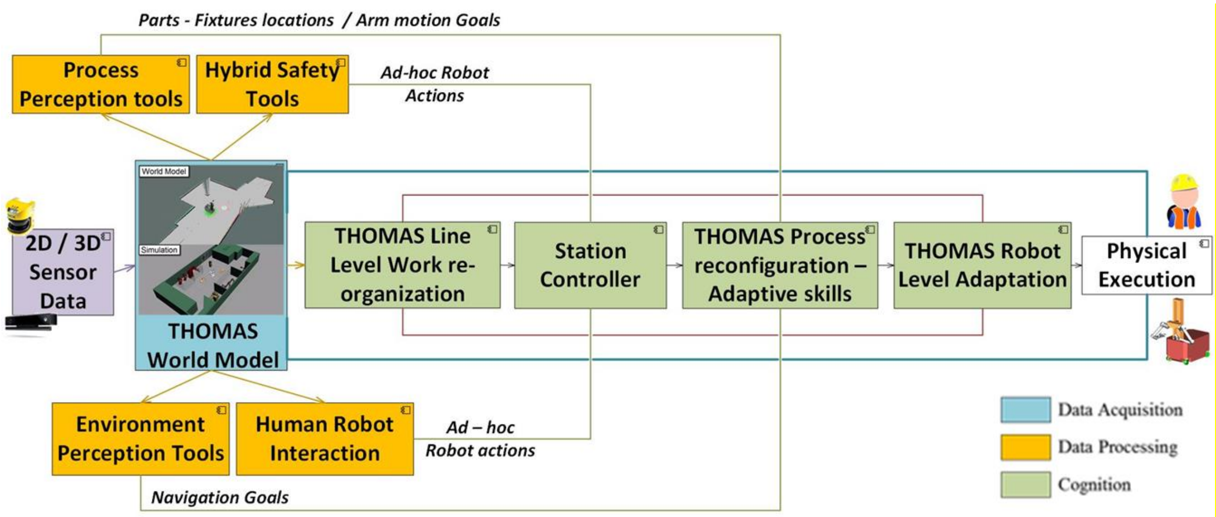


Figure 1: THOMAS Overall Application Component Diagram

- **Station Controller:**

The final prototype for the Station Controller for task can execute and coordinate tasks using the THOMAS Network of Services so that the robot's cognition systems can adjust them for real world execution. The final prototype is an updated, improved version of the initial prototype. The

final prototype is an updated, improved version of the initial prototype. In addition, it offers a set of features:

- **Task Take Over:** The station controller is capable to coordinate the taking over of tasks from one resource to another dynamically. For instance, a human operator could overtake a task normally assigned to the robot, in this case the Station Controller would proceed with the execution of the schedule normally.
- **Closed Loop Task Planning and Execution:** The station controller integration with the HRC Dynamic Task Planner allows the partial or complete rescheduling of tasks of a schedule.
- **Unexpected events handling strategies:** The station Controller can implement strategies that aim to address unexpected events, such as execution faults or safety events.

- **HRC Dynamic Task Planner**:

HRC Dynamic Task Planner is the core decision making software that facilitates the cognitive aspects of the line level work re-organization. The Task Planner can be triggered to re-plan and re-allocate the required tasks to the existing resources. The Task Planner triggering may be outcome either of human input / request or any other unexpected event such as resource breakdown.
The following key steps constitute the assignment process executed by the HRC Dynamic Task Planner:

- **Definition of possible assignments** (alternatives) of tasks/operations to resources at a decision point according to.
- **Selection of criteria** for the assignment of tasks/operations to resources.
- **Evaluation of generated alternatives** according to the selected criteria.
- **Selection of the highest scoring alternative**.

The aim of this document is to present the initial prototype developed under WP5 concerning the following functionalities:

- **THOMAS Resource Shared Perception** through THOMAS World Model including the develop components for resource/sensor/layout managing and 3D scene reconstruction and representation.
- **THOMAS Line work re-organization** through the: a) HRC Dynamic Task Planner for the decision-making part and b) Execution Coordinator for the work queueing and regulation part.
- **Resource level adaptation** through dynamic motion and path planners based the real time sensor data and resource status exposed through the THOMAS world model.

To comply with the prototype requirements the THOMAS overall workflow has been broken down into three high level layers as presented in Figure 2:s

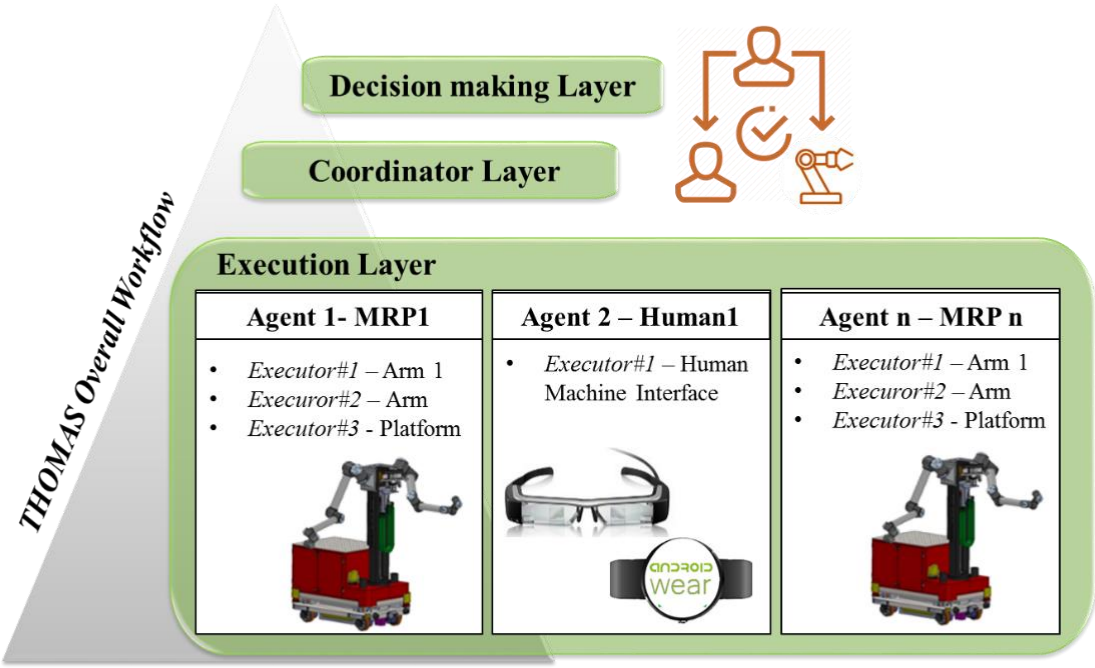


Figure 2: THOMAS Overall Workflow layers

- **Decision making layer:** This layer hosts all the modules that are responsible for reasoning over the available information in order to derive effective plans for the scenario execution/
- **Coordinator Layer:** This layer hosts all the components that contribute in the efficient coordination of the high-level execution, dispatching the assigned tasks and ensuring the regulation of precedence relations and synchronization when needed.
- **Execution layer.** This layer hosts the components that contribute in the low-level control of the resources – both human and mobile robot resources.

The developed prototype has been based on this workflow concept. The implementation details for each of the discussed modules are presented in the main body of the deliverable.

3. World Model

3.1. World Model overview

THOMAS World Model is the infrastructure for enabling the shopfloor data acquisition as well as combine them in a common representation and making them available to the overall THOMAS system. A continuous feedback from the actual shoopfloor (using resource and sensor data) enables the dynamic update of THOMAS world model. This world model has two main functionalities:

- **Virtual representation of the shopfloor** using multiple sensor data combination and CAD models. The digital shopfloor will be rendered in the 3D environment exploiting the related capabilities provided by Robot Operating System (ROS) framework.
- **Storing Semantic Information** in THOMAS World repository. A unified data model will be implemented in order to semantically represent the geometrical as well as the workload state. This data model should be generic enough in order to be able to address multiple cases as well as to be consumed by multiple components inside THOMAS system.

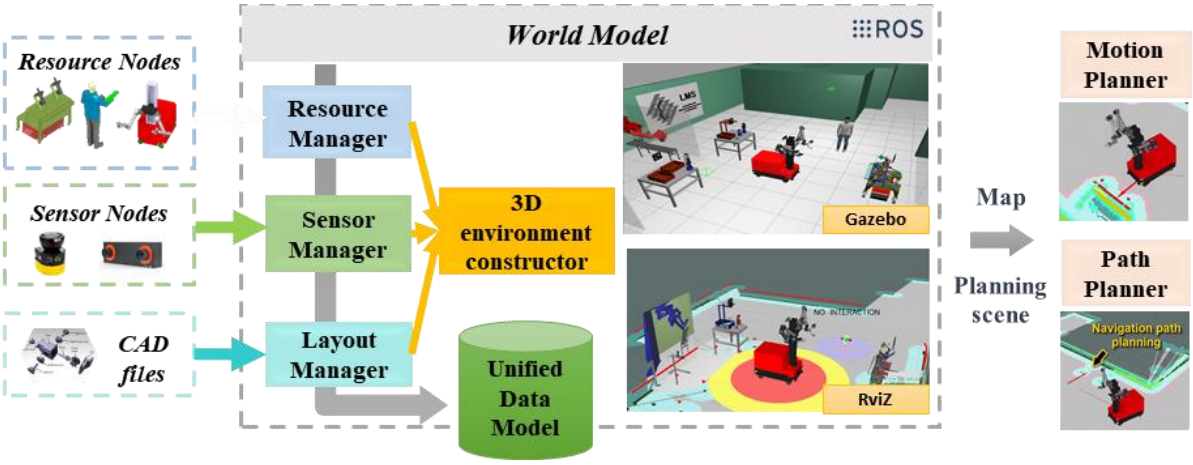


Figure 3: THOMAS World Model overall diagram

3.2. Final prototype

In the final prototype of THOMAS World Model, the main subcomponents have been remained the same as described in D5.2. However various updates and addition have been performed in order to meet the requirements of THOMAS Pilot cases. The following components have been updated.

3.2.1. Unified Semantic Data Model

All the semantic information that are stored and/or used from the World Model and other components inside THOMAS system has been organised under a common data model. This data model is presented in Figure 4. This model aims to capture the whole shopfloor state and is continuously updated from the World Model managers.

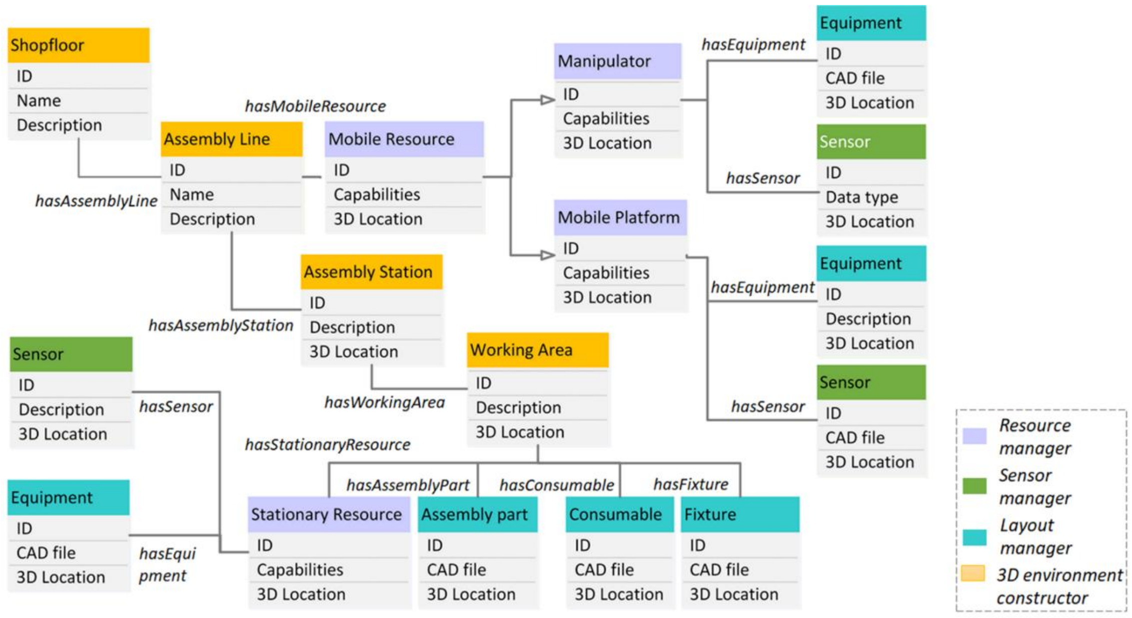


Figure 4: UML representation of World Model Data Model

3.2.2. Gazebo simulation environment

The simulation world that is provided from 3D environments contractor has been updated in order to match with the PSA demonstrator setup at LMS premises. The working tables models have been updated and all the AprilTags have been placed in the same position with the physical setup. In the following figure is presented the latest version of the Gazebo world for PSA demonstrator. Resource manager has been configured to monitor three resources Human operator, MRP and MPP. The digital models for these resources are also included in the environment construction procedure.

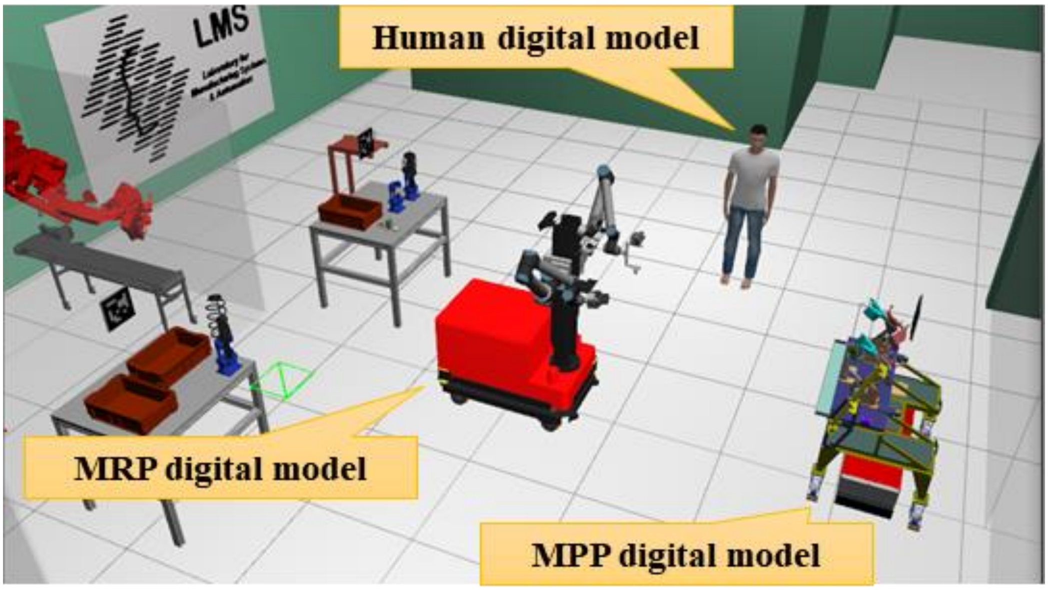


Figure 5: Gazebo PSA demonstrator layout

One addition that took place in the simulation environment but is used in the entire THOMAS system is the addition of static parts position in the global TF tree. As the Figure 6 shows some static

positions that are not changing dynamically for example the working areas, are published in the global TF tree and used from other components e.g. Navigation path planner.

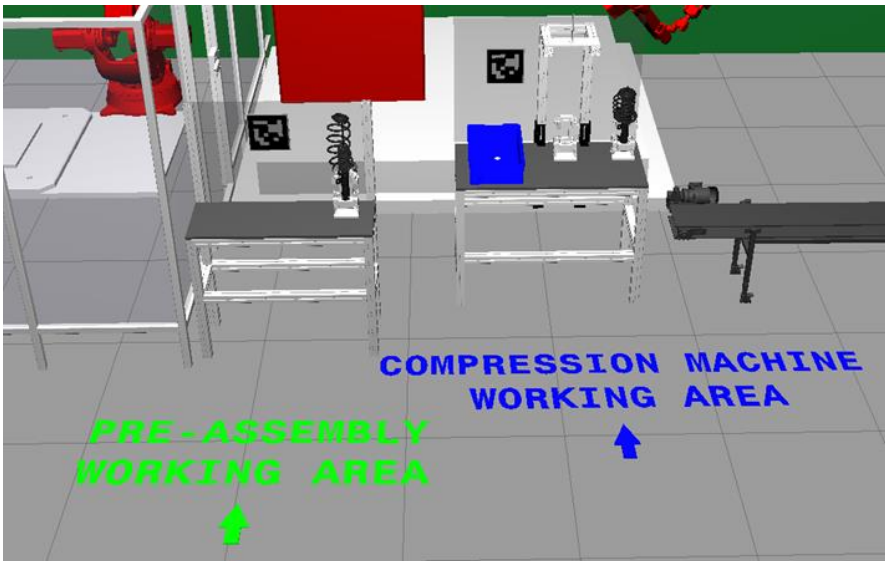


Figure 6: Docking points in front of working areas

3.2.3. Physical environment representation

The final version of THOMAS World Model contains many additions in the representation of the physical environment. All the information that is stored in THOMAS unified repository finally represented and visualized geometrically in RViZ as shows in Figure 7. Some of the latest additions are the visualization of the protective fields that comes directly from the Safety configuration and also the human detection visualization has been changed to an arrow. Station controller information regarding the status of the process execution is also available in the World Model. This kind of information is also consumed from the Human operator AR application. In Table 1 are listed the data that are able to be visualized.

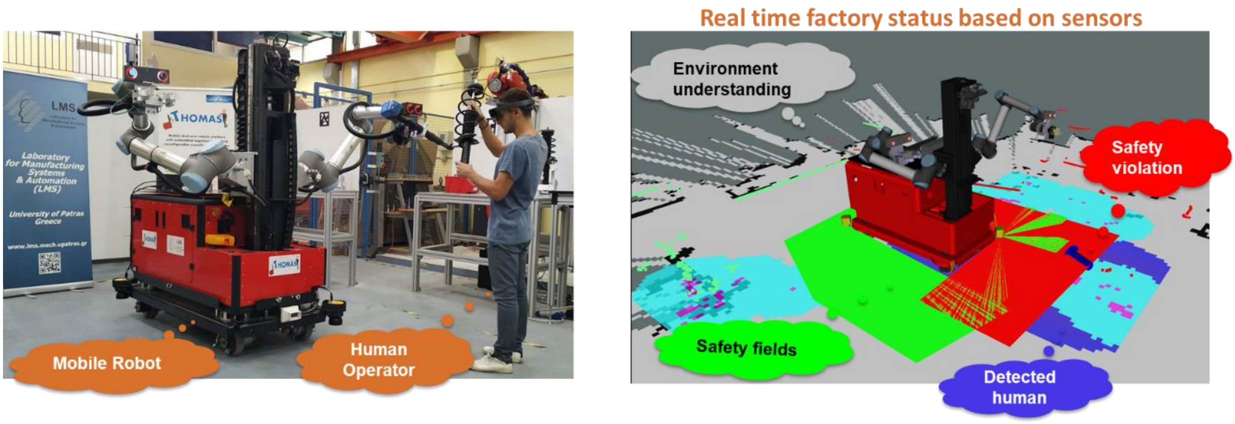


Figure 7: Representation of physical environment

Table 1: List of visualized data in RViZ

Data	Topic Name	Description
Robot Model	/robot_description	A visual representation of a robot in the correct pose (as defined by the current TF transforms).
Robot Motion planning	/moveit/planning_scene	Information about a planning scene and can animate solution paths within that planning scene.
Transformation Tree	/tf	The TF transform hierarchy.
Navigation Map	/map	An occupancy grid on the ground plane from a nav_msgs::OccupancyGrid.
Obstacles Costmap	/move_base/global_costmap/costmap, /move_base/local_costmap/costmap	An occupancy grid on the ground plane from a nav_msgs::OccupancyGrid.
Navigation path	/move_base/global_plan, /move_base/glocal_plan,	Data from a nav_msgs::Path message as lines
Lasers Scanner data	/front_laser/scan_filtered, /rear_laser/scan_filtered, /merged_scans	The data from a sensor_msgs::LaserScan message as points in the world, drawn as points, billboards, or cubes.
Detected Humans	/front _laser/people, /rear_laser/people	The position and the velocity of the detected humans displayed as arrows with the use of visualization_msgs::Marker messages
Protective Fields	/front _laser/fields, /rear_laser/fields	The area that covers each protective field displayed with the use of Polygon visualization_msgs::Marker messages
2D Image	torso_realsense_camera/rgb/image, rc_160/ rgb/image, rc_65/ rgb/image	Displays an image from a sensor_msgs/Image topic, similar to image_view
3D Image	torso_realsense_camera/depth/points, rc_160/depth/points, rc_65/depth/points	Displays a point cloud from a sensor_msgs::PointCloud2 message as points in the world, drawn as points, billboards, or cubes
Station Controller's informarion	/mrp/status, /operator_#ID/status, mrp/current_task, /operator_#ID/current_task,	All these topics provides textual information which is presented somewhere statically in the enviroment.

4. THOMAS Station Controller

4.1. General Overview

The station controller is responsible to coordinate the order of execution at line-level. The station controller implements the Execution Coordinator which is responsible to enforce the required sequence of tasks execution and also keep track of the new, pending and active tasks within the Station Controller. The station controller execution process is interactive and dynamic. In this sense the Station Controller can respond to unexpected events in a coherent way that minimizes the impact to the orchestrated tasks.

The Station Controller accepts as input a schedule which includes a set of Tasks assigned to resources. Each task can be broken down to actions that are orchestrated and tracked individually. The description of the resources inside the schedule is only detailed enough to allow the reference of these resources and the communication of the station controller with these resources by using the ROS Action Lib protocol.

Therefore the resources that need to orchestrated by the station controller need to have a related ROS Lib Action Server implemented as the example that is presented in Figure 8. In this diagram one can see a set of different resources and their related action servers. Each of these action servers is connected to the THOMAS Network of services in order to coordinate a resource following the Action Lib Protocol.

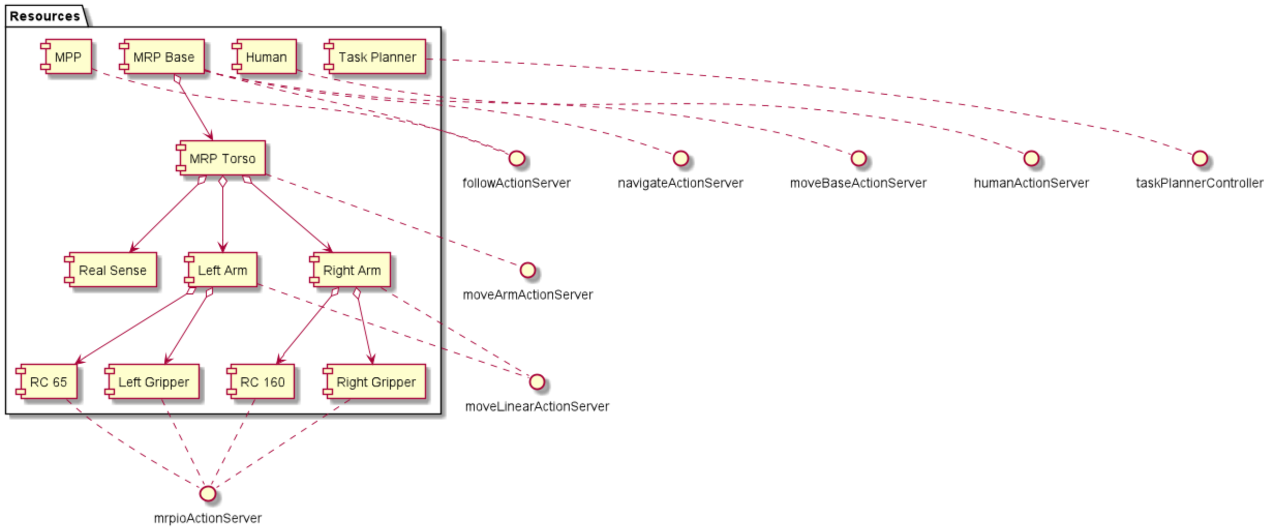


Figure 8: Resource's Action Servers

4.2. Station controller architecture

The conceptual architecture of the Station Controller is described in deliverable D5.3 named "THOMAS Service Oriented Network of Resources – Initial Prototype". This section provides some a further description of the implementation architecture of the Station Controller while also including some key elements of the conceptual architecture for reasons of completeness. The Station Controller architecture is built on top of the actor model abstraction.

The actor model abstraction allows you to think about your code in terms of communication, not unlike the exchanges that occur between people in a large organization.

Use of actors allows us to:

- Enforce encapsulation without resorting to locks.

- Use the model of cooperative entities reacting to signals, changing state, and sending signals to each other to drive the whole application forward.
- Stop worrying about an executing mechanism which is a mismatch to our world view. [2]

While the Station Controller has been built on top of the actor model abstraction, its interaction with other systems follows a service-oriented approach. The service-oriented approach followed aimes to be suitable for orchestrating a wide variety of human robot collaboration that includes both stationary and mobile robots and are can be applicable to different manufacturing environments.

The service-oriented approach has been proof tested in other settings that involve mobile robots. For instance in [3] a service oriented, web based software has been developed for the monitoring of the shop floor status and the parts' supply dynamic scheduling, based on time and inventory. The proposed system has been applied to a case from the automotive sector demonstrating the ease of deployment and efficiency in the co-ordination of the mobile units' operation.

The architecture of the station controller follows a three-tier approach that can also be found in IoT systems. For instance in [4] the authors propose a services framework that utilizes the three-tier industrial IoT system architecture that enables services' operation both at the edge and in the platform tier and uses the enterprise tier for implementing specific applications that provide interfaces to end-users. According to them this is a well-established system architecture pattern for implementing coherent industrial applications.

Figure 9 shows a conceptual diagram of the Station Controller Architecture. In principle the Station Controller is mainly run by the Station Controller Actor. The Station Controller actor will create a different instance of a Schedule Execution Actor for every individual schedule that it needs to run. Then each schedule actor will create a set of action actors. Each action is represented by a specific action actor, that implements the logic to execute the action and each action actor knows the task that his action belongs. Every action is always part of a single task. The Schedule Execution Actor for each schedule also maintains a reference table that allows him to send messages to all action actors that belong to a specific task.
A key element in every action actor implementation is the ROS Lib Action Client that is capable to communicate with the related ROS Lib Action Server.
The Station Controller implementation is distributed, so as shown in Figure 9 the different ROS Lib Action Servers can be deployed in different computing hardware. Each ROS Lib Action Server is connected to a particular resource and is responsible to communicate its status to the related clients.
The Execute Schedule Action Server is part of the Station Controller Actor and is the API by which the Station Controller is notified for Schedules that need to be run.
In a similar fashion the Task Planner Action Client is connected to the Task Planner Action Server. In this way the Station Controller can request a rescheduling of a set of Tasks to the Task Planner.
Another important feature of the Station Controller is its inherent ability to operate in a multi master environment. The different action servers of the resources need not be in the same ROS Master. The Station Controller could also interoperate with different Task Planners that have different Action Servers, as long as the ROS Lib Action interface is the same.
However, the Station Controller should is tied to a specific ROS Master for the Execute Schedule Action Server in order to receive Schedule Execution Requests.

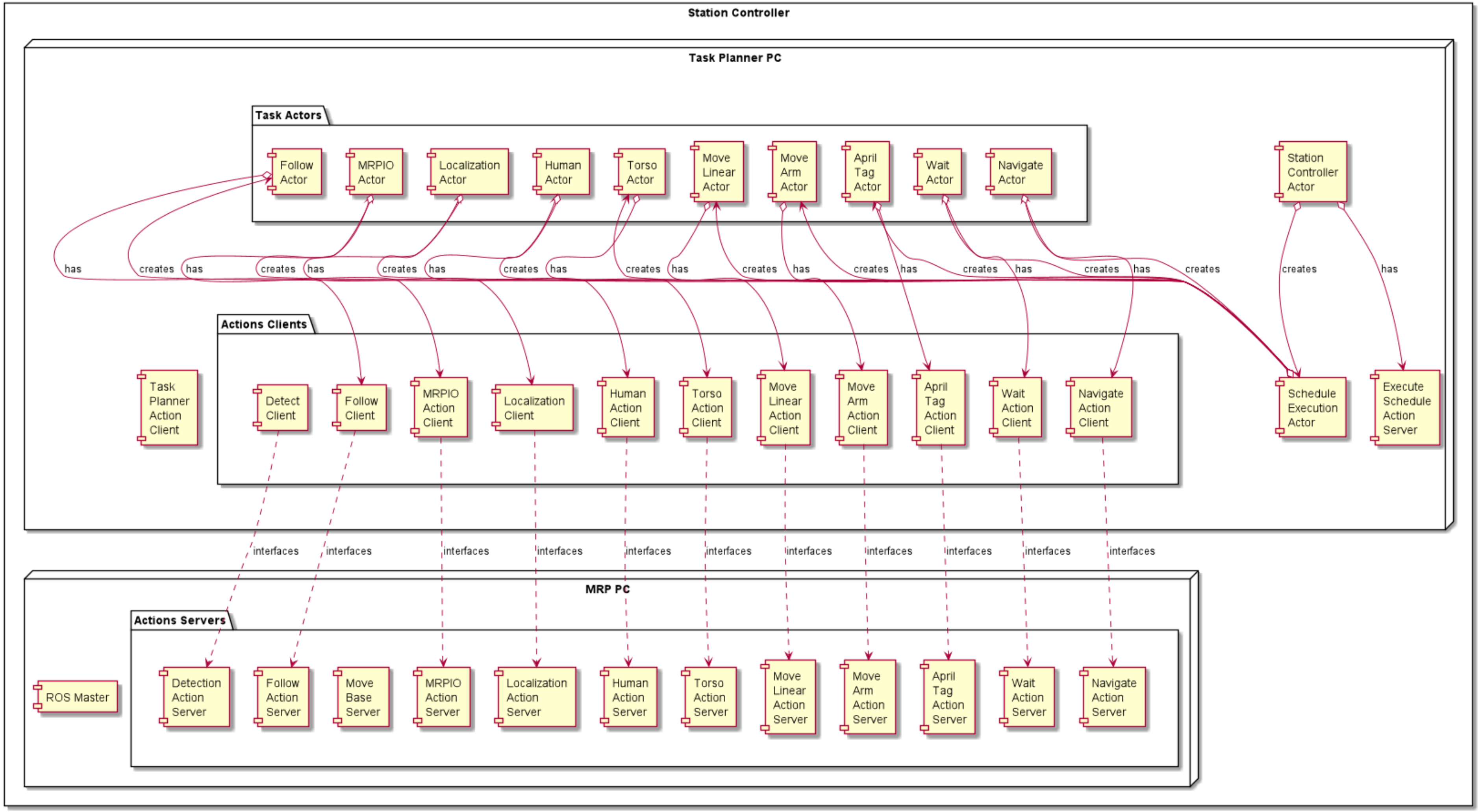


Figure 9: Station Controller Diagram

4.3. **Station Controller API**

Station Controller offers a set of ROS Lib Action Servers and Services that facilitate the integration of the THOMAS system. The different Action Servers and services are described below. This section provides a description of the Station Controller for completeness. As the Station Controller API is part of the final prototype of the THOMAS Network of Services, the complete details of the API offered by the station controller is provided in detail in deliverable D5.4 “THOMAS service based integration and communication network – Final Version”.

4.3.1. Schedule Execution Control Integration

The Station Controller provides an action that allows the caller to initiate a command related to the execution of a schedule as well as to track its status of execution.

In particular the commands that can be executed by the station controller in respect to a given schedule are the following:

- START
- STOP
- PAUSE
- CANCEL

4.3.2. Task Take Over Integration

The Station Controller supports Human Robot Interaction for Task Take Over via a Smartwatch application were developed. The final version of the Task Take Over integration is differentiated in respect to the previous prototype in respect to the API. In particular the final prototype uses a ROS Action as an API for Task Take Over as an enhancement in comparison to the services defined in D5.3. The current improvement allows the task take over to take place asynchronously, in a fire and forget request. Moreover, the Task Take Over action allows for the Simultaneous request of multiple tasks to be taken over. This API is used as the Human interacts with the Robot using the aforementioned Task Take Over functionality implemented in the station agent that controls the MRP.

4.3.3. Get Tasks Available for Take Over

For a given schedule under execution not all tasks can be taken over at any time. Tasks and actions that have been executed or are under execution limit the options available. For instance, a completed task should not be taken over. For this reason, the Station Controller. A service is offered to allow the Human Operator Software to know which tasks can be taken over.

4.3.4. Information Providing Services

The station controller offers a set of information providing services. These services are offered that provide information about the shopfloor. These are the following:

- **/station/resource_list**: to download the available resources at the current time instant and select the appropriate resource among them.
  The same service is used so the user can login by selecting the human resource she corresponds.
- **/station/task_list**: to download the assigned tasks for a particular resource

5. THOMAS TASK PLANNER

5.1. General Overview

The core decision making that facilitates the cognitive aspects of the line level work re-organization is the HRC Dynamic Task Planner. The Task Planner is triggered to re-plan and re-allocate the required tasks to the existing resources. The Task Planner triggering may be outcome either of human input / request or any other unexpected event such as resource breakdown.
The implementation of the Task planner is mainly based on information about each task saved in a database. Also feedback from the mobile resources motion and path planners is used, given that the final goal is to minimize the time required and the distance travelled for each task execution targeting on maintaining sustainable cycle times.
Through the implementation of the search-based algorithm (described in detail under D5.1) for multiple alternative generation, multi-criteria decision-making mechanisms have been integrated for evaluating the multiple generator alternatives based on user defined criteria.
The assignment procedure is executed by the HRC Dynamic Task Planner following the next steps:

1. **Definition of possible assignments** (alternatives) of tasks/operations to resources at a decision point according to. A decision point occurs whenever there is a status change of the system. In order to enable human-robot teamwork, a human operator has also the capability of triggering the decision-making algorithms by asking the robot to execute an operation, via proper communication interfaces with the scheduling system. If this is the case, the working conditions and related decision-making parameters are evaluated and if feasible a new alternative sequence of assignments is generated to match the worker's request. Possible decision points are predetermined in time by using time stamps according to a set of rules related to the current assembly specifications. The procedure continues until the maximum number of alternatives, edited by the user, is reached. Also, the alternatives are generated only within the time interval from the current decision point until the second next possible decision point, in order to allow human to add a new decision point in the middle.
2. **Selection of criteria** for the assignment of tasks/operations to resources. For example, the suitability of the resources may be evaluated according to parameters such as the payload and the reachability of the robot as well as the availability of each resource. The criteria selection is performed using a rule-based system. The rules may be set to reflect the overall production objectives for the current facility.
3. **Evaluation of generated alternatives** according to the selected criteria. According to the production objectives different weights may be used for the evaluation. For example, if the objective is to minimize the cycle time then the time criterion will have bigger weight than then other criteria such as operators fatigue or competence.
4. **Selection of the highest scoring alternative**. After scoring all alternatives in the previous step the highest ranking one is selected as the most suitable to match the current objectives.

After defining all the possible alternatives, all the different scenarios have to be evaluated, based on some selected by the user criteria, such as:

a) Flowtime

$$F = \sum_{i=1}^{n} (f_i + t_i)$$

Where:
F= Flowtime
f_i=Each operation's processing time
t_i=Time between operations n=Total operations

Target → Minimize F

b) Payload handled by human

$$P = \sum_{i=1}^{m} pw_i$$

Where:
P= Weighted Payload
pw_i=Each part's weight
m=Total operations done by humans

Target → Minimize P

c) Utilization

$$U(i) = \left[\frac{T_i}{A_i}\right], \qquad i = 1, \dots, r$$

Where:
U= Utilization
A_i= Each resource's availability
T_i=Each resource's busy time
r=Total resources

Target → Maximize U(i)

d) Distance covered

$$D(i) = \sum_{j=1}^{n-1} d_{j+1,j} \quad i = 1, \dots, r$$

Where:

- D= Distance covered
- $d_{j+1,j}$ =distance between two consecutive operations' locations done by the same resource
- n=Total operations done by each resource
- r=Total resources

Target → Minimize D(i)

e) Non adding value activities time

$$I = f - \sum_{i=1}^{n} f_i$$

Where:

- I=Non adding value activities time
- f=Flowtime
- f_i=Each operation's processing time
- n=Total operations

Target → Minimize I

Following, to identify the most preferable solution, the utility value for each alternative is calculated through the following equation.

$$U = \sum_{j=1}^{n} W_c C_j$$

Where, Wc is the criterion's weight factor, C_j the value of each criterion and n the number of criteria. The alternative with the highest utility value is the most preferable.
The input for the decision-making framework includes the resources, the tasks, the task-resource suitability, the task precedence constraints, the duration of tasks and the time of starting and completing a task. The proposed decision-making framework selects which resource will be assigned for the execution of a task based on the suitability of the resources.
There are at least one or more suitable resources for a certain task, evaluated by the corresponding algorithm. The suitability of the resources is decided upon the human skills, such as flexibility, problem solving capabilities, complex perception and manipulation, and the robot capabilities, namely repeatability, efficiency, accuracy, high payload capability. The suitability algorithm evaluates the availability of the resources and their ability for tools/grippers handling are considered.
The alternative solutions that are selected from the decision-making framework are evaluated against multiple criteria in order to select a solution in a short time frame. The criteria selection is based on the requirements and specifications of the user. There is no restriction on the kind and number of criteria that can be selected.
Moreover, in case of an abnormal event the system may reschedule the remaining or the entire task according to a predefined set of rules and constraints. In this case, depending on the task's nature and its stopping point, the manufacturing process may adapt and continue its execution, by re-assigning a number of operations to a different resource.

5.2. Architecture

The general architecture and internal involved components are listed below:

1. **Data Model**
   All the information that describes the Front Axle Assembly model is inserted manually into a database and is used to execute the task planning. Information about all levels of assignment (Plan→Process→Task→Operation→Action) and the connections between them, as long as information about the resource compatibility, existing workstations and needed tools existed in the database. Finally, all parts existed in the model, along with their parameters are also defined in the database. A small sample about these tables that form the database is presented below in Figure 10.

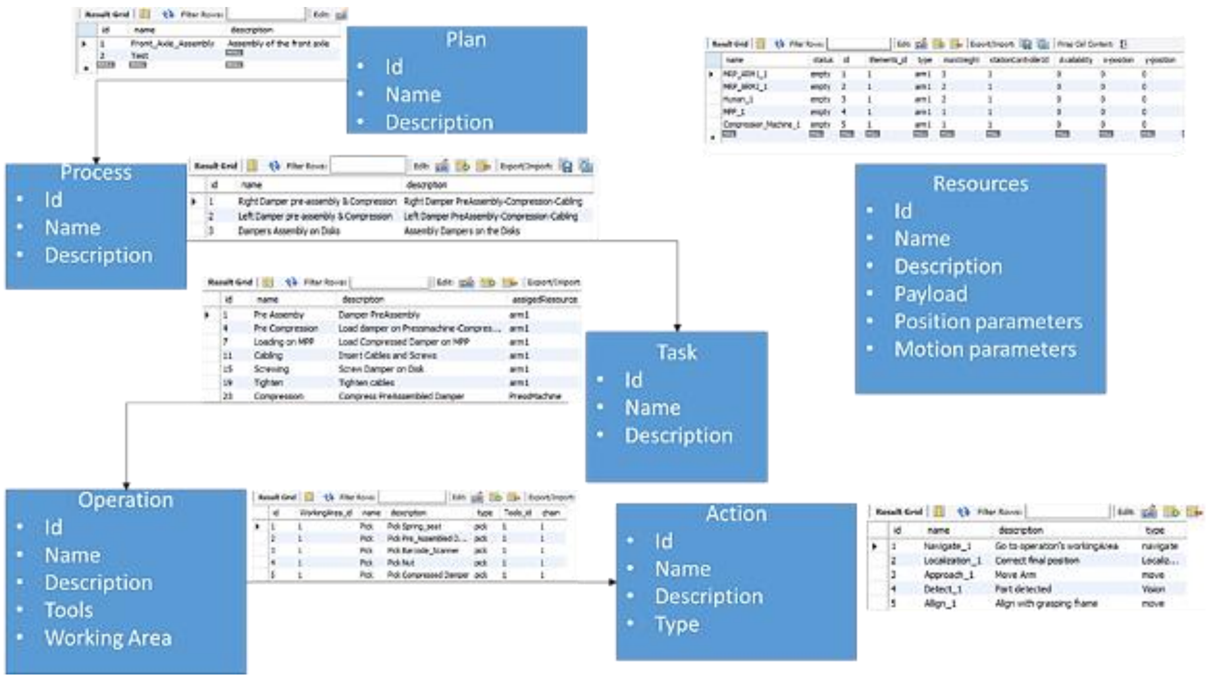


Figure 10: Task Planner Database

2. **Find compatible resources for a task**

The resource suitability evaluation (Figure 11) aims to reduce the scheduling time by providing a quick and simple method for classifying the operations suitable resource type, thus eliminating the need for further time-consuming evaluation of impossible plans. The criteria that are used for the initial task allocation are derived from the part characteristics, such as weight, size and flexibility which are cross-checked with some basic resource type characteristics, as acceptable strength, payload, and level of dexterity. Finally, the tool needed for each task must be part of the resource's compatible tool list.

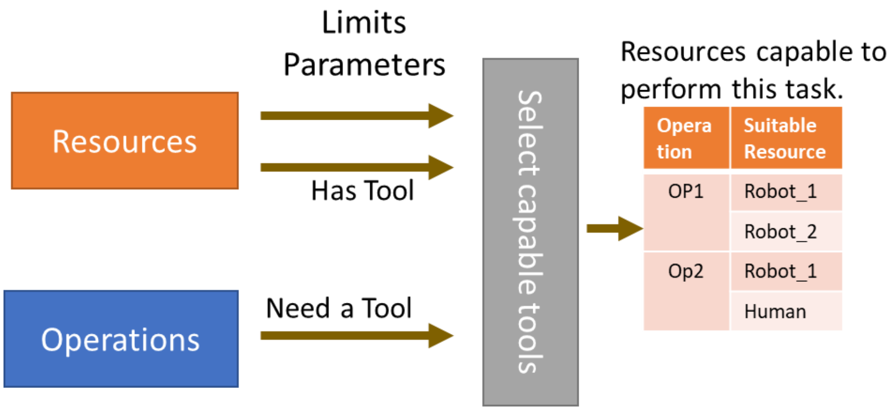


Figure 11: Resource suitability evaluation

3. Fill templates

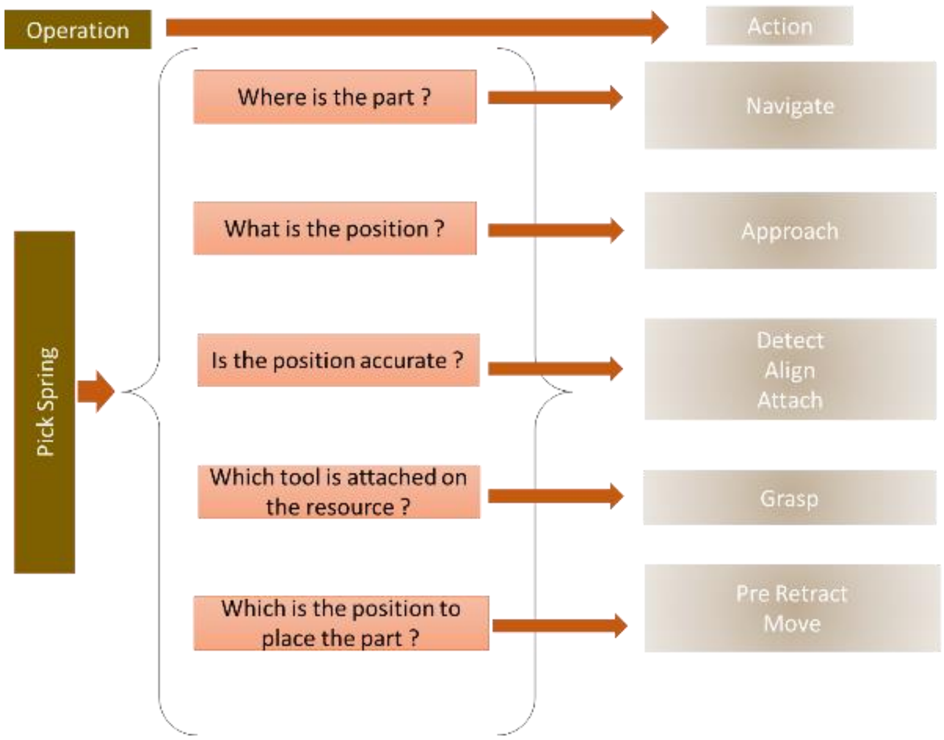


Figure 12**: Templates filling**

5.3. Task planner Services

The Planner uses information about the plan, the available resources and the existing working stations, which is saved into the database. Whenever there is a status change of the system, the decision-making algorithms are triggered. After generating and evaluating all the alternatives, the Task Planner returns as a result a list of operations and actions needed to be executed to complete the plan, along with the resources that every operation and action is assigned to and the working station each operation and action took place.

The implemented prototype in JAVA has been tested on the automotive assembly scenario. In particular, the workload for the front axle assembly line has been used as input for the planning module considering as available resources: one human operator, two MRPs and two compression machines.

Dedicated Graphical User Interfaces have been implemented so for the user to import the workload and facilities related data following the hierarchical modelling performed during the design phase.

The first step is to start the server. By doing that a localhost(“localhost:4200”) is created in Chrome, which is the environment through which the user interacts with the Planner. There the user chooses chooses whether he wants to see information about the digital world model, or to “run” the Task Planner.

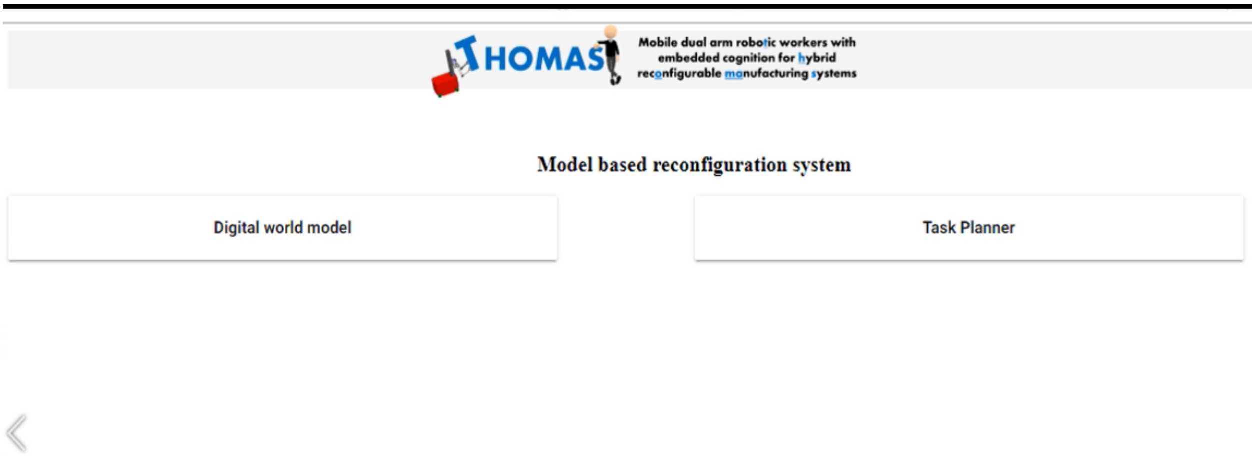


Figure 13: Plan Choice Menu

Upon selecting the Digital world model option the user has access on the elements save in the database. The high-level information that is shared through this data base includes: a) The digitalization of the factory describing the included assembly lines that breakdown into working stations. The available resources are also described, b) The digitalization of the process which includes all the required task to be performed for the assembly of the different products and relevant sub-assemblies in the referenced factory and c) the visualization of the 3D environment. Indicatively, Figure 14 the Graphical User Interface that visualizes as an example an assembly line which is composed by specific working stations which in turn may involve different working areas. Selecting the “MPP working area” the user can see the available resources in this working area.

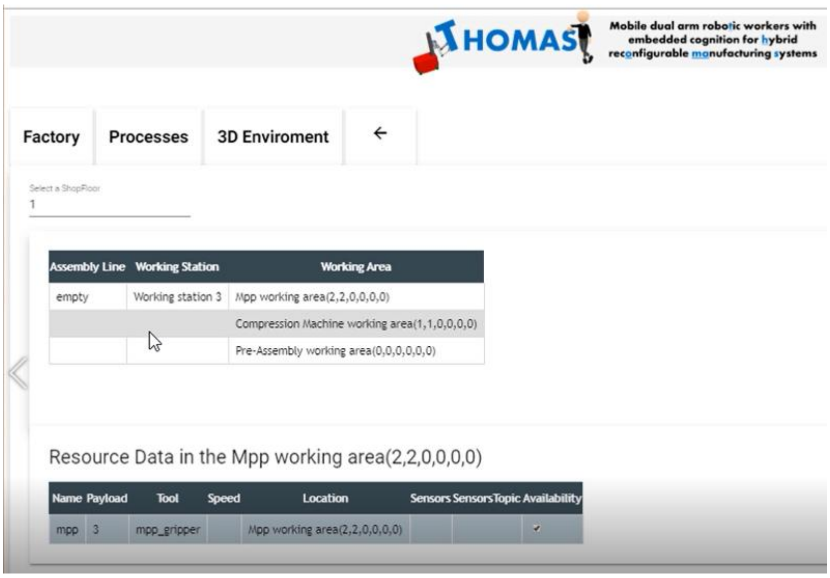
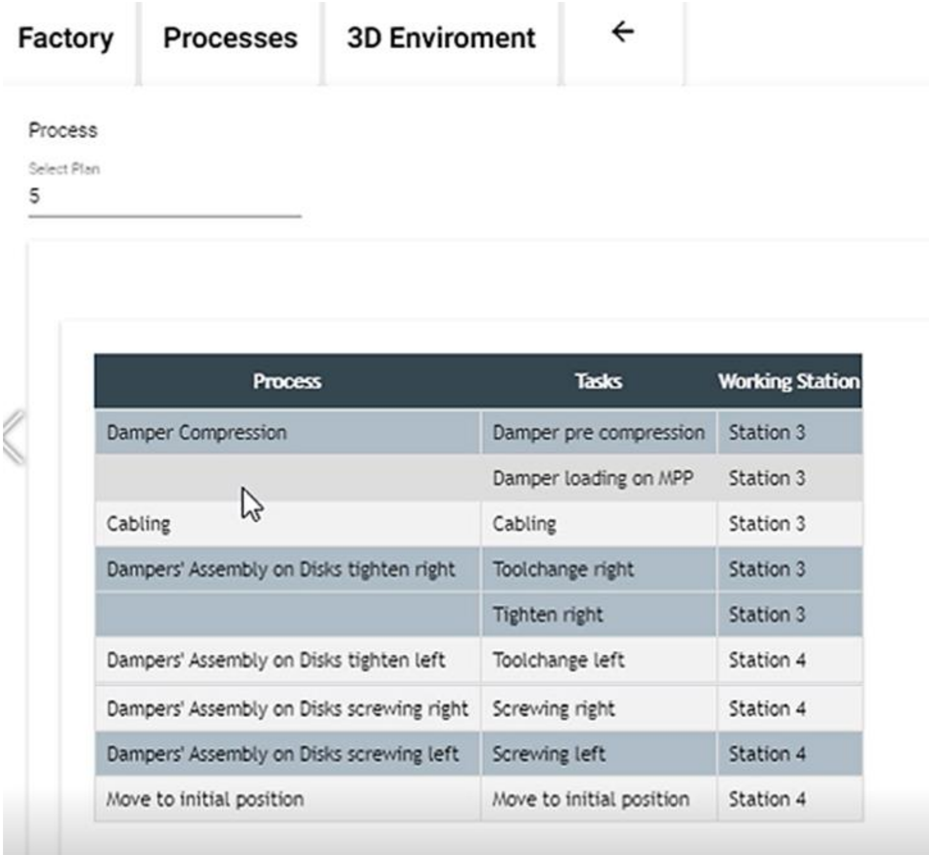


Figure 14: View working stations

Upon clicking on the “Processes” tab, the user may select the to access the assembly plan of the products available in the databased. Figure 15 visualizes indicatively the breakdown of the assembly plan for the assembly of the left and right damper on the disk that takes place in the front axle assembly line.



Process	Tasks	Working Station
Damper Compression	Damper pre compression	Station 3
	Damper loading on MPP	Station 3
Cabling	Cabling	Station 3
Dampers' Assembly on Disks tighten right	Toolchange right	Station 3
	Tighten right	Station 3
Dampers' Assembly on Disks tighten left	Toolchange left	Station 4
Dampers' Assembly on Disks screwing right	Screwing right	Station 4
Dampers' Assembly on Disks screwing left	Screwing left	Station 4
Move to initial position	Move to initial position	Station 4

Figure 15:View list of Processes-Tasks-Working Areas

Under the tab “3D environment” the user can visualize the digital world model of the working station in the assembly of the selected part (Figure 16).

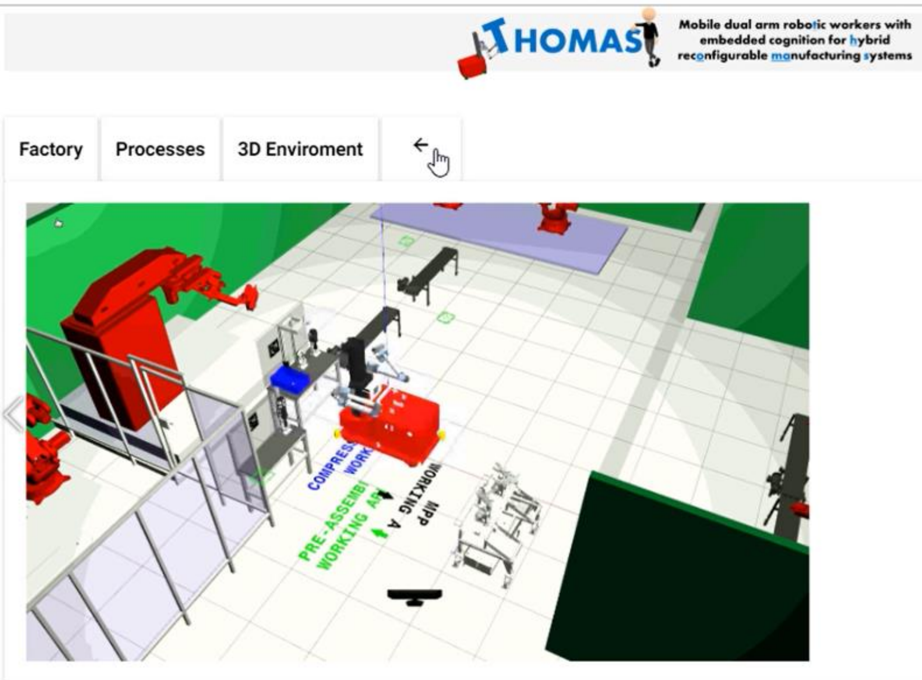
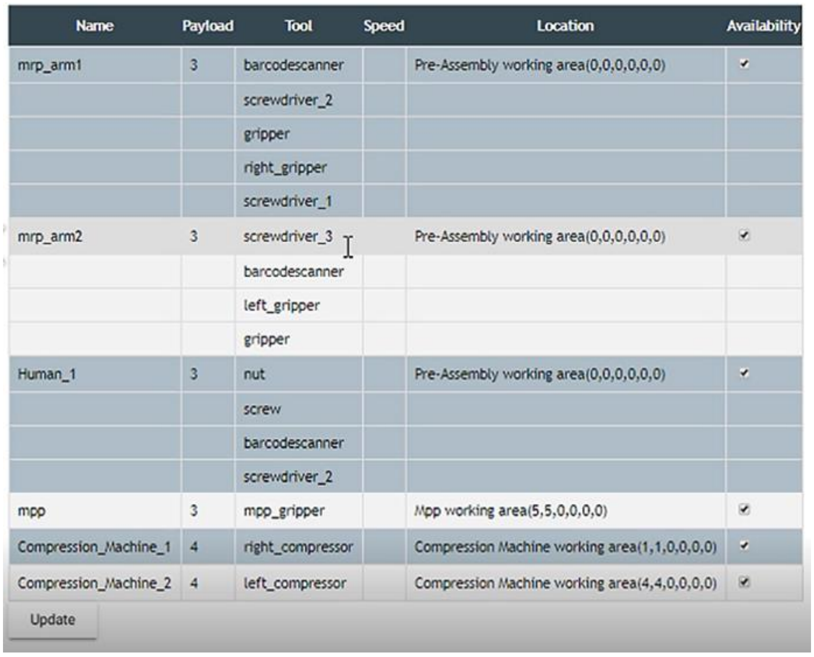


Figure 16:View digital world model

By clicking the “Task Planner” option of Figure 13, the user is able to select the assembly processes he/she prefers to “plan”. Then”, the list of processes that constitute the chosen assembly sequence appears and by clicking the option “view” the user can see the list of processes, tasks and working stations that form the Plan (Figure 15).

Moreover, by clicking “View resources” button, a list of the available resources is pop up in the GUI. The available tools and their current location and availability are provided to the user.



Name	Payload	Tool	Speed	Location	Availability
mrp_arm1	3	barcodescanner		Pre-Assembly working area(0,0,0,0,0,0)	✔
		screwdriver_2			
		gripper			
		right_gripper			
		screwdriver_1			
mrp_arm2	3	screwdriver_3		Pre-Assembly working area(0,0,0,0,0,0)	✔
		barcodescanner			
		left_gripper			
		gripper			
Human_1	3	nut		Pre-Assembly working area(0,0,0,0,0,0)	✔
		screw			
		barcodescanner			
		screwdriver_2			
mpp	3	mpp_gripper		Mpp working area(5,5,0,0,0,0)	✔
Compression_Machine_1	4	right_compressor		Compression Machine working area(1,1,0,0,0,0)	✔
Compression_Machine_2	4	left_compressor		Compression Machine working area(4,4,0,0,0,0)	✔

Update

Figure 17: View list of available resources

By clicking the “view criteria” button, a list of the available criteria is shown up, in order for the user to select the ones he prefers.

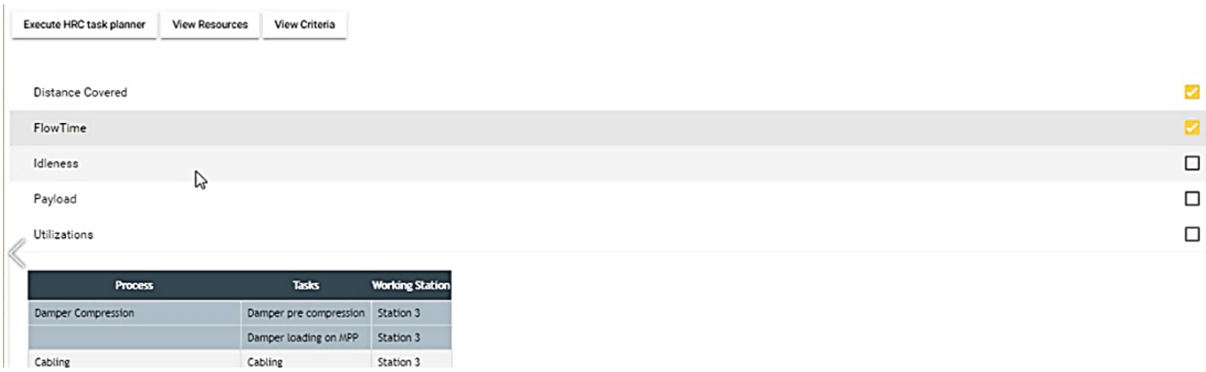


Figure 18: View list of criteria

In order to run the Planner, the user has to click “Execute HRC task planner”. By doing that the list of operations along with the assigned, in each operation, resources and working station is shown. Figure 19 visualized an indicative generated schedule where the tasks are assigned to one human operator and one MRP. The tasks assigned to MRPs are further detailed to operators so to specify which actions need to be undertaken by which MRP arm.

Task	Operation	Resource
Damper pre compression	Demo_Pick pre-assembled damper	mrp_arm1
	Demo_Navigation to compression machine	mrp_arm1
	Demo_Detect ficture	mrp_arm2
	Demo_Place pre-assembled damper	mrp_arm1
	Demo_Move to home pos left	mrp_arm2
	Demo_Pick upper cup	Human_1
	Demo_Place upper cup	Human_1
	Demo_Pick alignment rod	mrp_arm2
	Demo_Place alignment rod	mrp_arm2
	Demo_Pick nut	mrp_arm2
	Demo_Insert nut	mrp_arm2
Damper loading on MPP	Demo_Pick compressed damper	mrp_arm1
	Demo_Place compressed damper on MPP	mrp_arm1
Toolchange right	Demo_Right change tool to screwdriver	mrp_arm1
Toolchange left	Demo_Left change tool to screwdriver	mrp_arm2
Cabling	Demo_Pick cables and screws	Human_1
Tighten right	Demo_Tighten cabling right	mrp_arm1
Tighten left	Demo_Tighten cabling left	mrp_arm2
Cabling	Demo_Place cables and screws	Human_1
Move Mpp	Demo_Move MPP	mpp
Screwing right	Demo_Screw right damper	mrp_arm1

Figure 19: View list of operations-resources-working areas

6. STATION CONTROLLER AND TASK PLANNER DYNAMIC INTERACTION

6.1. Introduction

The developed final prototype has been used in a pilot and test case for the assembly of a passenger vehicle's front axle. The study considers four stations of this line, as presented in Figure 20: Station 1 - Right Damper Assembly, 2) Station 2 - Left Damper Assembly, 3) Station 3 - Screwing and 4) Station 4 - Cabling stations. In these stations various tasks are performed sequentially and most them are performed manually.

The introduction of the autonomous dual arm robots (MRPs) and the Mobile Product Platforms (MPPs) requires several updates in the manufacturing setup. The MRPs will act as assistants to human operator during the assembly of the dampers since they will be handling the heavy parts and the screwing tasks for all variants, while the MPPs will act as a moving "workstation" for human operators and MRPs. This approach (Figure 21) targets in saving time and space relative to the original one.

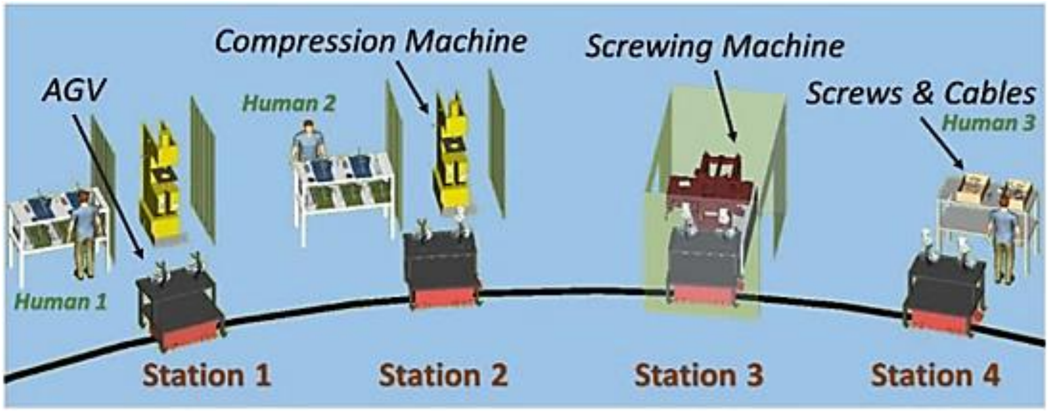


Figure 20: Front axle assembly line – Manual Assembly

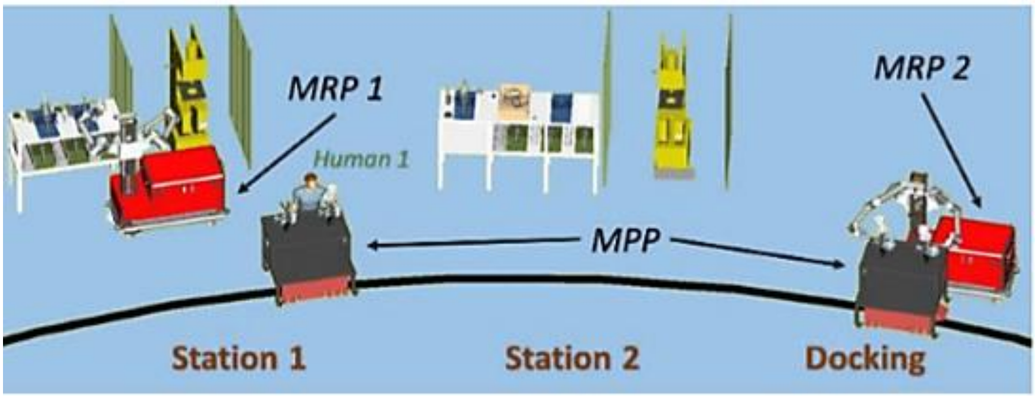


Figure 21: Front axle assembly line – Mobile Robot Assembly

6.1.1. Involved Resources

The developed prototype follows a hierarchical model for the resources involved in the assembly. Figure 22 illustrates an example of a hierarchical resource model, which includes mobile and stationary resources. The mobile resources comprise humans and robots, whilst assembly table,

fixtures etc. constitute the stationary resources. The suitable grippers and tools are the latest layer of active resources, when assembly components, fixtures and parts belong to passive resources.

Through the resource suitability evaluation, the plan's workload model is created (Figure 23), where is evident, which resource is able to take over each task.

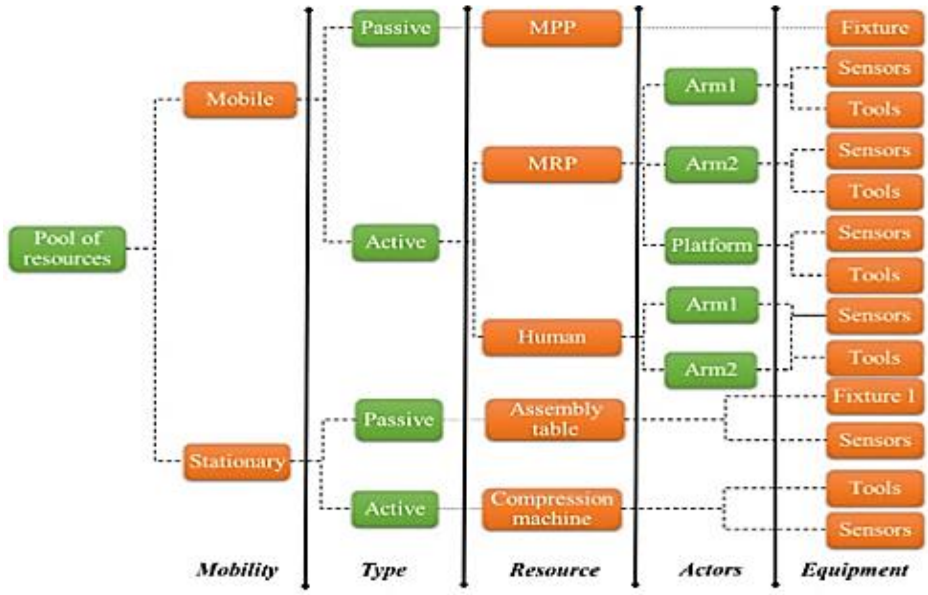


Figure 22: Resource model

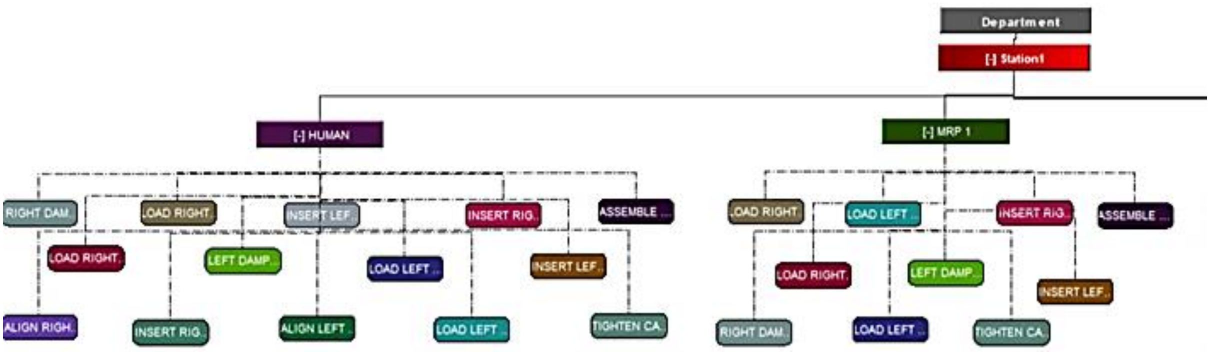


Figure 23: Front axle's assembly resource model

6.1.2. Required Tasks

In order to model the assembly process in the developed prototype, the same principles with the resource modelling have been used. Under these principles the workload activities have been modelled at a five-level breakdown. As shown in Figure 24 , the upper level is the Plan, which is decomposed in a set of Processes. A Process involves a group of Tasks, which constitute a group of Operations. Finally, each Operation consists of a group of Actions.

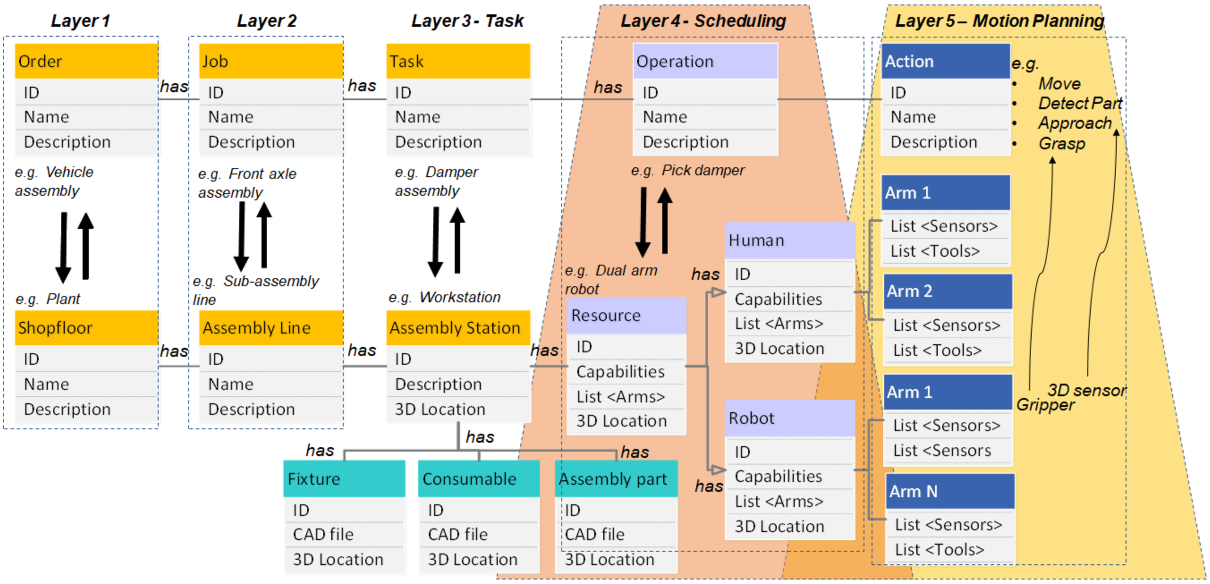


Figure 24: Workload model architecture

The tasks that constitute the case study are presented in Figure 25. The task list along with information about each task, such as each task's operations, workstation, compatible tools and constraints or precedence relations, are saved into the database and are used as input data for the task planning algorithm, in order to create the plan's alternative tree. The plan's workload model (Figure 25) shows these relations between each task as arrows.

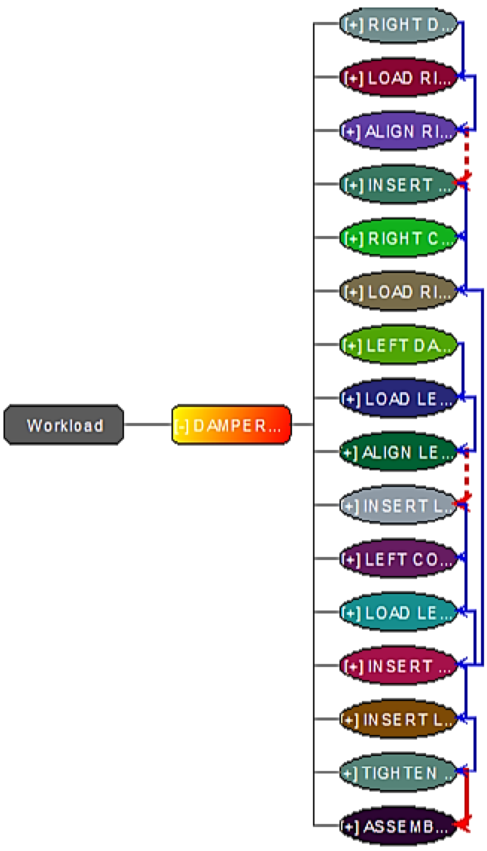


Figure 25: Workload model

6.2. Task Planning

Description of the planning process

The task planning will be used into two different categories of application. One case is the request of planning for a specific task and the other case is the rescheduling case. In this case a specific scenario will be stop, for example in case of a robot malfunction, and the task planner will have to reschedule the remaining scenario for the available resources.

A) Initial planning
   a. For the initial planning the task planner only needs to get the id of the requested process. Using this “id” the planner can access the world model though the database. The “id” will be sent to the task planner application using a socket request.
B) Rescheduling
   a. In case of the rescheduling the task planner needs information about the current status of cell. In more details the task planner needs the last task that have been executed in the cell, so the remaining task can be determined. Since this is an emergency case the updated list with the resources should be sent.
C) Generate the alternative and evaluation.
   a. During the task planning (in both case) the evaluation of the alternative needed to be made. During the evaluation the task need to call the simulation tool so to acquire information about the task planner and the resources.

6.3. Task Plan Execution

The task planning execution process is an interactive process that involves the Task Planner, the Station Controller, the THOMAS Network of Services and the different THOMAS Resources and Sensors.

Figure 26 shows a conceptual overview of these entities in an abstract level that is focused to describe the overall process. In particular the first step would be the task planning process, this starts when an external system, such as an ERP, requests the scheduling and execution of a set of tasks. In a real-world scenario, the request should come from outside the system boundary, but it is also possible to use the system provide UI to initiate a scheduling and execution process.

The task planning is executed as described in section 6.2. Then the Task Planner generates an executable schedule in JSON format. The Executable Schedule describes the sequence of Tasks execution, described in the action level based on the workload model that is shown in Figure 24: Workload model architecture.

Then the station controller executes each task and action. The execution of each action is tracked individually using independent actors. The use of actors in the Station Controller Architecture is described in section 4.2.

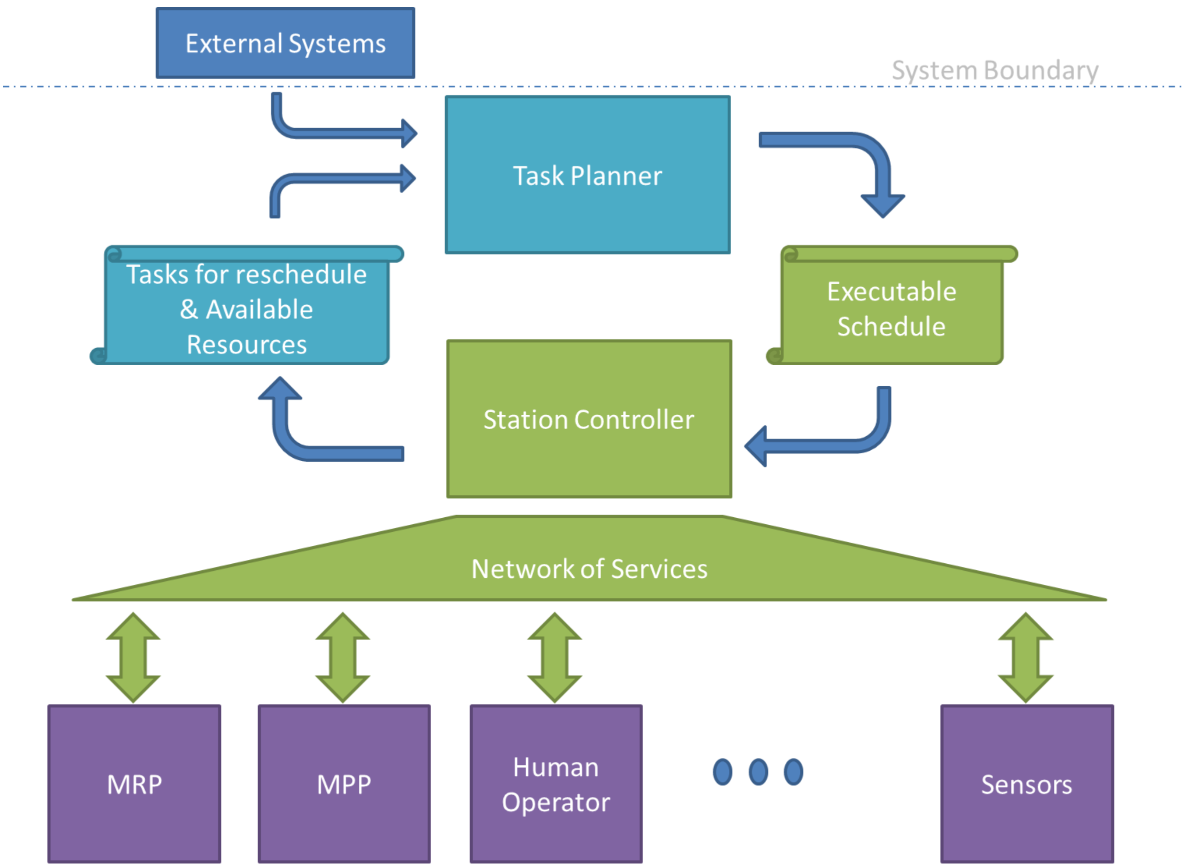


Figure 26: Conceptual Task Plan Execution

The concrete implementation of each of the resources shown in Figure 26can be practically a set of independently coordinated hardware and software modules. For instance, the MRP resource design as shown in Figure 27, includes several operation, network and safety subsystems.

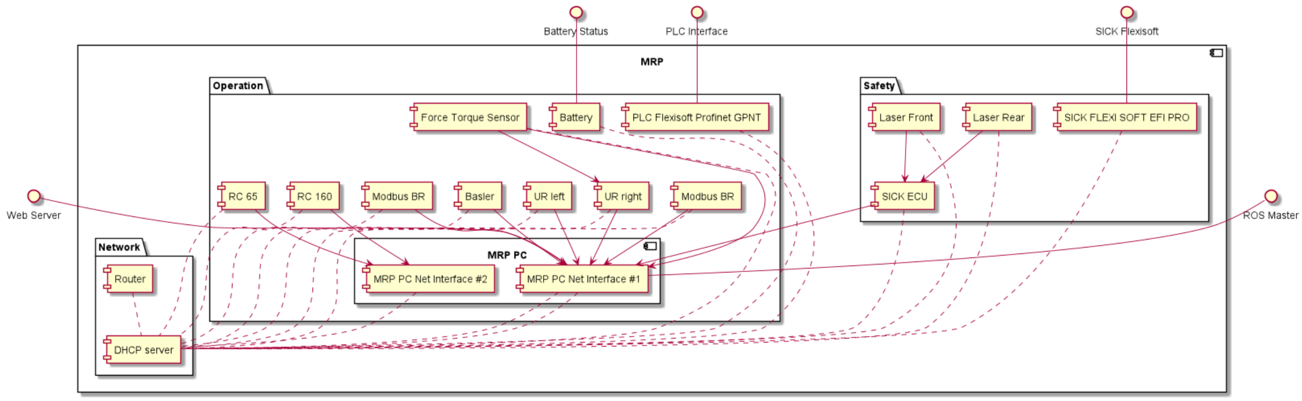


Figure 27: MRP Resource Design

Each actor manages a specific action and is always running independently from all other actions. Each actor will execute a set of operations on the completion of the action that he is responsible. A key actor operation would be the sending of a message to the actors that are responsible for the actions that follow to notify them that they should start the execution of the actions that they are responsible.

The behavior of each actor can be modified by several configuration parameters that are available on each action. The set of operation parameters are responsible to control the behavior of actors are shown below.

Max Retries: The actor will retry to execute the action, in case the resource is unable to execute it for any reason. If the max retries number is zero, the action actor will notify the station controller that the

Retry Wait: The time in milliseconds that the actor will wait before retrying to execute a particular action.

Can Pause: If the action can be paused then this parameter is true. If the action cannot be paused for some reason, then this parameter should be set to false. Actions that cannot be paused are normally binary in nature, very fast or occur momentarily such as the open and closing of a relay. If not explicitly specified, then this parameter is by default true (actions can be paused by default).

On Resume Action: Each action can be linked to a resuming action. In case a specific On Resume Action is not specified when the action is resumed then the action itself will run, provided that the can pause parameter is not false.

Task Take Over when Active: If this parameter is true, then the action can be taken over even when the action is actively executed.

Task Take Over when Inactive: If this parameter is true, then this action is subject to "take over" when this action is not actively executed.

Figure 28 shows a diagram of a sample demo Schedule with some tasks and actions. In the diagram the "On Resume Actions", "Task Take Over" and "Pause" parameters are shown. In addition the special "Start Schedule" and "Stop Schedule" actions are shown. These are part of the Station Controller's schedule orchestration actions. These actions are listed and described in Table 2.

Station Controller Actions	Description
Start Schedule Execution	Action that indicates the single start of a schedule
Stop Schedule Execution	If at any point the station controller needs to execute an end action it will cancel all active actions (if any) and stop the execution of a schedule.
Sync-Any	The action completes instantly and station controller executes all the next actions when any of the previous action of this action is finished.
Sync-All	This action completes instantly and station controller executes all the next actions when ALL of the previous actions of this action is finished.

Table 2: Station Controller Schedule Orchestration Actions

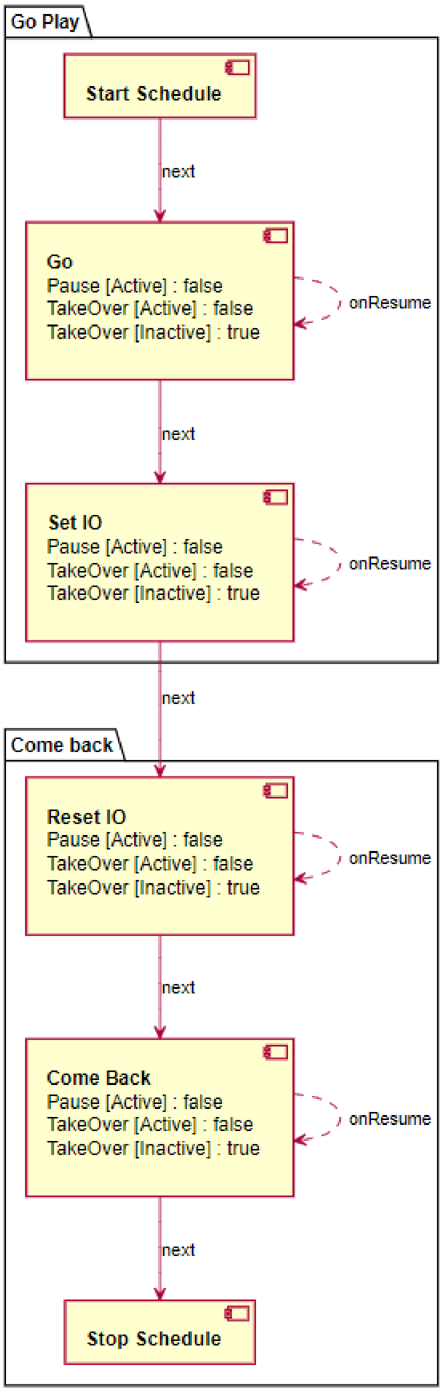


Figure 28: Schedule with Action Parameters depicted

7. CONCLUSIONS

The D5.5 namely the "THOMAS Station Controller – Final Version" is a fully developed prototype software that can execute and coordinate tasks using the THOMAS Network of Services so that the robot's cognition systems can adjust them for real world execution. In addition, it integrates with the HRC Dynamic Task Planner which is the core decision making software that facilitates the cognitive aspects of the line level work re-organization. The final prototype utilizes and augments the outputs of the previous deliverables in particular:

- D5.5 includes the modules that have been developed in D5.2 "Dynamic work reorganization module – Initial prototype"
- D5.5 is developed by augmenting and improving the functionality of D5.3 "THOMAS Service Oriented Network of Resources – Initial Prototype".
- D5.5 follows the design principles that have been established in D5.1 "Methods for dynamic work balancing of human robot collaborative environments - Design".

The final prototype fully utilizes and enhances the components developed in the previous deliverables such as the THOMAS World model in order to access information about the scheduling, task planning and resources modelling of the resources within the system.

Furthermore, the functionality foreseen for the D5.5 has been tested, improved and verified during the development of the THOMAS Use Cases in the premises of LMS. This document accompanies and describes the developed final prototype and provides details about its subcomponents, architecture and features.

The aforementioned functionalities are described in a context based in the automotive use case, which also demonstrated how the manufacturing ontology described in D7.3 is used during the operation of the system.

The latest version of THOMAS Digital world model is documented in this deliverable.

The D5.5 THOMAS Station Controller – Final Version" will be used, connected and integrated with the developments of the other THOMAS work-packages in the demonstrators of THOMAS and be part of the THOMAS Open Production Station that will be the output of WP6. Furthermore, within WP7 "THOMAS Demonstrators and Assessment" the capabilities of the final prototype of the THOMAS Station Controller together with the other THOMAS developments will be used to implement the pilot cases. The pilot cases of the automotive and aeronautics sectors will validate the effectiveness of each result based on the metrics and acceptance criteria that are defined in T1.1.

8. GLOSSARY

BO	Business Object
DAO	Data Access Object
ECU	Electronic Control Unit
MPP	Mobile Product Platform
MRP	Mobile Robot Platform
ROS	Robot Operating System
SONAR	Service Oriented Network Adjacent Resources
SQL	Structured Query Language
UML	Unified Modelling Language
VHD	Virtual Hard Disk
VM	Virtual Machine
VPN	Virtual Private Network
XML	eXtensible markup language
GUI	Graphical User Interface

9. REFERENCES

1. Chryssolouris, G., Manufacturing Systems: Theory and Practice, 2nd Edition, Springer-Verlag, New York, New York, (2006)
2. How the Actor Model Meets the Needs of Modern, Distributed Systems , AKKA Documentation, Lightbend. Available online: https://doc.akka.io/docs/akka/current/guide/actors-intro.html . Last accessed 2019.06.16.
3. Kousi, N., S. Koukas, G. Michalos, S. Makris, and G. Chryssolouris. 2016. “Service Oriented Architecture for Dynamic Scheduling of Mobile Robots for Material Supply.” Procedia CIRP 55: 18–22. doi:10.1016/j.procir.2016.09.014
4. Alexopoulos, K., Koukas, S., Boli, N., & Mourtzis, D. (2018). Architecture and development of an Industrial Internet of Things framework for realizing services in Industrial Product Service Systems. Procedia CIRP, 72, 880–885.