TRACEBOT

D6.2 Initial setup using 2 x Robotic arms with prototypes or similar functional components attached, using ROS middle-ware mock-up

Deliverable 6.2

Deliverable Title	D6.2 Initial setup using 2 x Robotic arms with prototypes or similar functional components attached, using ROS middle-ware mock-up
Deliverable Lead:	ASTECH PROJECTS LIMITED
Related Work Package:	WP6: Integration & Evaluation
Related Task(s):	T1.2: System Architecture T6.1: Demonstration Hardware Integration T6.2: Software Integration
Author(s):	Phil Kewish Ben Gordon
Dissemination Level:	Public
Due Submission Date:	30/11/2022
Actual Submission:	15/12/2022
Project Number	101017089
Instrument:	Research and innovation action
Start Date of Project:	01.01.2021
Duration:	51 months
Abstract:	D6.2 - Initial setup using 2 x Robotic arms with prototypes or similar functional components attached, using ROS middle-ware mock-up. The integration will not be mounted within a framework at this point and will not be within a sterile environment

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089), 2021-2025

Versioning and Contribution History

Version	Date	Modified by	Modification reason
v.01	09 Dec 22	Phil Kewish (AST) Ben Gordon (AST)	First Issue
v.02	13 Dec 22	Anthony Remazeilles (TEC)	Overall review
v.03	13 Dec 22	Anthony Remazeilles (TEC) Ben Gordon (AST)	Ready for internal revision
v.04	14 Dec 22	Torben Cichon (INV) Carl-Helmut Coulon (INV)	Revision Invite#1
v.05	14 Dec 22	Ben Gordon (AST) Anthony Remazeilles (TEC)	Revision & Corrections
v.06	15 Dec 22	Ben Gordon (AST)	Updated Formatting Final version ready for submission

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Table of Contents

Horizon 2020
TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Horizon 2020
TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

1 Executive Summary

This deliverable describes the advancement of the integration of the different development onto a single physical platform. After the first integration effort during the first year, where the global architecture was validated under simulation, we achieved in this second year a first deployment of the components on a real platform, which can perform bi-manual manipulation, combining the perception, control, and the digital twin functionalities.

The first sketches of the physical platform have been refined, to get the final design of the physical platform that has been assembled. A significant work has been conducted to gather each individual partner development into a unique demonstrator. Periodic meetings, in addition to presential integration workshops were arranged to identify the potential issues raised by such integration, to define the adjustments required, and to monitor the progress of the integration effort.

The physical system is now able to showcase the canister insertion into the pump tray, as well as the needle cap removal and the needle insertion into a bottle, which are the two use-cases targeted for the second year. Some adjustments are still needed to improve the demonstrator and complete the component integration, and we expect to conduct them in the following months.

This work is strongly related to Milestone 2 of the project, which goes beyond the integration of the software development. We provide in the document some description of the work achieved according to the Milestone objectives. A YouTube playlist compiling several videos is also provided in section 9.2 to illustrate these advancements. We consider that the main objectives of the Milestone 2 have been achieved, even though some items still need to be further advanced.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

2 Introduction

This document gives an overview of the outcomes of this second year of integration, in the continuity of the activity conducted during the first year, which was reported in D6.1.

We started the year with an integrated system, functional in simulation. An isolated use case was selected, the canister insertion into the pump tray, and we defined the main components and interfaces required to perform the operations, in relation with the expected contribution of the project partners. Most of the components were implemented as mock-ups, with simple emulation of their expected behaviors, so that we could get a functional emulation of the process.

In this second year, a first deployment of functional components onto a physical dual arm platform has been performed. The simulated operation showcased during the first year is now implemented onto a real platform. A bi-manual robotic arm, equipped with regular grippers (the TraceBot dexterous gripper is scheduled for the end of 2023), and with a depth and RGB camera is detecting the canisters, grasping them, and inserting them into the pump tray, while updating the digital twin accordingly. The needle management has been also considered, and a demonstrator showcase the needle grasping, cap removal and insertion into a bottle.

This achievement was obtained thanks to a significant integration effort, involving periodic integration follow-ups and several integration workshops. The initial process (plan defining the operations to be conducted, as a set of robot skills to be executed) defined for the simulation was progressively updated to the real settings, and most of the mock-up components were replaced with real and functional modules providing the expected functionalities.

The next section provides an overview of the work conducted to obtain the expected demonstrator.

The document is organized as follows: next section provides a view on the physical integration, presenting the evolution and final set-up design, as well as the main hardware components embedded into the demonstrator. Section 4 highlights the work conducted on the software integration. Section 5 describes the obtained demonstrator associated to the two use cases targeted for 2022, and section 6 focuses on the Milestone 2 providing evidence on the work conducted to complete it. Then section 7 highlights the deviations with respect to the work plan, and section 8 provides the conclusion. In Annexes, a listing of the figures and the videos gathered in relation to the Milestone and available on YouTube are provided.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

3 Hardware Integration

3.1 TraceBot Design & Build

3.1.1 Early Designs

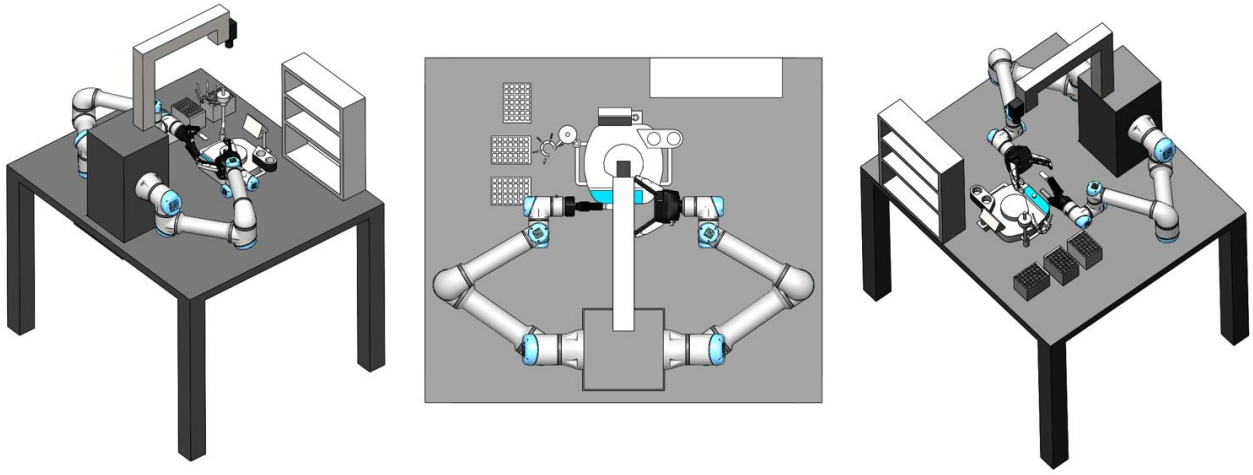

FIGURE 1 - INITIAL CAD VISUALIZATIONS OF THE TRACEBOT WORKSTATION

The initial concept design (Figure 1) consisted of 2x UR5e arms mounted at 90° from the vertical, and a camera mounted directly overhead.

The column that the arms are mounted to is mounted directly to the work surface. This design was never intended to be used for testing, but as a rough draft to kick-start the design process.

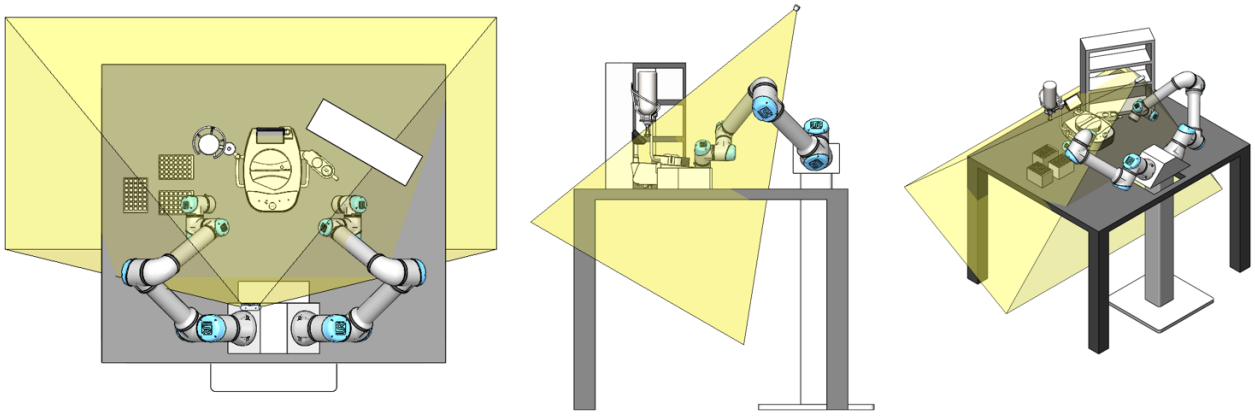

FIGURE 2 - SECONDARY CAD VISUALIZATIONS OF THE TRACEBOT WORKSTATION USED FOR INITIAL TESTING

The design was later modified with arms at an angled 45° and a camera to positioned over the top of the arms and angled towards the worksurface, rather than looking directly over the top of the table (Figure 2). This design is much closer to the final layout of these components. The position of the camera allows for greater coverage of the worksurface, as well as allowing for easier identification of the objects on the table by providing more of a profile view.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

This design was used for most of the early testing in simulation in gazebo, and the general positionings were used as the foundation for the initial design.

3.1.2 Final proto-type design

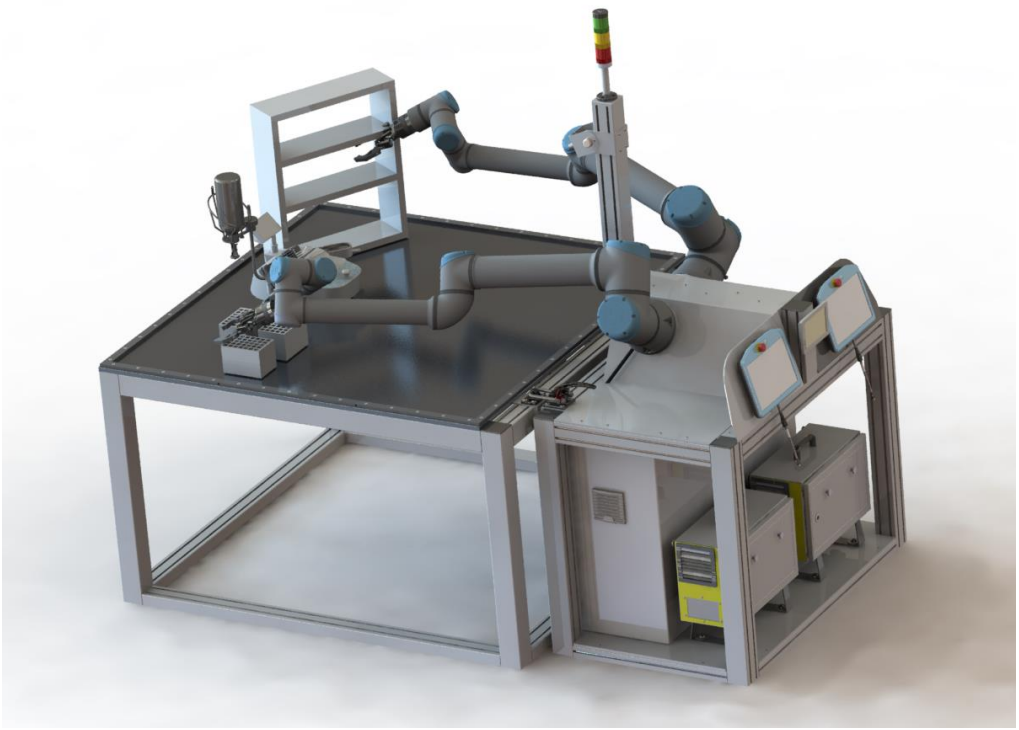

FIGURE 3 - FINAL PROTO-TYPE DESIGN

The final design (Figure 3) consists of 2x UR10e arms mounted to a base at 60° from the vertical. This angle was adjusted slightly from the previous 45° for increased maneuverability.

The base also includes:

- 2x Robot controllers
- An electrical cabinet for housing the PC and electrical components
- Mounting points for the pendants
- A small touch screen monitor for interaction with the system
- A central column for mounting the camera,
- And a Green, Amber, Red beacon atop to display the system state.

The camera bracket is mounted on the central column, which allows the height of the RealSense D435 camera to be positioned between 0.271m and 0.852m above the table.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

3.1.3 First build

FIGURE 4 - INITIAL BUILD

The physical build of the system (Figure 4) is built at the Astech site, with integration meetings (the first of which was in November 22) which will give other partners an opportunity to come in and test on the system. This will be the main system for developing and testing all work packages.

Tecnalia also have a similar setup on their own site, which will be used for development and testing of their robot controllers, as well as test verification to ensure that that tests work on multiple builds of the system.

3.1.3.1 Changes from the design

- Screen size and position (Detailed in 3.2.1)
- Robot Arm orientation (Detailed in 3.2.5)
- Camera Bracket
  - The camera bracket's design was changed to implement a single notch allowing for a range of movement from 0° to 90° from the horizontal, from the initial design of three set notch positions. This provides a greater range of positions to experiment with camera positioning.
  - The initial Camera Bracket was to be mounted directly to the column between the two arms. However due to the way it was manufactured, curvature of the inner corners meant that the flat surface would not sit flush with the column. Therefore a "packer" had to be used, causing the camera to be slightly further back than the initial design. This was updated in the model,
- Locking Handles
  - The camera brackets have been relocated from the top to the bottom of the table/arm mount. This is due to the new orientation of the arms, causing a collision between the handles (when locked) and the arm's shoulders.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

3.2 Key Hardware Specifications

As is detailed in the Vision PC/Control PC later in this document, the architecture was changed from using two individual PCs to just one, therefore any reference to either the Control PC and/or Vision PC will be referencing the same PC.

3.2.1 System Architecture

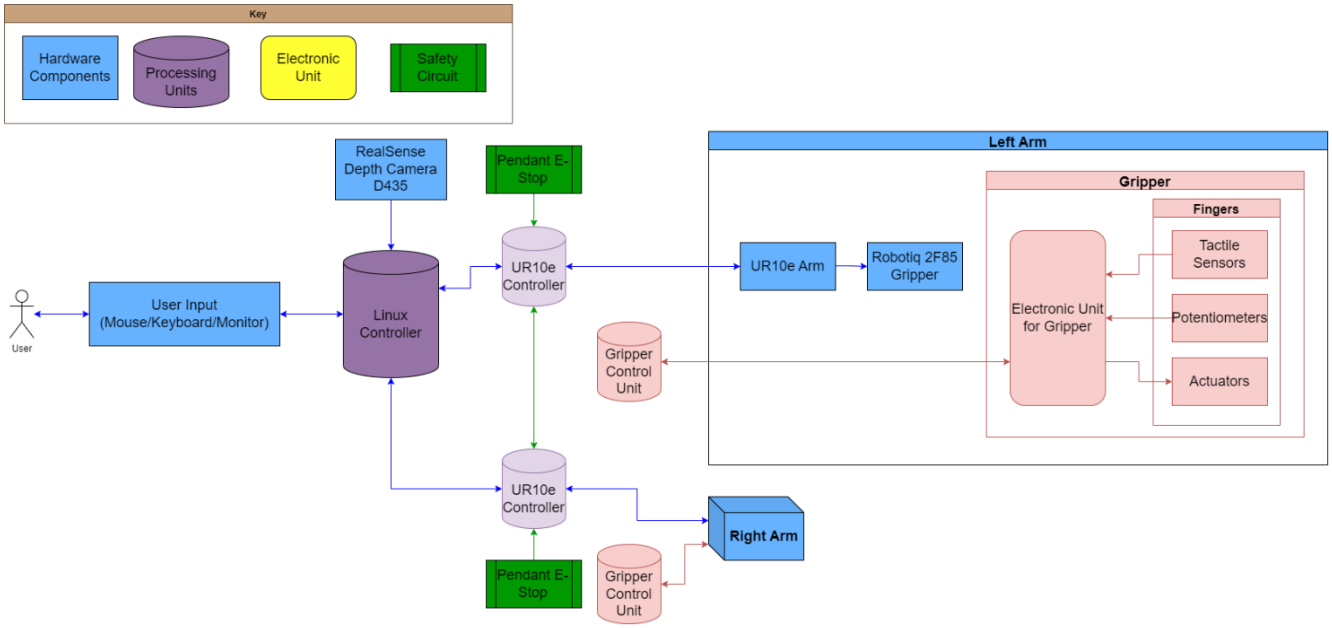

FIGURE 5 - CURRENT HARDWARE ARCHITECTURE

The initial TraceBot build stayed close to the original hardware architecture and was detailed in D1.1.

The main change is the combining of the "Dedicated Digital Twin & Visual Processing Unit" and "Linux Controller" onto a single Control PC. This decision was made due to technical issues with the computer used for the Control PC causing frequent disconnects of the UR arms when an increasing number of packages[1] were being run. A revised system architecture, with the combined controller is shown above (Figure 5). The current Robotiq grippers that we are using have been added to this, with the bespoke gripper highlighted in red as this is not currently implemented on the system. 2 E-Stops have also been added representing the two pendant E-Stops for the robot controllers.

Given the specs of the vision PC, it can sufficiently run everything that was required and has so far had no issues running all the packages.

[1] The term package in ROS refers to a code module. It is usually providing for execution a set of nodes, as individual execution flows can interact through the ROS communication means. In this document, we use the term package to refer to the code itself, as well as the main functionality or component it provides.

Horizon 2020
TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

With the decision for using only a singular PC, all the required packages (from https://gitlab.com/tracebot) have been installed on one computer. However, these have currently been broken into three workspaces:

- Digital Twin
- Vision
- Controller (containing the remainder of the packages)

3.2.2 User Input

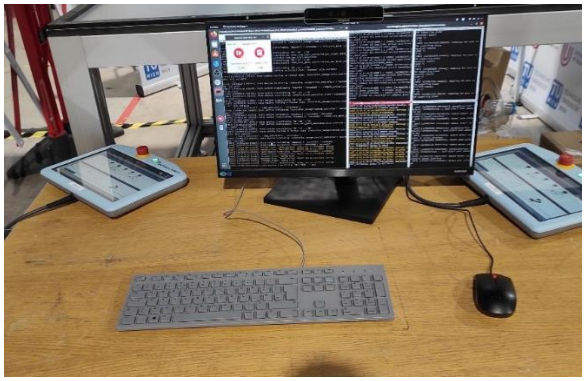

FIGURE 6 - TEMPORARY USER INPUT SETUP

We currently have a monitor connected to the main pc via HDMI and a (USB Type-A to) USB Type-B connection for the USB ports on the monitor. The keyboard and mouse are connected to the monitor (Figure 6).

This is not the final setup however is convenient for current testing and allows for use of a proper workspace (desk and chairs). The initial design showed a monitor mounted on the rear of the robot (Figure 7), however for both safety and ease of use we have decided to have the monitor located on the front of the robot opposite the desk. Here it is out of range of both arms and allows a better view of the workspace when testing. We will also be opting for a larger screen than is on the design, along with a keyboard & mouse mount to make it easier for development.

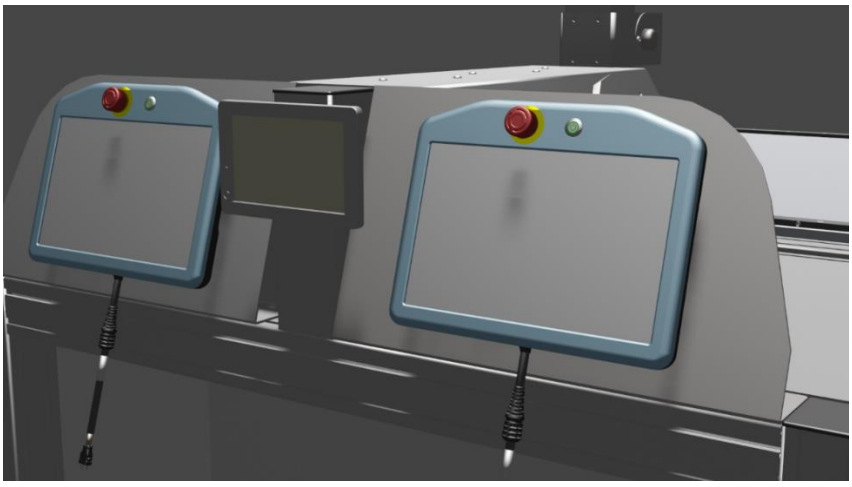

FIGURE 7 - INITIAL SCREEN LOCATION

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

There are also two pendants for the robot controllers. These allow for manual control of each robot for teaching positions, setup etc. The pendants are also currently being used as the primary E-stops. There will later be 4x E-Stops positioned at all four corners of the machine.

3.2.3 Dedicated Digital Twin & Visual Processing Unit / Linux Controller

FIGURE 8 - DT/VISION PC & CONTROL PC

The Digital Twin (DT)/Vision PC (Figure 8) includes a GeForce RTX 2080 Ti graphics card, which is more than capable of handling the Camera drivers, vision based object detection packages *and* Digital Twin that rely on CUDA.

With the decision to combine both Controllers onto a single PC, the 12th Gen Intel Core I7 processor and 32GB ram on board, the computer can run all the packages, and we are yet to run into any issues.

3.2.4 Camera

FIGURE 9 - INTEL REALSENSE CAMERA D435

We have an Intel RealSense Depth Camera D435 (Figure 9) with a range of 0.3m – 3m which is more than sufficient as the maximum distance that would be required for this setup is approximately 1.74m between the camera and the edge of the table. The camera itself is mounted onto a column between the robot arms, this allows the camera to be positioned between 0.271m and 0.852m from the table

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

and can be angled between 0° and -90° from the horizontal. This allows for sufficient range for testing the optimal camera location. The current demo was performed with the camera at max height and pointing down at -60° from the horizontal.

The camera also comes with calibration software to calibrate the accuracy of the camera.

3.2.5 Robot Arms (UR10e Controller & Arm)

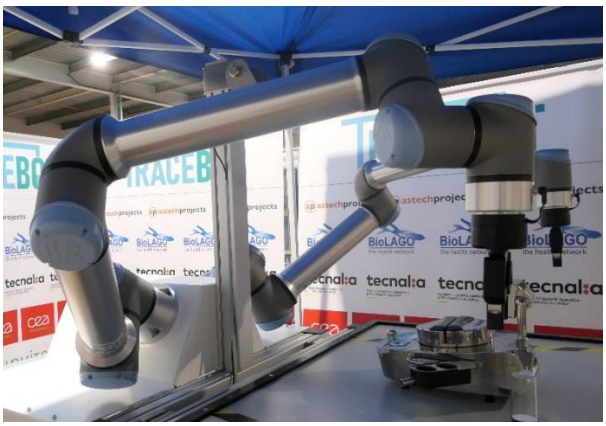

FIGURE 10 - UR10E ARM

We utilized the UR10e Robot arms (Figure 10) which are mounted at 60° from the vertical. Due to the large area of reach (1.3m), safety restrictions have been put in place on the UR controllers to prevent the end effectors from leaving the area of the table. The UR arms are connected to the Control PC via an ethernet switch.

The initial position of the arms had the elbow joint on top (Figure 11), however this caused many issues with trying to position and maneuver the end effector as most of the joints end up positioned below it. This would often cause the arm to lock-up when trying to get into certain positions.

FIGURE 11 - UR10E SHOULDER ON TOP (LEFT) AND UNDERNEATH (RIGHT)

This was changed to position the shoulder bellow the robot instead. This is done via changing the robot position in software (rotating the joint by 180°) rather than physically repositioning the robot. This then allowed for a greater range of motion and led to far less issues for the end effector to get into the desired positions. You can see by the image above (Figure 11) that the setup with the shoulder

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

bellow gives much more room underneath the wrist joints for the end effector to move, as well as reducing the chance of collision.

3.2.6 Grippers (Gripper Control Unit)

Whilst the CEA grippers are still in development, we are using the Robotiq 2F85 gripper for both arms (Figure 12). These are connected directly to the UR arms and controlled via their controllers.

FIGURE 12 - ROBOTIQ 2F85 GRIPPER

3.2.7 Safety Circuit

Only part of the safety circuit as so far been implemented. Currently the controller safety circuits are chained together (Figure 13), allowing either of the two pendants to be used to e-stop the robot arms, so that an e-stop of one causes an e-stop of the other. Next year additional safety stops will be added to the system, adding an E-Stop button to each corner of the workspace.

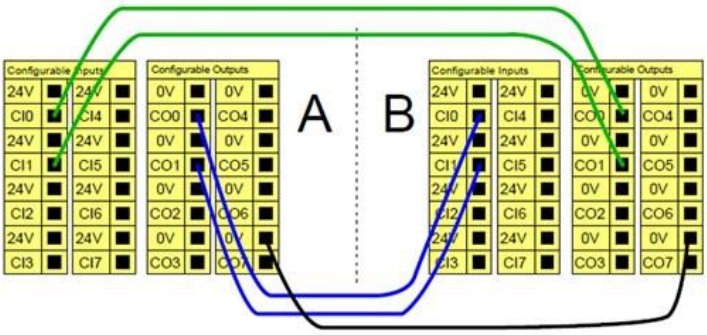

FIGURE 13 - UR ARM SHARED SAFETY [1]

Next year additional safety stops will be added to the system, adding an E-Stop button to each corner of the workspace.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

4 Software Integration

The software architecture (Figure 14) has not varied much from D6.1. The key difference is that the digital twin and vision packages have been relocated to a single computer with the rest of the controller packages, and then broken into individual workspaces. In a few words, at the highest level, the activities of the robot are defined through a yaml description of the different operations to conduct, a named process and contains a set of *skills* to be executed (see section 4.6). This process file is intended to be generated in the final prototype with the cognitive programming interface (see section 4.7). At run time, the Skill execution engine (left on Figure 14) loads the process, and progressively launches the skills described. The skills are strongly connected to the Tracer, as all input and output details are logged through the Tracer to populate the audit trail. Most of the skills themselves rely on and interact with specific ROS nodes (or packages) for launching specific functionalities, such as the camera process node (see section 4.1), the Digital Twin (see section 0), the arm controllers (section 4.3), or the gripper controller (section 4.5).

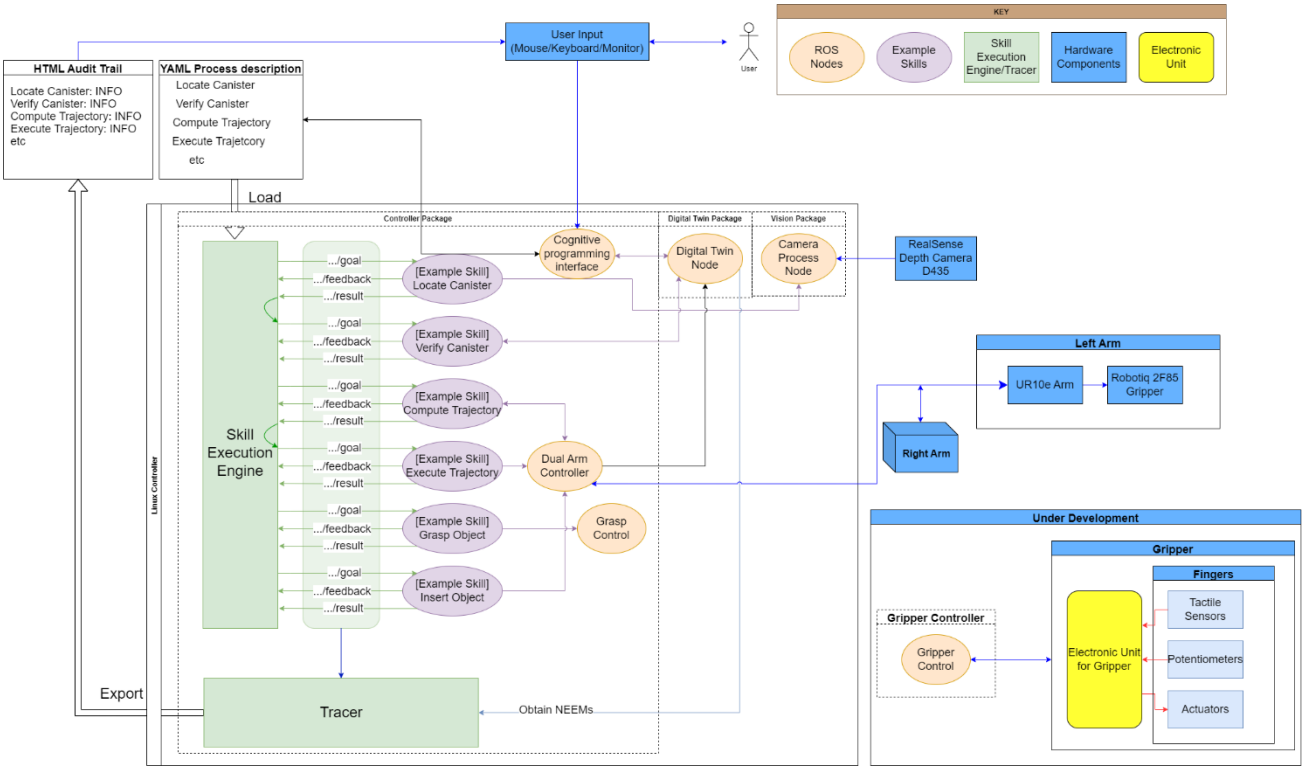

FIGURE 14 - CURRENT SYSTEM ARCHITECTURE

The development of the core components (i.e., ROS nodes) is mainly conducted in the work-packages they are related to. The integration effort consists in (i) validating the development in the partner site, (ii) deploying on the integration site at Astech, and (iii) making sure the combination with the other components is effectively functional. In this document we focus on the two last points.

The integration effort was highly supported by the responsibility distribution which was defined around the first isolated use-case, and that was used during the first-year integration (see Figure 15). When significant updates from the partners were provided on the shared git code repositories, the components were tested at the Astech facility to confirm their integration, and/or highlight some

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

required development. We will highlight here some of the major deployment achievements, which are the perception integration, the arm controller, the digital twin combined with the controller and the perception layer, and the overall process and scene descriptions, as well as the cognitive programming interface. We illustrate these aspects, making relation to the related steps in the first isolated use case when appropriate.

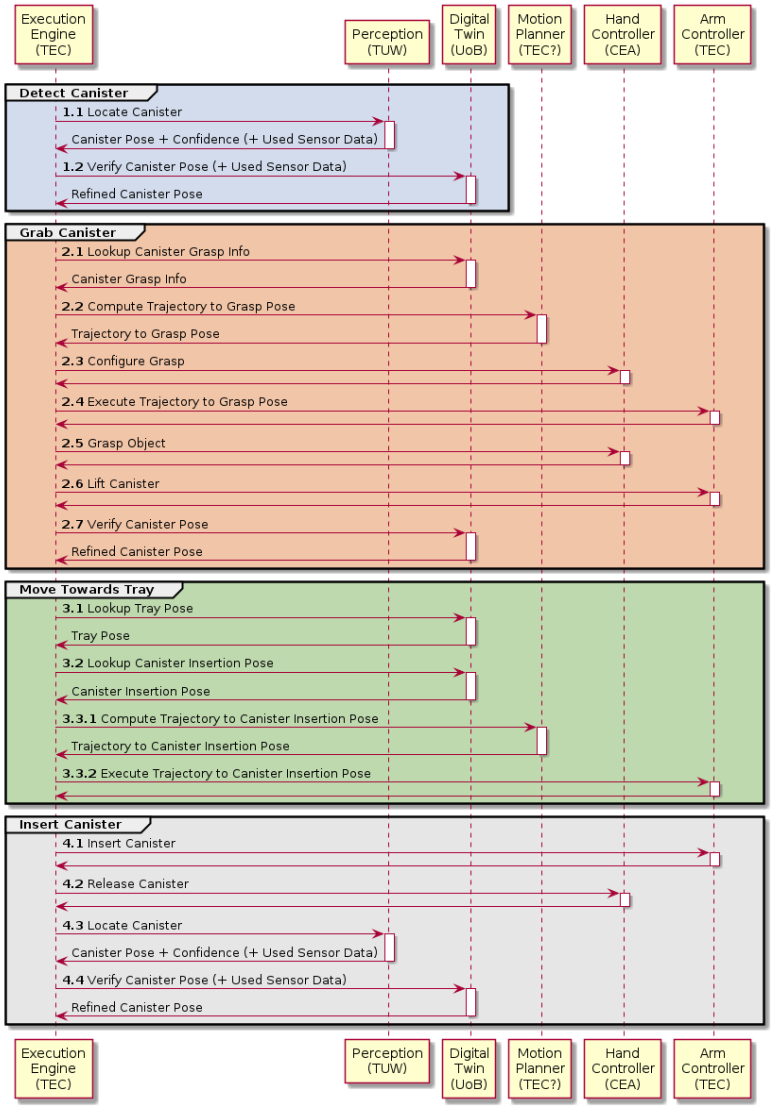

FIGURE 15 - ISOLATED USE CASE "INSERT CANISTER INTO TRAY", AND PARTNER DUTY (FROM D6.1)

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

4.1 Perception integration (steps 1.1, 4.3)

The perception package (Camera Process node in Figure 14) for the location of objects was one of the first packages to be implemented on the system. An early version of the canister perception was implemented for painted canister (Figure 16), and it was replaced during the year with a version able to detect the transparent ones (Figure 17). The component is providing a first estimation of the object pose, which is also refined through a vision-based verification scheme.

The perception component was also updated to handle the needles, as illustrated on Figure 16. We identified that the sterility kits may contain three distinct types of needles, but we are confident the system can detect and distinguish the three of them.

Right now, we do not involve the perception layers to detect the pump, as we can assume it would be static in the environment, and at a known position. Nevertheless, with a flexibility purpose during the integration activity, we detect its position at run-time using a fiducial marker (ArUco) placed onto it.

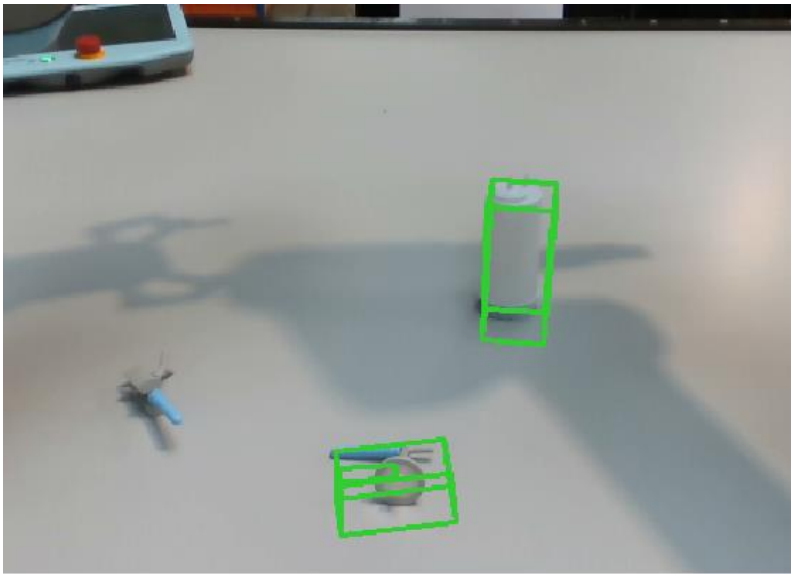

FIGURE 16 - DETECTION OF PAINTED CANISTERS AND NEEDLE

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

The location of these objects is outputted as TF frames (ROS transforms) (Figure 18), which can then be used for the remainder of the process to grasp the objects. Once the object has been located, its pose is then spawned into the Digital Twin (Figure 19), to update the digital representation of the global scene.

FIGURE 17 – REAL CANISTER, TRANSPARENT

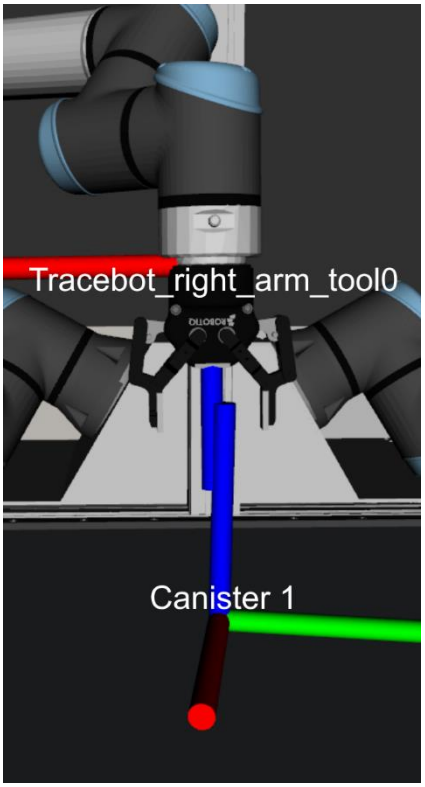

FIGURE 18 – TF FRAME ESTIMATED OF THE CANISTER

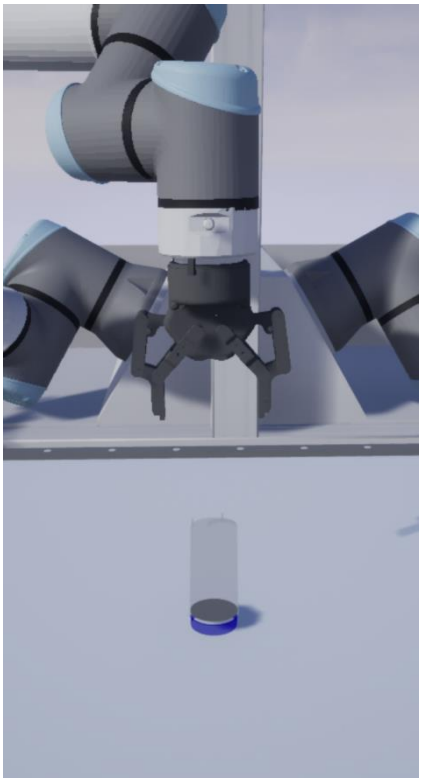

FIGURE 19 – DIGITAL TWIN SPAWNED CANISTER MODEL

4.2 Digital Twin and reasoning system (steps 1.2, 2.1, 2.7, 3.1, 3.2, 4.4)

The required simulation aspects result from the hardware setup as well as the tasks to be executed. To handle changes in the robot, e.g., changes in joint positions through calibration, we made it possible to spawn the robot in the Semantic Digital Twin (Figure 21) (semDT) during runtime, given a robot description. This is particularly relevant as small adjustments of the hardware mounting may be required, and because all partners having a robotic system do not have the same mounting.

To emulate the action of the real robot (Figure 20), we had to replicate the motion of the real robot. For this, we use the published joint state of the real robot. Since the real robot perceives and manipulates the environment. We also created an interface to spawn and move objects for the milestone use cases in the semDT. Because the semDT also acts as a rendered knowledge base it is possible to extract the pose of objects. Some of the tasks in the context of TraceBot require modelling of certain behaviors and object interactions. This included the insertion of the needle into the septum, as deformable or destructible objects are not supported natively by the simulation engine. We investigated and developed constraint-based object interaction models to simulate attachment and insertion operations (Figure 22).

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

FIGURE 20 – REAL BUILD

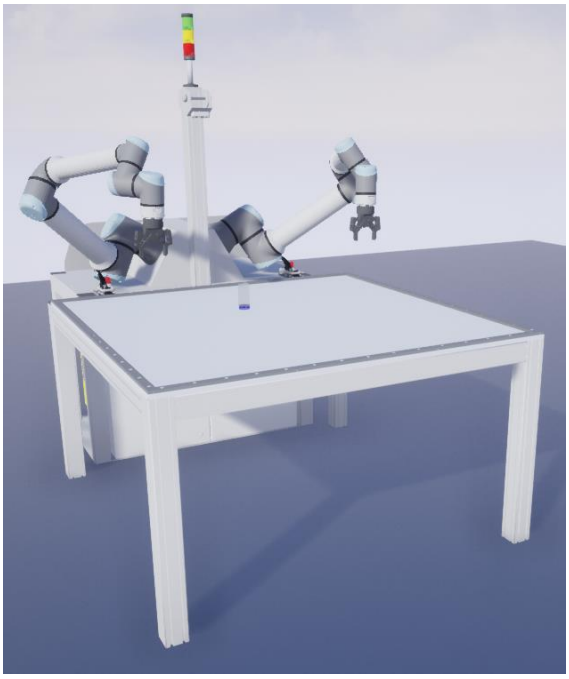

FIGURE 21 – DIGITAL TWIN MODEL

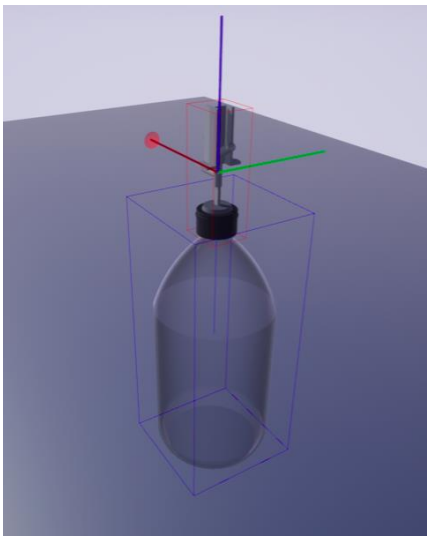

FIGURE 22 - BOTTLE MODEL

Spawning and moving objects also required special attention. This became especially clear during the integration with the real system. Pose estimates based on real camera images are noisy. This can lead to physically impossible situations such as a canister partially being inside the robot or other environmental geometries. To fix this, we introduced physical reasoning. Through this we simulate different states of the object that make physical sense and match the state that best describes the observed scene.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

With the collected information, the Digital Twin can verify the quality of the perception outcomes, and contrast its physical soundness, thanks to the physical simulator involved in the Digital Twin (see the YouTube Video 7 - Video 11 in Section 9.2).

When inserting the Digital Twin into the process, we could confirm the good transmission of the arm motions, as the simulated environment was updated accordingly. At the object grasping phase, we identified that the gripper finger motions were not published in the ROS framework, and proper fixes were developed.

To connect the hybrid knowledge representation and reasoning framework from WP5 with the components from other work packages, we introduced a formal query language interface. The fundamental interactions with the system are either of type TELL or ASK. TELL allows the system to enrich the available belief state and assert new or updated facts. This can be done for example by the perception system asserting newly perceived objects or updating the pose of known objects. ASK allows the system to extract knowledge from the knowledge framework, for example by requesting the expected pose of an object or verifying the success of an action. The power of the language resides in the richness of the ontology. Therefore, an ontology containing the language of TraceBot was created. The basis for the TraceBot ontology is SOMA ontology (Socio-physical Models of Activities). This ontology was enriched by object-related concepts, such as Canisters and Pumps as well as the relation between them, as well as action-related concepts and state-related concepts occurring in the context of the sterility testing use cases. An interface to the perception executive is defined by the perception task language. This language can roughly be divided into detection-based perception tasks, which analyze the camera image directly and imagine-enabled perception tasks that use mental imagination in addition to the real camera image. The description of the interfaces is further detailed in Deliverable 5.2 which was submitted by the end of November 2022.

4.3 Arm Controllers (steps 2.4, 2.6, 3.3.2, 4.1)

The motion of the robotic arms (mentioned as *Dual arm controller* on Figure 14) is handled with components developed in WP3. To validate the control layers independently of the other involved components (in particular the perception layer), the controllers were validated in processes in which the perception modules as presented in section 4.1 were replaced by the detection of fiducial markers (ArUco) placed in the vicinity of the objects of interest. Once validated, the fiducial marker-based process was deployed at Astech, while it was tested with the real perception components. The change of the perception required some light adjustment of the process configuration, which were successfully conducted for the canister insertion process at Astech.

The following pictures illustrate the progressive validation of the controller layers. the controller execution during the canister insertion at Tecnalia (Figure 23, Video 1), with fiducial markers, its deployment at Astech (Video 10), and the final version using the real perception layer mentioned in section 4.1 (see Video 11, Figure 24).

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

FIGURE 23 - CANISTER INSERTION CONTROL MODULE AT TECNALIA, USING FIDUCIAL MARKERS

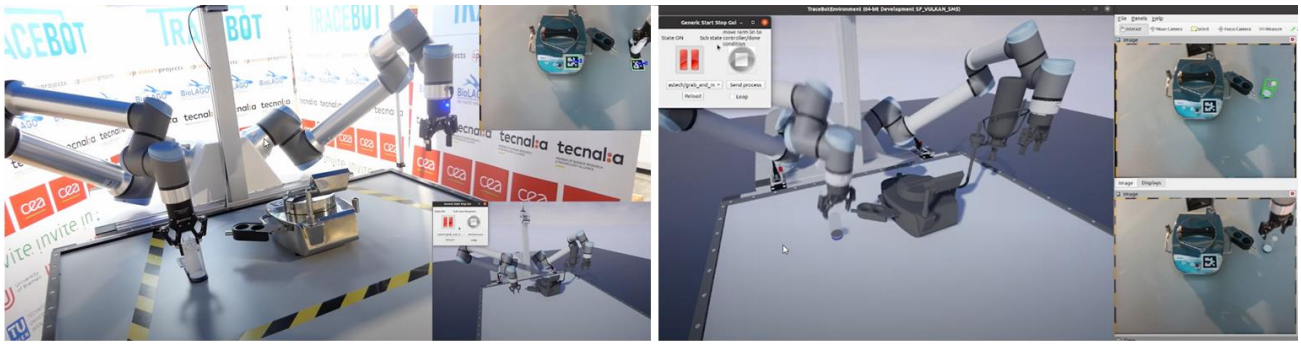

FIGURE 24 - CANISTER INSERTION MODULE AT ASTECH. LEFT: FIDUCIAL MARKERS ARE USED TO LOCATE THE CANISTER. RIGHT: PERCEPTION MODULE IS USED TO ESTIMATE THE CANISTER POSE

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Similar strategy was used for the needle management, development, and deployment. The control layers were first validated at Tecnalia using a marker solution (Video 2, Figure 25) before being deployed at Astech (Video 9, Figure 25). The following figures illustrate it. To simplify the needle grasping, the needle is placed on a convenient support. Also, during the integration, we identified that the needle used for the manipulation was not the same one as initially used for training the perception layer. Both the manipulation and the vision component have been updated to handle the other types of needles, and we are now working on their integration to combine these updates to finalize soon this integration.

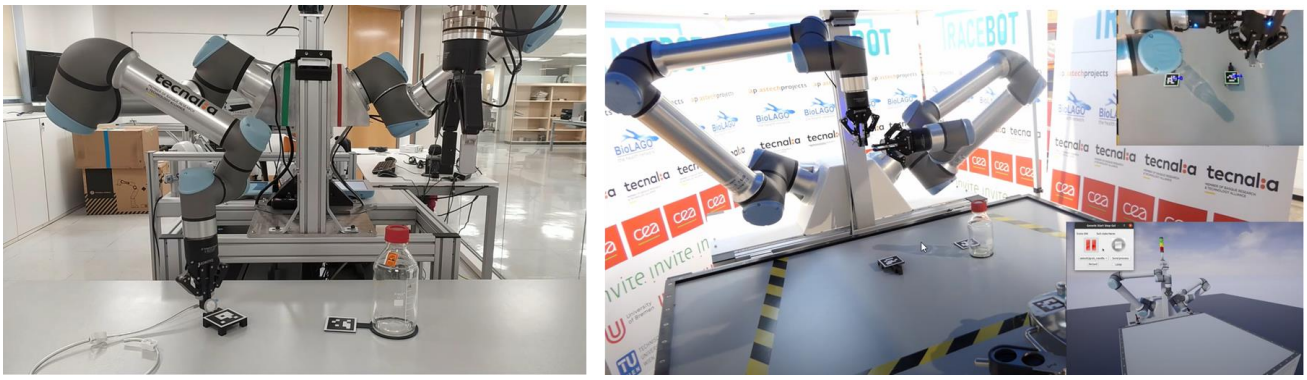

FIGURE 25 - NEEDLE MANAGEMENT PROCESS, AT TECNALIA (LEFT) AND DEPLOYED AT ASTECH (RIGHT)

4.4 Motion Planner (steps 2.2, 3.3.1)

For the first integration milestone in simulation, the MoveIt component (displayed on Figure 14) was used to plan the motion of the robotic arms, in particular for the point-to-point displacement. If this presents the advantage to cope with the possible obstacles present in the environment (if properly detected), it requires a proper management and tuning of the planner, as the definition of the arm motion is then totally delegated to it for such motions.

We decided to switch to a simpler mechanism for this first integration, which is based on a simple but efficient linear interpolation, which is handled through the same low-level control scheme developed in WP3 for more complex motions. We consider this option as viable at short and long term, as the environment is not too cultured, and we did not see any issues during our experimentations. Such controller is currently used for all the movements involved in the needle use case; target points being updated through the perception components. It is also used for the canister manipulation, expect for the final insertion of the canister, which involves a specific controller for generating a compliant motion of the robotic arm during the insertion process.

4.5 Hand Controller (steps 2.3, 2.5, 4.2)

The controller of the dexterous gripper (item *Grasp control* on Figure 14) is not yet implemented on the build, as the dexterous TraceBot gripper is expected to be delivered on the latter half of 2023. Therefore, Robotiq 2F85 grippers are being used in the current demonstrations (see section 3.2.6), and a ROS driver implemented by Tecnalia is used for controlling the gripper opening and closure.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

4.6 Overall process and scene description

The overall control of the robotic system is still based on the skill framework, in which all robot behaviors are encapsulated into skills, and a hierarchical process is defined as the successive set of skills that need to be executed to conduct the desired operation (items *Skill Execution engine* and *Yaml process* on Figure 14).

The initial process prepared for the simulated system during the first year has been updated to match with the updated specification and interfaces of the different modules involved in the global process. Another process was also prepared to cover the second use case, i.e., the needle management.

A particular care has been also devoted to the alignment of the scene description (mainly the relative robot and camera positions into the environment). Indeed, the first version was manually created to get a first functional simulation environment. We had to adjust that description to match with the calibration outcomes of the system assembled at Astech. We also organized the scene description so that it can be more easily adapted to different system configurations, without having to change the content of the process. This way, for instance, the same process is used on the setup used at Astech and on the setup available at Tecnalia. Only the configuration setting of the environment is changed, not the process. The fact that we can adjust at runtime the scene loaded by the Digital Twin is also an option that provides more versatility for adjustments in the future developments.

4.7 Cognitive Programming interface

The cognitive programming interface (item *Cognitive Programming Interface* on Figure 14) developed in WP3 was not intended to be integrated during the second year, but we started already to deploy it on the demonstration site, to identify any integration issue. The cognitive interface has been encapsulated into ROS component to ease their deployment. It covers the following items (see Figure 26):

- A knowledge base: a semantic description of both the robot's environment and the skills (including parameters, effects and preconditions, item *Ontology file* on Figure 26)
- A module to exploit dynamically the knowledge base (layer Dynamic knowledge exploitation on Figure 26)
- A user interface to hide the complexity and allow the user to build the skill sequence, with the support of the interface (user interface on Figure 26, see Video4 in section 9.2).

We created a knowledge base, based on an ontological model to define the objects, their affordances and the skills involved in the TraceBot processes. One thing is the way we encoded the effects of the skills into the ontology: we encoded them in strings data as instructions to be read by the exploitation module to modify the knowledge base according to the skill selected by the user.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

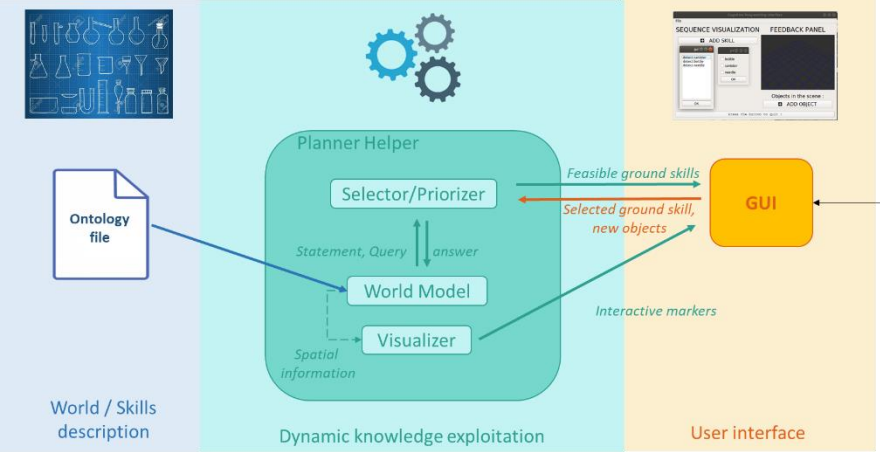

FIGURE 26 - COGNITIVE PROGRAMMING INTERFACE

We created a knowledge base, based on an ontological model to define the objects, their affordances and the skills involved in the TraceBot processes. One thing is the way we encoded the effects of the skills into the ontology: we encoded them in strings data as instructions to be read by the exploitation module to modify the knowledge base according to the skill selected by the user.

Then we implemented all necessary components of the exploitation module (middle area of Figure 26):

- The world model module: it stores the state of the robot's environment directly exploiting the ontology (knowledge base) dynamically. Consequently, it can update the state of the world according to the skill selected by the user.
- The planner helper module: it selects all the feasible skills among the full set of skills to form a list. Then it sorts this list, putting at the top of the list the skill that are most likely to be used. The priority order is made of the two following assumptions: (i) the skills that are feasible a step k but not at step k-1 have the highest priority (situation set by the user to a skill) (ii) the skills that involved already used objects have high priority (user may perform several skill including the same object in a raw).
- Finally, to reduce the size of the list, one can use a filter to only keep the skills fulfilling some constraints.

In parallel, we implemented the Graphical User Interface (Figure 27) that allow the user to:

- Load initial scene

- Add a Feasible skill to the skill sequence

- Add objects to the scene

- Save sequence on YAML file to be executed by the executive part

- Filter the list of feasible skills defining some constraints (based on objects)

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

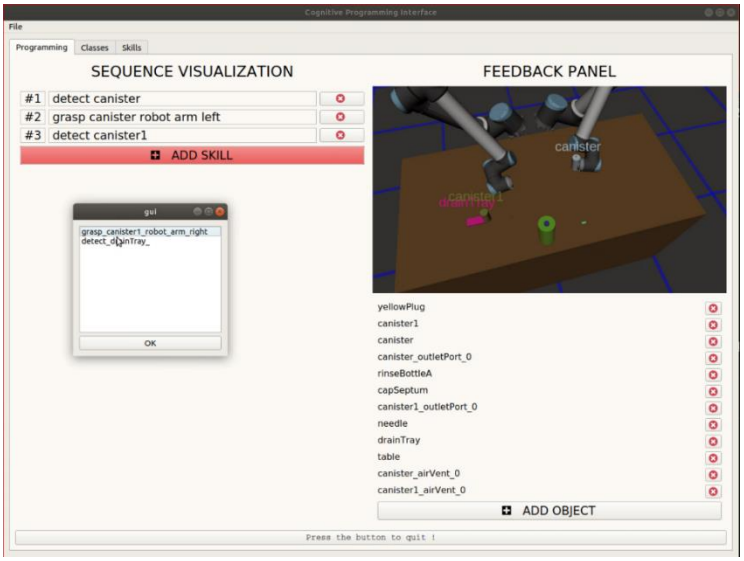

FIGURE 27 - SEQUENCE VISUALIZATION EXAMPLE

A schema to sum up the previous information is provided below (Figure 28):

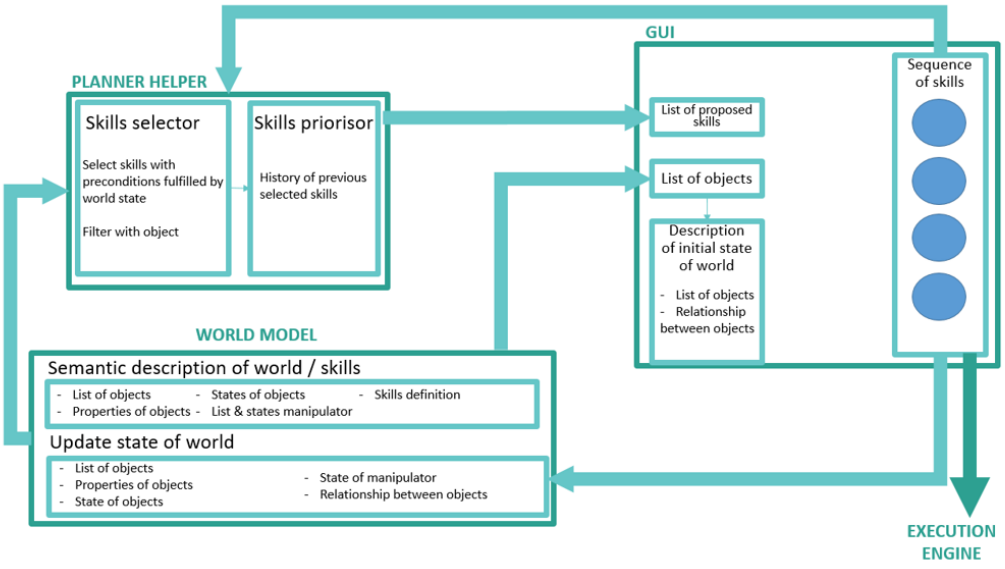

FIGURE 28 - SKILLS SCHEMA

During the next period, we will continue working on this component, to cover the additional use cases covered, and demonstrate the capability to generate a process that can be executed by the robot.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

5 Use case demonstrator states

The canister insertion into the pump tray is the most advanced demonstrator, as it is the one we started with. The transparent canister is detected by the perception layer, and the Digital Twin is getting updated accordingly. The canister is then grasped, brought on the top of the pump tray, and then correctly inserted into it (Video 11). Thanks to the integration effort, we could combine the latest perception layer able to estimate the pose of transparent canisters with the manipulations parts to get a combined demonstration. The communication with the Digital Twin is also enabled and the DT is updated with the perception outcomes and the motion of the robotic arm. As already commented, additional work is required to handle robustly in the digital model of the world special cases when the robotic system contacts objects of the scene (like when grasping the canister).

The needle management through the robotic system, second use case targeted for this period, has been also deployed onto the Astech platform. Nevertheless, we could not yet complete the integration with the perception modules to get a joint demonstrator. The current robotic implementation relies on the usage of fiducial markers (Video 9), and we are currently working on the upgraded version including the perception of the needle pose, component that is already working in standalone version. As of today, the needle management demonstrator covers the needle location and grasping, cap removal and needle insertion into a bottle, as demonstrated in the videos associated to this document.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

6 Link to Milestone MS2

Deliverable D6.2 is related with the second Milestone of the Project, entitled *"First demonstration: proof of concept of integrated sterility test case."*

Several videos illustrating the achievements are available in Annex 9.2 linked to the Milestone. We also detail in the following text our position with respect the means of verification.

[WP2] Implementation of the newly designed modular finger embedding tactile sensing

The present Video 3 - TraceBot WP2 Milestone2 illustrates the capabilities of the robotic finger for the future gripper of the TraceBot project. This video reflects three main scientific and technical contributions, which are summarized in the following.

Modular finger description

This section presents the design of the mechatronic multi-phalanges finger through an animated CAD model, which highlights its multimodal capabilities.

The interphalangeal joints are controlled by a back drivable transmission, which allows us to estimate the applied external forces at the phalanges.

The finger is underactuated, exhibiting three electric motors to control four joints.

The motion of the intermediate phalanx produces the displacement of the distal phalanx thanks to coupling pulleys. The use of a passive spring in the distal phalanx enables to generate object shape adaptation.

Moreover, the multimodal sensitivity apparatus of the finger combines the following elements:

- Three incremental encoders located at the shaft of all the electric motors.
- An absolute encoder is placed at the passive joint of the distal phalanx.
- A multi-layered tactile sensor pad per phalange is implemented: a piezoresistive sensor layer will be dedicated to compute spatial data (e.g., position of the contact point, estimation of the pressure distribution), while a piezoelectric sensor layer will be used for detecting critical event detection (based on the treatment of high-frequency signals).

Finger motion

Short video clips illustrate some available degrees-of-freedom of the finger.

The rotation around the orthogonal axis to the flexion axes of the phalanges produces the out-of-plane orientation of the three phalanges with a range of motion from 0° to 90°.

The flexion motions of the three phalanges relies on a coupling phenomenon in the motor-to-joint mechanical transmission due to cable routing.

The antagonistic movement of the distal phalanx is due to the effect of the torsion spring.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Tactile sensing capabilities

The visualization of the data provided by the piezoresistive sensitive layer are made possible thanks to an 8x8-pixel colourmap image that highlights the estimation of the location of the contact point and the intensity of the pressure distribution. In addition, the data provided by the Piezo sensors are displayed on a graph, which helps to detect high frequencies content that has been produced during critical events (e.g., clicks or impacts). Demonstration of these sensors are shown in Video 5 - 2021-CEA-PiezoR and Video 6 - 2022-CEA-PiezoE

[WP3] Manipulation skill framework fully implemented

The framework orchestrating the activity of the robotic system through the skill concept is used in the integrated system as expected. A first integration was conducted for the first Milestone where all the mock-up components were already using the skill concept. The framework has been extended and adjusted until MS2 to cope with the requirements generated by TraceBot. In particular, (i) the concept of Traceable Skill has been introduced to provide a more systematic tracing of the robot operations, (ii) specific interaction with the episodic memory has been prepared to synchronize the launch of the robot actions with the recording of the traces, (iii) specific meta-information has been added to the skill description to create the relationship with their related description within the ontology. The implementation of the skill framework is thus considered as completed, although we acknowledge that slight adjustments may be required to cope with the upcoming needs of the other work package in the next years.

[WP3] Unimanual manipulation

The canister insertion process demonstrated in simulation for the first milestone has been developed and demonstrated onto the Tecnalia setup, as demonstrated in Video 1 - 2022 07 TraceBot Tecnalia Canister. To validate the manipulation process as an isolated process, we used a simplified perception layer, based on fiducial markers. This demonstration has been also deployed at the demonstration facility in Astech as seen in Video 10 - TraceBot WP6: Canister Insertion Demo (ArUco Markers).

The canister insertion use-case involves a grasping process, and a specific controller to control the insertion of the canister into the tray, involving a rotation around a pivot point while considering the forces perceived by the arm force sensor.

We also proposed a simpler control scheme for handling the needle capture, cap removal and insertion into the septum of a bottle. This process involves vision for locating the different component and adjust the robot motions according to the location of the different items. The current videos illustrating this, recorded at TECN and AST, focus on the control validation, and rely on fiducial markers for the perception layer. The management of the system is based on the skill framework extended in TraceBot as previously mentioned.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

[WP3] Cognitive programming interface mockup implemented

Programming of the canister insertion process is demonstrated in the implementation of the mockup of the cognitive programming interface. This latter relies on the implementation of a semantic representation of both the robot's environment (objects and manipulators in the scene) and the skills present in the skill library (parameters, preconditions, effects). It also relies on the reasoning part that computes which skills are feasible according to the state of the environment.

Through this interface, the user can add feasible skills to the sequence of skills to be executed later. To reduce the number of proposed feasible skills, the filter module has been implemented. The filter allows the user to select the objects of interest that are in the scene to only display the feasible skills involving these objects. The priorizer module has also been implemented to put on the top of the list the skills most likely to be used (depending on the previous selected skills). Additionally, the user can save an existing semantic scene or load a semantic scene for future use. He can also add objects already defined in the knowledge base to the current scene.

A feedback panel shows semantic information to the user so that he can forecast the effects of its program on the robot's environment.

Video 4 - TraceBot WP3 CEA Insert canisters shows someone programming the canister insertion through the mockup of the cognitive programming interface. It shows the use of the filter and the priorizer on the list of feasible skills displayed to the user.

[WP4] Initial tactile, visual, and functional verification tasks integrated into traceability framework

The piezo electric and piezo resistive components have been tested and used to perform two verification tasks:

1 – Contact detection:

The modular sensitive phalanx uses the piezoresistive sensor to measure the pressure distribution and with a simple threshold on the pressure integral we can detect the exact contact moment with the object. The sensor allows us to visualize the relative pressure distribution on the phalanx as well as the form of the contact surface. We also detect the exact moment of loss of contact using the same procedure.

2 – Detect the clamp click

In this test, the system is required to detect the clamp closing using its piezoelectric capabilities. The piezo electric sensor has a high bandwidth of 10KHz allowing it to perceive vibrations in a wide range of frequencies. The signal is first high band filtered and the vibration spectrogram is computed in Realtime. A pattern recognition is applied to the signal to extract the specific signature of the click. The click event has a specific pattern and is clearly distinguishable from the noise. The pattern recognition assures that this is actually a real click and not just a collision or another form of vibration. The pattern recognition parameters will be recomputed when the final sensor coating and the mounting conditions are chosen as this can have an impact on the perceived click signature.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

3– Visual verification

In the first version of the visual verification component, we proposed and integrated a method to verify non-transparent object poses. We combine hypotheses verification with object pose refinement guided by physics simulation. This allows the physical plausibility of individual object pose estimates and the stability of the estimated scene to be considered in a unified optimization. The proposed method can adapt to scenes of multiple objects and efficiently focuses on refining the most promising object poses in multi-hypotheses scenarios. Verification is based on a rendering-based score measuring the fit between the data perceived and the pose estimated. This verification drives a pose optimization procedure that uses physics simulation steps and ICP-based refinement steps, such that they prevent each other from diverging.

Regarding visual verification for transparent objects, we are developing methods based on inverse rendering, which is physically based renderers implemented using only differentiable steps, enabling a backpropagation step to optimize for a variety of parameters in the scene. This approach is particularly well-suited to the project as it can take advantage of all the knowledge of the scene available. The first step consists in the object pose refinement process. This optimization is directed by the fit between the estimated silhouette and edges of the transparent objects in the scene and those of the known object model in the estimated pose. The process is also supervised using a collision-based loss (computed using the environment signed distance function) guaranteeing physically plausible methods. The convergence of this optimization procedure gives us a mean of verification of the estimated pose. Further experiments are ongoing to use a similar method to also estimate the position of the tube, attached to the canister, and the fill level of transparent containers, opening the door to more functional verification steps through visual means.

4– Functional verification

Functional verification is a higher-level verification modality that should ultimately decide whether the actions of the robot were successful, whether it is or was even correct to perform an action within a context and whether the equipment are working properly. Up to this milestone, we formalized the problem of functional verification, and provided the fundamental software and data models (i.e., part of the TST developments in WP5) for solving the problem. Furthermore, a query language (see D5.2), encapsulated into ROS actions, was also provided to the rest of the TraceBot's robotic system as an interface for submitting functional verification tasks to the reasoning system (i.e., part of the TST). Finally, the feasibility and the success verification of the grasp-canister action was illustrated.

Note that functional verification can be reduced to making the robot understand its actions and the desired effects of these actions by eventually adding the so-called physical verification actions (e.g., pushing the canister to check if it is well inserted). For this reason, we firstly digested the detailed description and specification of the TraceBot use case elaborated in the consortium into a formal ontology (extension of SOMA - Socio-physical models of activities) that establishes the fundamental truths about objects and tasks in the world. Particularly important regarding such an ontology with respect to functional verification was the formalization of action pre- and post-conditions, which represent the necessary conditions and effects of a given action. Then, we grounded the action skill library from WP3 into the ontology in such a way that any action performed by the robot has a meaning in the ontology, which then allows to tell or ask the TST information about such an action.

This being said, and given an ongoing robot process, we enrich the actual state of the process also known as process context with information coming from different verification sources (e.g., tactile, visual, semantic digital twin). Given that the simulation-based reasoning methods, namely the

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

physical and imagistic reasoning, enabled by the semantic digital twin are of particular importance in supporting this process understanding, we could also ensure the required simulation aspects in terms of semantic models and interfaces (See WP5 description). To enrich the knowledge base with this contextual information, a subset of the query language mentioned above, called TELL-queries, was provided, and presented in D5.2. Finally, this enriched process context is combined with the generic action post- and pre-condition rules from the world ontology to decide about an action feasibility, action success or equipment functioning. Note however that the semantic models for the functional verification in the TST only evolves with the time as an increasing number of TraceBot scenarios are tackled.

[WP4] Traceability framework tested on first use case

The Traceability framework has been designed with the purpose of creating a level of self-awareness to the robot, in that every action performed can be checked against specifications and expectations. This is achieved through the use of Traceable actions, which transmit all relevant information to a central tracer component, and the implementation of "checking actions" in the process defined, that explicitly compare the system provided values to the expected ones. This system is therefore tightly integrated with the skill execution engine, but also with the Knowledge Base component, as relevant information is provided to it to ensure the correct creation of Narrative-Enabled Episodic Memories (NEEM). The NEEM let a human observe all the system knowledge after the fact as a justification for the reported result of checking actions, giving transparency traceability process. While the set of checking actions will continue to grow, all components enabling the system described above have been tested and used in all the process run on the robot.

[WP5] Software interfaces for first demonstrator defined

To connect the hybrid knowledge representation and reasoning framework from WP5 with the components from other work packages, we introduced a formal query language interface.

The fundamental interactions with the system are either of type TELL or ASK. TELL allows the system to enrich the available belief state and assert new or updated facts. This can be done for example by the perception system asserting newly perceived objects or updating the pose of known objects. ASK allows the system to extract knowledge from the knowledge framework, for example by requesting the expected pose of an object or verifying the success of an action.

The power of the language resides in the richness of the ontology. Therefore, an ontology containing the language of TraceBot was created. The basis for the TraceBot ontology is SOMA ontology (Socio-physical Models of Activities). This ontology was enriched by object-related concepts, such as Canisters and Pumps as well as the relation between them, as well as action-related concepts and state-related concepts occurring in the context of the sterility testing use cases.

An interface to the perception executive is defined by the perception task language. This language can roughly be divided into detection-based perception tasks, which analyze the camera image directly and imagination-enabled perception tasks that use mental imagination in addition to the real camera image.

The description of the interfaces is further detailed in Deliverable 5.2 which was submitted by the end of November 2022.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

[WP5] Required simulation aspects are available

The required simulation aspects result from the hardware setup as well as the tasks to be executed. To handle changes in the robot, e.g., changes in joint positions through calibration, we made it possible to spawn the robot in the Semantic Digital Twin (semDT) during runtime, given a robot description. To emulate the action of the real robot, we had to replicate the motion of the real robot. For this, we use the published joint state of the real robot. Since the real robot perceives and manipulates the environment, we created an interface to spawn and move objects for the milestone use cases in the semDT. Because the semDT also acts as a rendered knowledge base it is possible to extract the pose of objects. Some of the tasks in the context of TraceBot require modelling of certain behaviors and object interactions. This included the insertion of the needle into the septum, as deformable or destructible objects are not supported natively by the simulation engine. We investigated and developed constraint-based object interaction models to simulate attachment and insertion operations.

Spawning and moving objects also required special attention. This became especially clear during the integration with the real system. Pose estimates based on real camera images are noisy. This can lead to physically impossible situations such as a canister partially being inside the robot or other environmental geometries. To fix this, we introduced physical reasoning. Through this we simulate different states of the object that make physical sense and match the state that best describes the observed scene. A report about the simulation aspects can be found in Deliverable 5.2 which was submitted by the end of November 2022.

[WP6] Initial hardware setup using 2 x Robotic arms with prototypes attached for demonstration at partner site (D6.2)

This deliverable, together with the provided videos, demonstrates the completion of that point.

[WP6] Sterility testing Pump Software integration and testing

Through our involvement in the BioSash initiative, subproject of funded of DIH-HERO leaded by BioLago, we got into contact with Merck to create a software bridge in between their pumping software and the SILA middleware, which is being promoted to interact with laboratory equipment. Unfortunately, we could not yet make this collaboration happen, and we will continue interacting with the pump manufacturer to advance in that direction. Note this does not affect the current development, as we are focusing more on the development and integration of the other perception manipulation and reasoning components.

[WP6] Verification testing of Integration

The final sprint to get an integrated system handling the targeted use cases did not permit us to correctly address this item. We envision to do so at the very beginning of 2023, in parallel to the on-going interaction. The objectives of these verifications will be to measure the robustness of the system by executing multiple times the required operations, and by verifying that the system is able to at least detect on its own, through its embedded verification means, any deviation to the expected nominal situation. These experiments will be described in the next reporting period document.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

[WP7] First demonstration at annual external conference

In the last two years there has been a general uncertainty in the planning and implementation of events due to the pandemic situation and the related practicality of people able attend physical meetings. The TraceBot consortium has decided to postpone the conference for a couple of months to a season where we are able to present our results to a broad public for a couple of reasons: First by the end of spring/ beginning of summer, there will (hopefully) neither be travel or meeting restrictions and the chance of speakers or key persons cancelling due to illness is less likely. Second, the integrated system is more developed, and a presentation can be done in more detail. Third the consortium has more time to prepare the conference and will potentially be able to establish a lasting format. The date for the conference is projected for mid-May and the concept is already established. More details about the state of the preparations and the steps taken in planning and marketing for this event will be shown in the technical report at the beginning of 2023.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

7 Deviations from the workplan

The integration effort to reach the first physical dual arm has been intense in the last period of the year, and we succeeded to bring together all the components required to get the first manipulations of the TraceBot objects, in the context of the use cases Canister insertion and Needle manipulation. As already mentioned, the usage of ROS eased the parallel development of building blocks at each partner site, as well as the deployment of these software layers at the demonstration site in Astech.

The preparation of the physical system took more time than expected, as we suffered supply chain difficulties. This affected the integration at the demonstration site, and we tried to mitigate this by launching integration efforts at some partner sites, especially during our first integration workshop at Tecnalia.

Nevertheless, some elements could not be completely validated in the expected time. The needle management use case is not yet completely integrated, as the integrated manipulation showcased is still using fiducial markers. The perception layer developed in WP3 can estimate the needle pose, but we need some additional work to adjust the process to use such layer in place of the fiducial markers. We assume this will be addressed in a close future, as no major technical issue has been identified to do so.

The current demonstrator does not highlight the audit trail content, even though the trace information is being generated and stored. Additional work is also needed to make that information visible in the demonstrator. Here again, no major technical issue is identified.

The delay in the integration process did not permit us to test the integrated system at the demonstration site, to report on the robustness of the system, on the capability to detect failures. Now that the integrated system is ready, we plan to conduct these acceptance tests at the beginning of 2023, and to report them in the next report. This will not block the partners to continue their development as scheduled for the next milestone.

Finally, the present document delivery has been delayed of two weeks, consequences of the integration delay, to collect all the information, and to prepare some videos illustrating the integrated system. This delay is not affecting the next actions planned in the project.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

8 Conclusion

This deliverable presents information on the advancement of the integration process after the completion of the second milestone. The main outcome is the deployment at the integration site at Astech of the components enabling to demonstrate the manipulation of the canister and the needle with a bi-manual system, involving the perception layers for detecting these objects, as well as the bridge with the Digital Twin to maintain an updated digital view of the overall scene.

The document highlighted the work conducted to reach the final design of the physical setup, as well as the main physical items integrated on the demonstrator. On the software side, the main integration aspects have been also inserted.

The *use case canister insertion in the pump tray* is the most mature demonstrator, as it fully integrates the main software items previously mentioned. Additional work is needed on the needle management use case, as the deployed system is as of today relying on fiducial markers. The needle detection with the perception layer is functional at the deployment site, but some adjustments of the integrated process are still needed to finalize this part. This should be achieved in the following weeks.

The integration process is one of the key advancements expected for the completion of the second Milestone of TraceBot. We also provided information to describe the other items related to that milestone. If some points still need to be further advanced, we consider that globally this milestone is reached. A YouTube playlist collecting several videos is also provided to complement the means of verification.

The achievement of this integration has been facilitated by the establishment of a periodic integration meeting during 2022, initially twice per month, and then after summer 2022 on a weekly basis. During these meetings, all partners provided an update of their individual developments, while considering the additional adjustments required to combine the different contribution into a unique deployment site. Additionally, two physical integration workshops were performed, one at Tecnalia in July 2022, and another one at the deployment site at Astech in November 2022.

In 2023, we envision maintaining the periodic integration meetings, to continue improving the generated demonstrator, to prepare the integration of the additional developments of the partners, including the dexterous gripper, and to manage the next use cases envisioned for the third milestone.

The integration was significantly facilitated by the usage of the ROS middleware. We managed to handle two similar, but not exactly equivalent, physical robotic setups, one at Tecnalia, and another at Astech. This required some effort on the software structure and module configuration, but it enabled us to get a system that can be more easily deployed on new sites, or more easily adjusted to cope with any evolution of the experimentation site. We expect to take this opportunity to deploy the complete system at other partner sites. Such effort permit to increase the usage of the system, which enables to identify faster potential issues, and therefore improve its stability and robustness.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

9 Annexes

9.1 Table of Figures

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

9.2 Table of Videos

A set of video is made available on the TraceBot YouTube playlist[2], to showcase the advancement of the integration and of the development:

- **VIDEO 1 - 2022 07 TRACEBOT TECNALIA CANISTER**

  Canister insertion, at Tecnalia facilities, using fiducial markers for perception

- **VIDEO 2 - WP3 TECNALIA NEEDLE MANIPULATION**

  Needle manipulation, at Tecnalia facilities, using fiducial markers for perception

- **VIDEO 3 - TRACEBOT WP2 MILESTONE2**

  Advancement on the design & implementation of the modular finger embedding tactile sensing

- **VIDEO 4 - TRACEBOT WP3 CEA INSERT CANISTERS**

  Current mockup of the Cognitive programming interface, to be used by the operator to generate the process of skills

- **VIDEO 5 - 2021-CEA-PIEZOR**

  Tactile sensing usage for detecting and characterizing the contacts in between the gripper fingers and the manipulated objects

- **VIDEO 6 - 2022-CEA-PIEZOE**

  Tactile sensing usage for detecting events, such as the “click” while closing the clamp.

- **VIDEO 7 - TRACEBOT WP5: DIGITAL TWIN OVERVIEW**

  Demonstration of the Digital Twin functionality

- **VIDEO 8 - TRACEBOT WP5: DIGITAL TWIN AND FAILURE REASONING**

  Demonstration of the verification functionality provided by the Digital Twin

- **VIDEO 9 - TRACEBOT WP6: NEEDLE INSERTION DEMO (ARUCO MARKERS)**

  Demonstrator deployed at Astech: needle manipulation (with fiducial markers)

- **VIDEO 10 - TRACEBOT WP6: CANISTER INSERTION DEMO (ARUCO MARKERS)**

  Demonstrator deployed at Astech: canister manipulation (with fiducial markers)

- **VIDEO 11 - TRACEBOT WP6: CANISTER INSERTION DEMO (LOCATE OBJECT PACKAGE)**

  Demonstrator deployed at Astech: canister manipulation, using the perception layer developed in TraceBot.

[2] https://youtube.com/playlist?list=PLWmZxrszLjCT6ChEEeFKAsXGIHkD7zDAo

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

10 References

[1] "CONNECTING AN EMERGENCY STOP DEVICE VS. A SAFEGUARD PROTECTIVE DEVICE," Universal Robots, 4 April 2020. [Online]. Available: https://www.universal-robots.com/articles/ur/application-installation/connecting-an-emergency-stop-device-vs-a-safeguard-protective-device/.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)