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 simplified programming framework – Fina
Reference	**:**	D4.4
Availability	**:**	Public
Date	**:**	30/09/2019
Author/s	**:**	TECNALIA
Circulation	**:**	EU, consortium

Summary:

This document provides an overview of developed easy-programming modules for THOMAS project (WP4-T4.1). The developments provide the following features: intuitive tools for programming through skills/primitives, a GUI for managing the skills and a generic dashboard for robot handling.

Table of Contents

1. LIST OF FIGURES

2. LIST OF TABLES

No table of figures entries found.

3. EXECUTIVE SUMMARY

THOMAS project introduces a new method for programming robot actions. This method focuses on the skills and primitives. The proposed skill-based programming technique based on skills' and primitives' linking actions. Under THOMAS project, skills and primitives create executable XML files. These files may be exported using a CATIA module developed by TECNALIA for this purpose. This is an offline programming method of robots based on the digital layout of the shopfloor in CATIA software. The main target of this document is to present the latest version of the easy-programming modules for THOMAS project previously documented in deliverable D4.2. The execution of robot tasks using the CATIA software for skill's configuration is being described in the next sections.

The execution of most Aeronautic use case's operations using the CATIA GUI is documented in this deliverable.

Last but not least, robots' programming inside THOMAS project may also be achieved through a generic dashboard designed by TECNALIA. This dashboard offers three types of GUIs for mobile robot's navigation and arms' motion actions. Using this user-friendly dashboard GUI, the user is capable to easily programming the robot. The following GUI are available through this generic dashboard:

- GUI for motion in cartesian space
- GUI for motion in joint space
- GUI for navigation actions

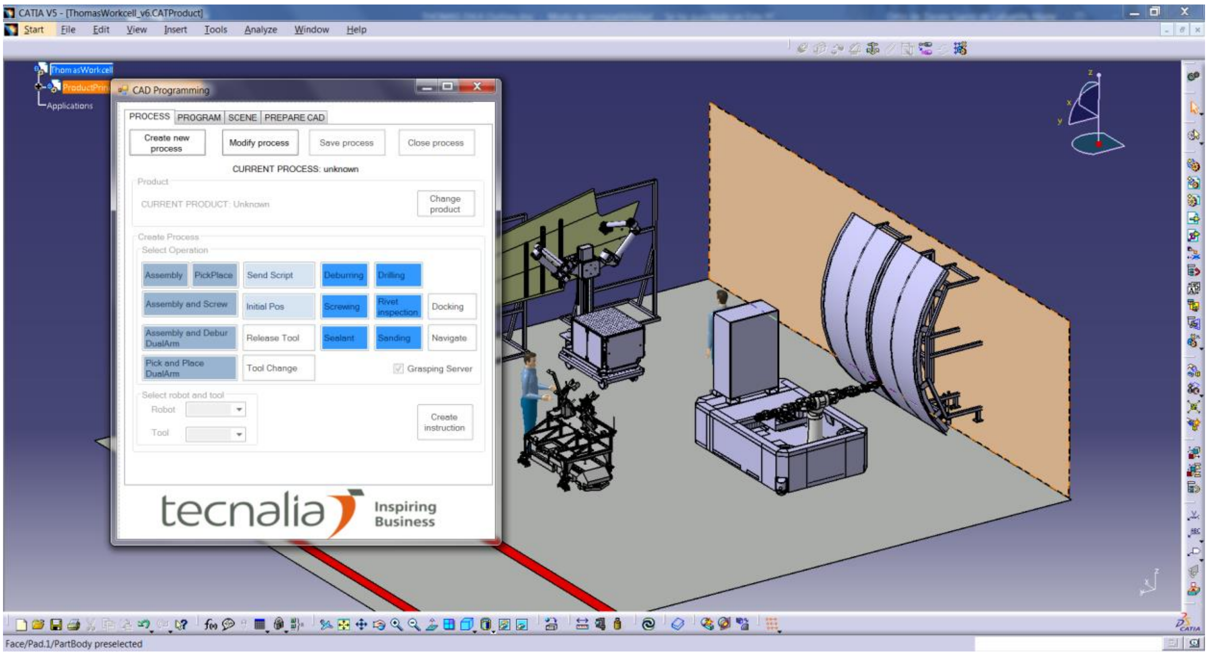

Figure 1: CAD programming tool based on CATIA software

4. INTRODUCTION

This document provides an overview of developed easy-programming modules for THOMAS project. Most of the documented modules have previously be documented in deliverable D4.2. The ongoing developments provide the following features: an intuitive tool for programming through skills/primitives, a GUI for managing the developed skills and a generic dashboard for robot handling. Skill programming technique based on a digital layout of the shopfloor in CATIA software and a custom GUI for CATIA software designed by TECNALIA. Skill parameterization may be achieved through this GUI too.

All the information documented in the following sections as follows:

- The final version of the skill parametrization and parameter's mapping are documented in section 5.
- Section 6 consists of the final version of CATIA GUI used for the configuration and programming of all robot tasks inside THOMAS Aeronautic use case.
- Finally, the final version of the dashboards may be used for robot's easy programming are presented in Section 7.

Through these tools the work done by the partners of the consortium (vision, navigation, manipulations, safety, etc.) can be easily parametrized and integrated into the MRP.

5. SKILL PARAMETRIZATION

5.1. Skill definition

Robot skills are a way of representing human capabilities through the composition of basic functionalities (primitives). In terms of implementation, a skill is no more than a mechanism for representing, storing and exchanging the links between primitives. The skills do not contain implementation code. A skill can be composed by other skills (there is no limit in the number of levels) and by primitives, e.g. a skill can be composed by one unique primitive (probably because the user could understand better the behaviour with provided name) or can be composed by a mix of skills and primitives. Due to this reason, the term action is used in the following lines. An action refers to a primitive or a skill. This enables to represent a sequence of operations in a XML file regardless of the element type. In the D4.5 the developed skills for THOMAS use cases are described in detail.

5.2. Parameter mapping

The parameter mapping consists on relating appropriately the input/output parameters of each developed skill or available primitive for the MRP. Since the Skill Based Programming is based on code re-using and in the capability encapsulation and isolation, a way for linking skills together and with the process itself is required.

On the one hand, in the skill/primitive level interaction, the necessary communication is implemented linking the input and output by reference. In the developed tool, despite the process of linking skill inputs/outputs is still manual, and not very user friendly, this is a topic that has been improving along the phases of the project, introducing improvements in usability that have reduced the set-up time and possible errors. Regarding the additional parameters, as can be seen in the Figure 2, the current prototype allows parameterizing the skills and primitives with literal values or process information (introduced below).

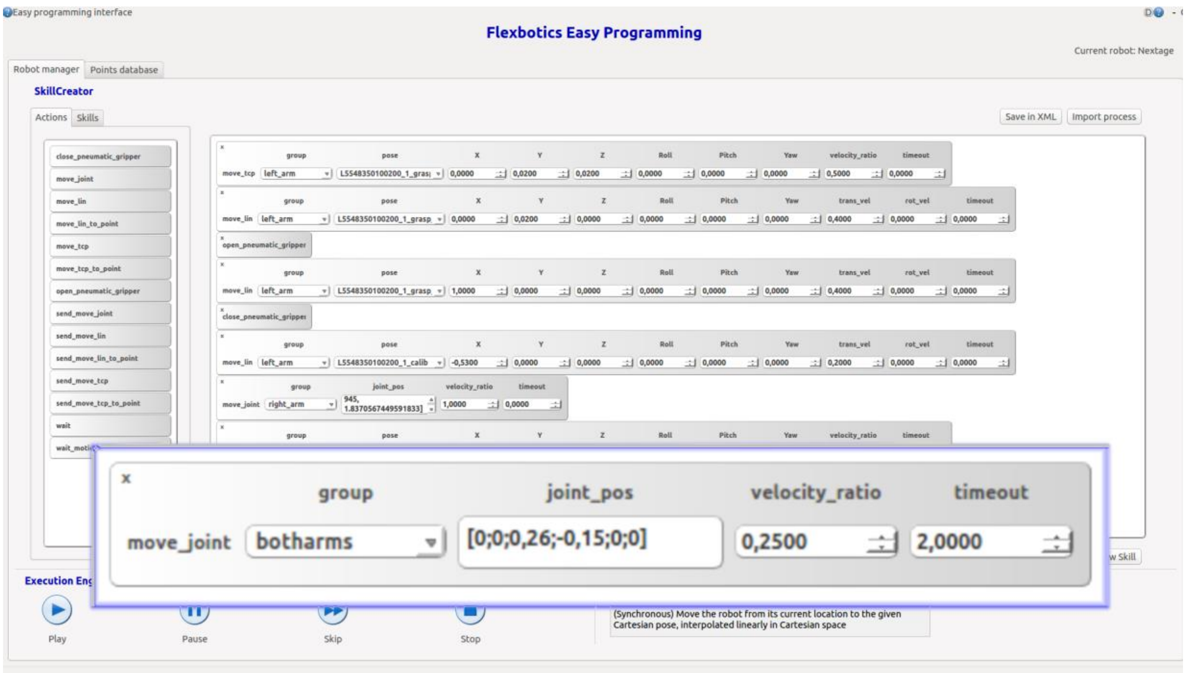

Figure 2: Skill/Primitive parametrization

On the other hand, the relation with the actual process is carried out through the support of CAD Based Programming (Figure 3). Thanks to the developed tools inside of CATIA, the relevant information of the process (reference frames, object features, object relative positions, etc.) can be easily extracted and linked with the developed skills.

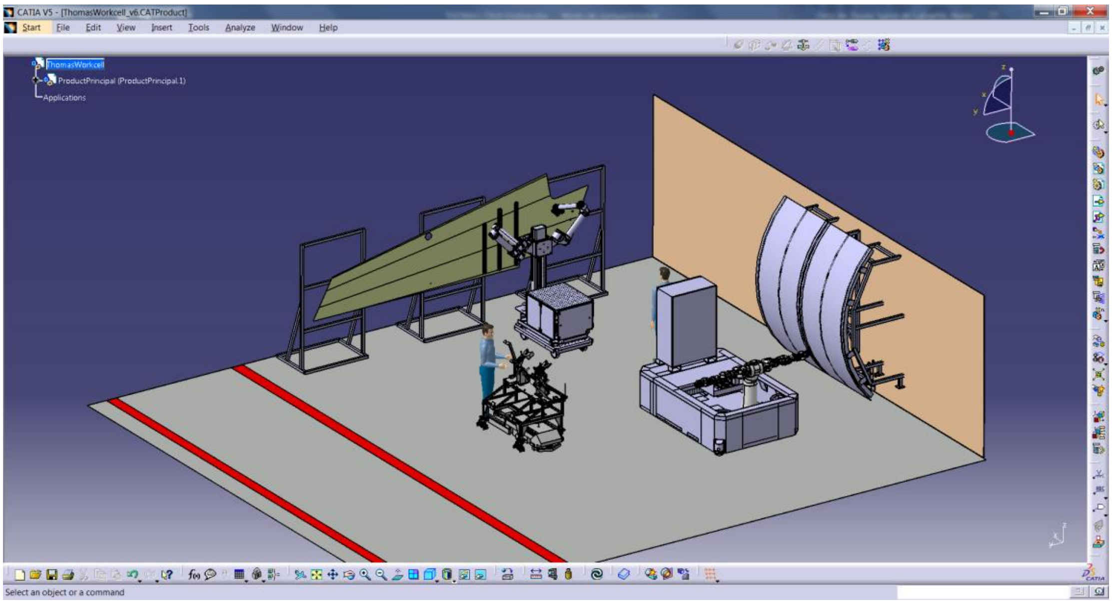

Figure 3: CATIA scene containing all the involved elements in the process

6. CATIA GUI FOR SKILL CONFIGURATION

6.1. Description

The implemented CATIA modules support Skill Based Programming providing actual process information and easing skill parametrization. It is an offline tool that can be used on the cell set-up and every time that the process suffers changes in their element references or locations (Figure 1,Figure 4).

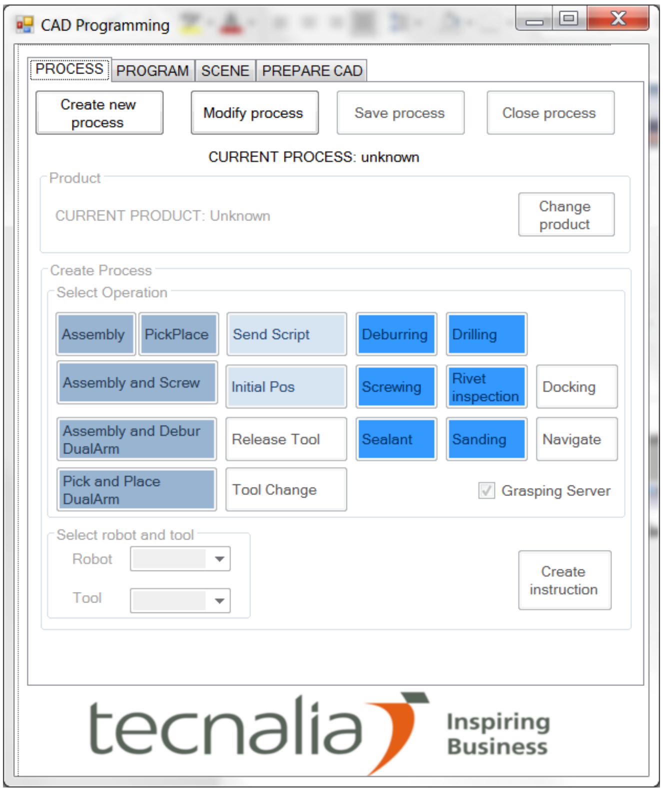

Figure 4: CAD Programming interface

6.2. Features

Through developed CATIA modules the following information can be extracted and exported to XML files:

- 3D scene of the cell. Composed by all the involved CAD models.

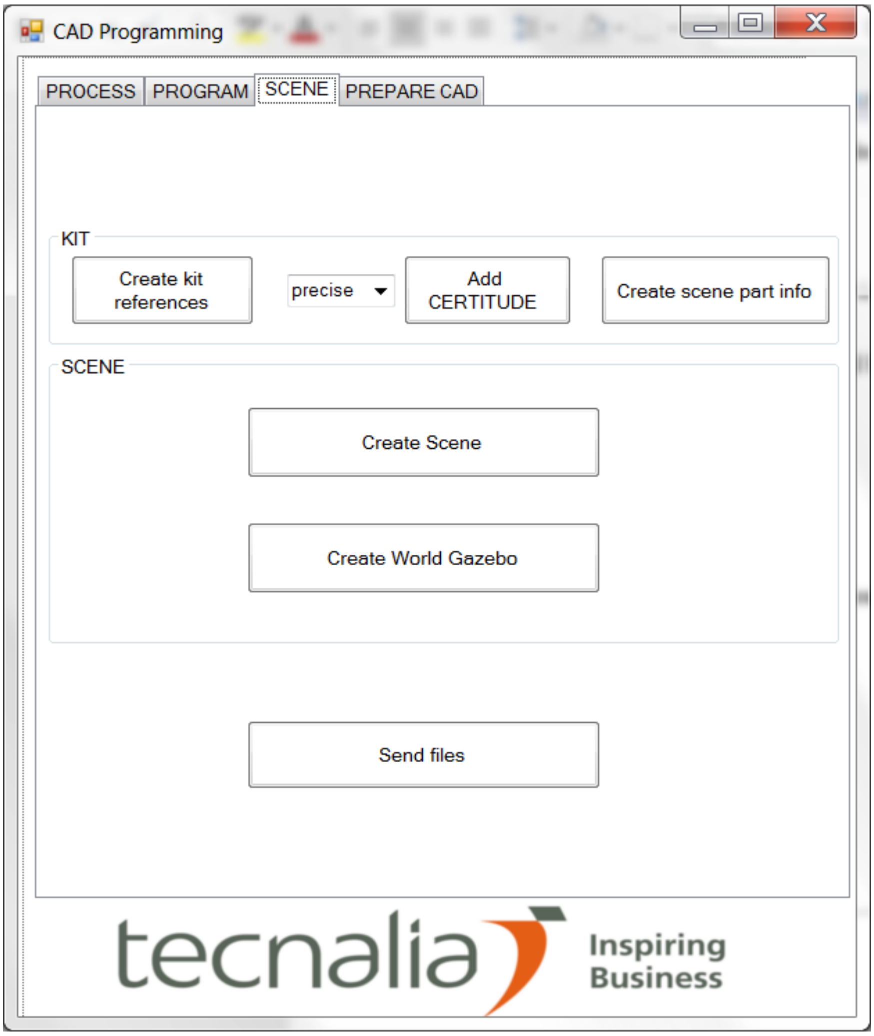

Figure 5: CAD Programming tool: create scene

- Reference frames of the elements. Not only the frames provided by the cell owner, but also new auxiliary frames that could help.
- Object features, namely: holes (Figure 6), relevant positions, edges, etc.

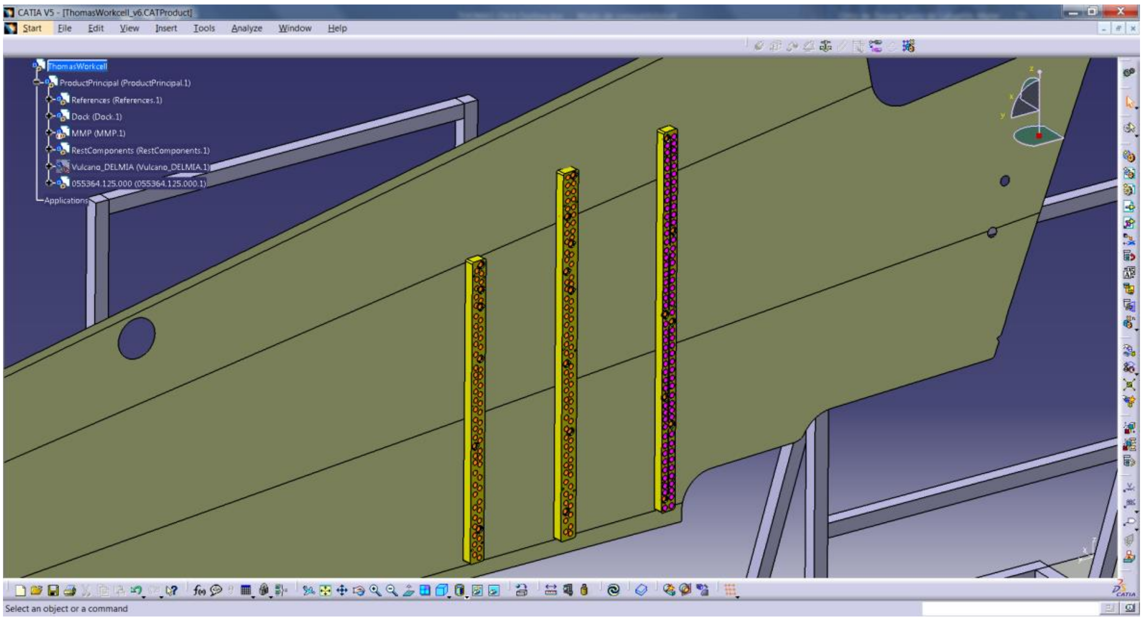

Figure 6: Hole extraction from CATIA

- Process related information: grasping poses and assembly poses. Currently this information must be added manually following a process that requires specific training.

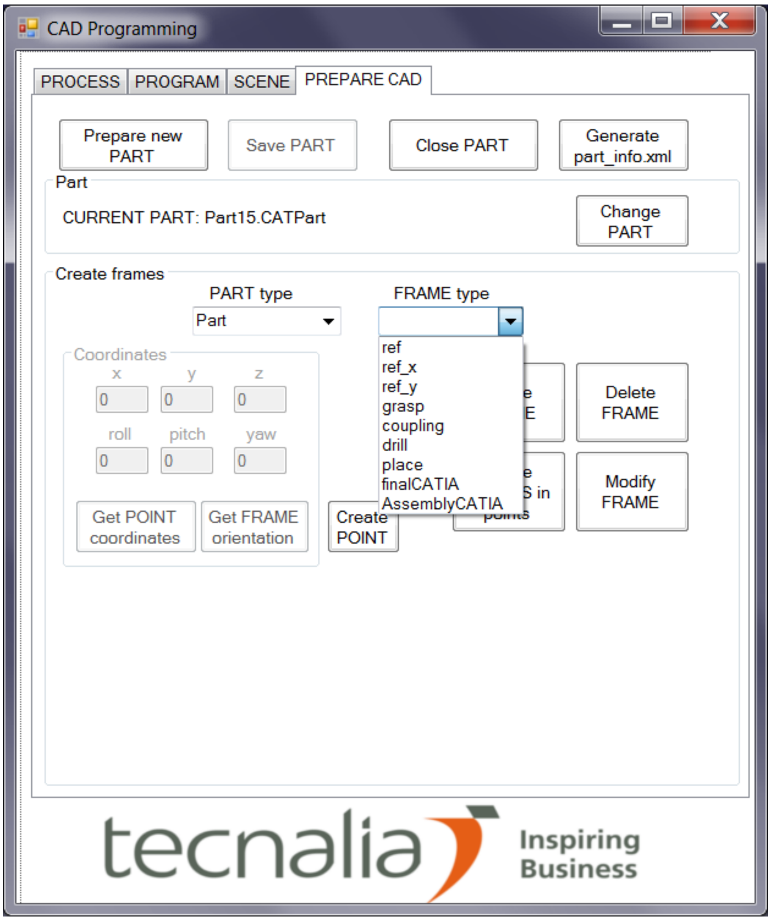

Figure 7: CAD Programming: prepare CAD

- Repetitive task parametrization taking advantage of previously prepared skill. Some skills can be prepared for being parametrized from CAD Based Programming.

- Process: not only the process sequence, but also the information of each skills. The available skills for CAD Based Programming are: tool exchanging, navigating, docking, drilling, sanding and rivet inspection (in progress).

6.2.1. Tool change/Release tool

The *tool exchange* feature requires selecting the desired tool for a specific robot (in the case of MRP, *left_arm* and *right_arm* are the available robots). In addition, the equipped tools can be released anytime throughout the process.

Figure 8: CAD Programming: tool change

6.2.2. Navigate

The *navigate* skill allows the user creating an instruction for moving the robot along the workspace. The application will ask the user to select the robot and its target position (frame).

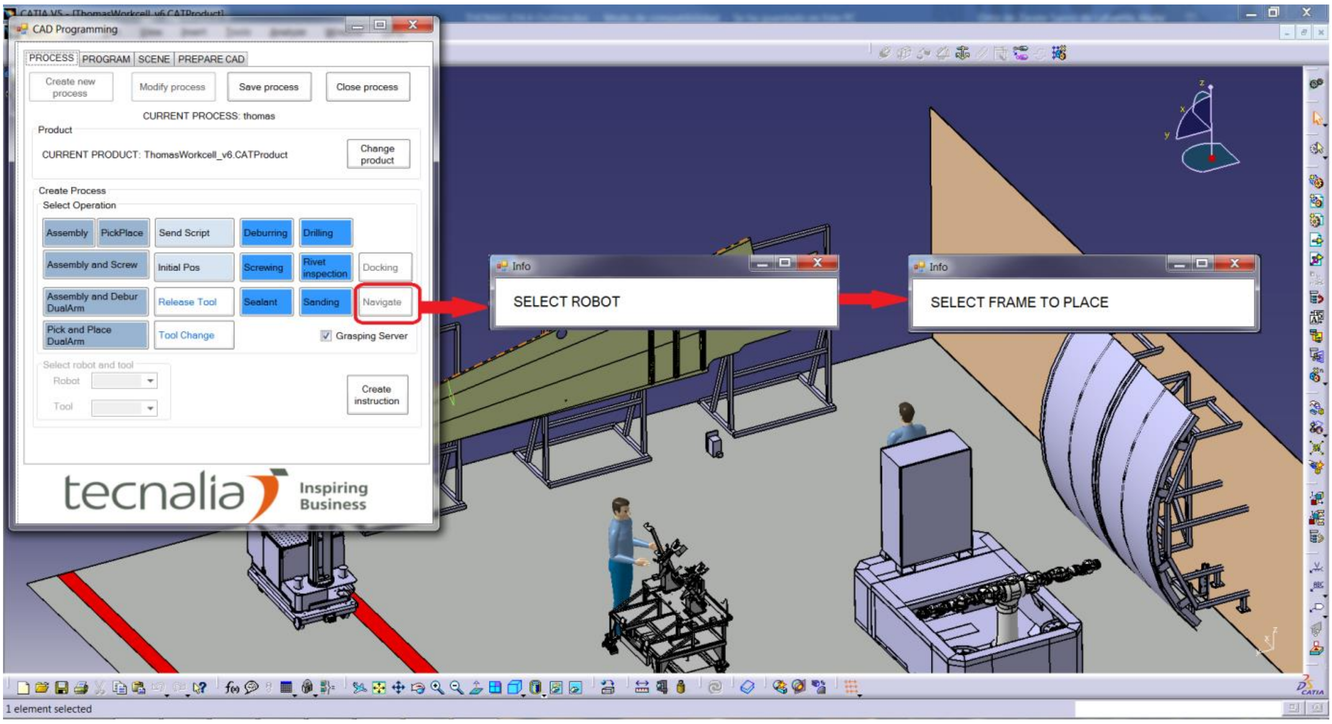

Figure 9: CAD Programming: Navigate

6.2.3. Docking

The *docking* skill button creates the required instruction for aligning the robot precisely in the desired docking place with the provided precision.

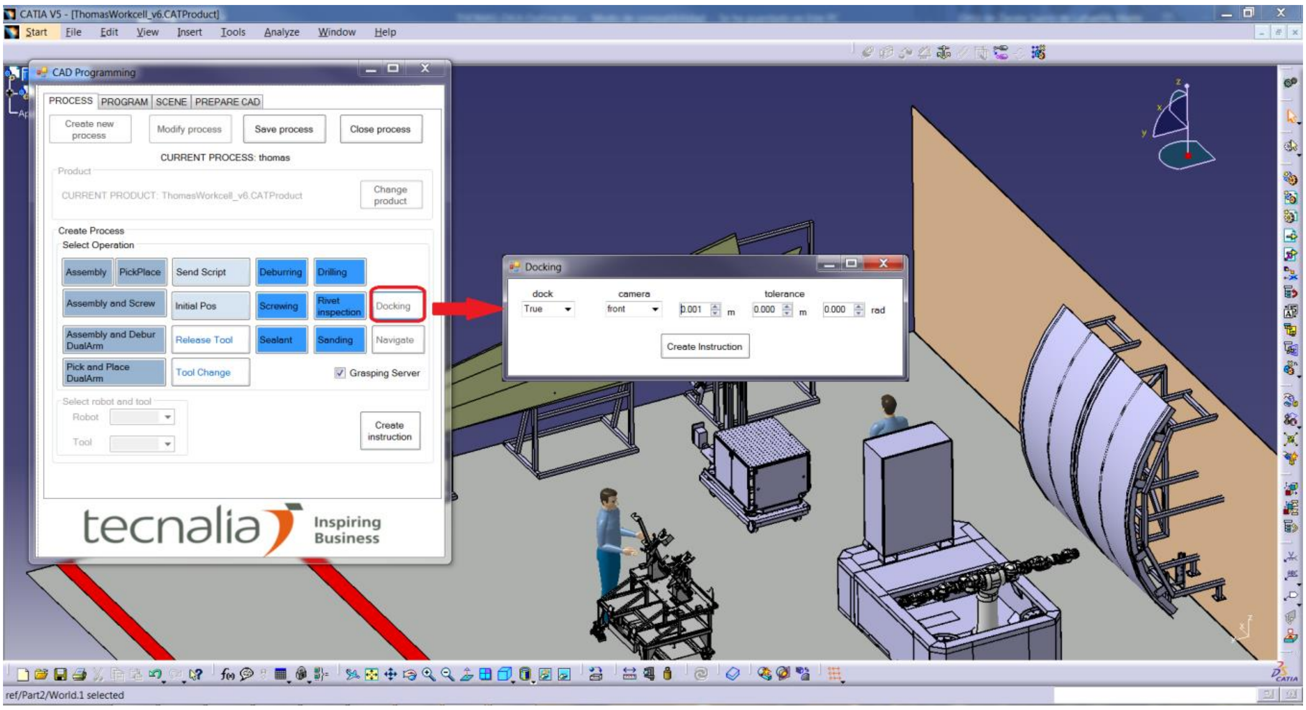

Figure 10: CAD Programming: Docking

6.2.4. Drilling

A *drilling* operation is configured through drilling skill button. Firstly, selecting the template to be drilled is mandatory, then the drilling frames (one per hole) will be visible. Finally, the user must select which frames need to be drilled in the performed operation.

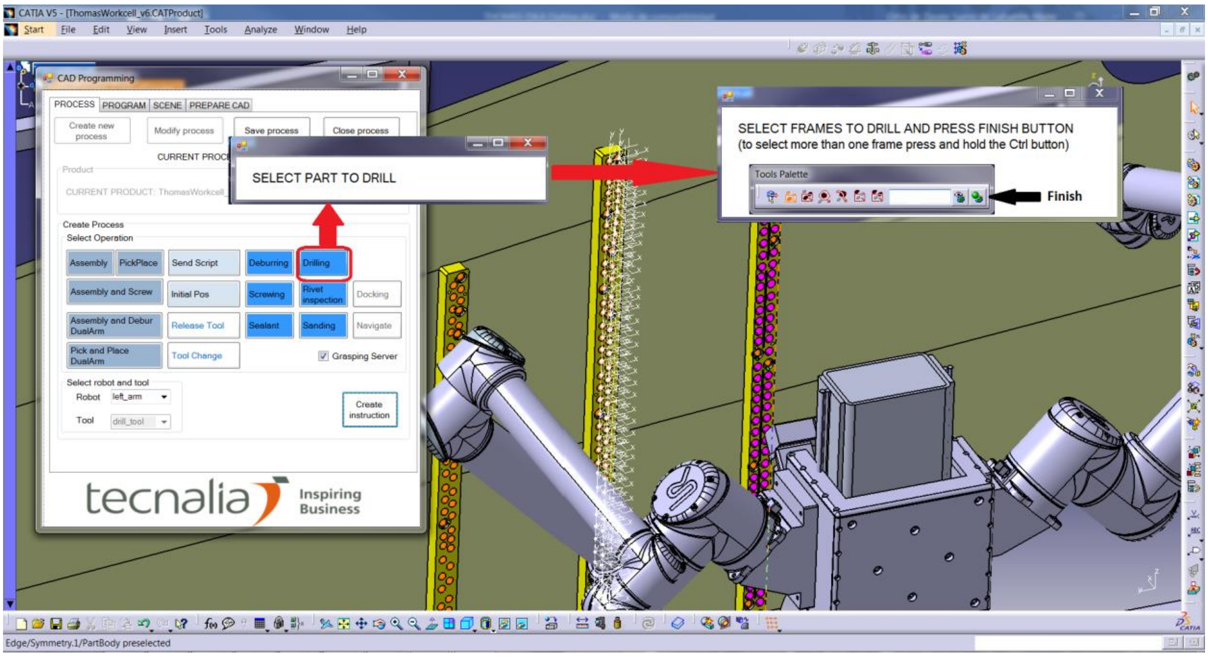

Figure 11: CAD Programming: Drilling

6.2.5. Sanding

The *sanding* skill prepares the robot for performing a sanding operation to the selected part.

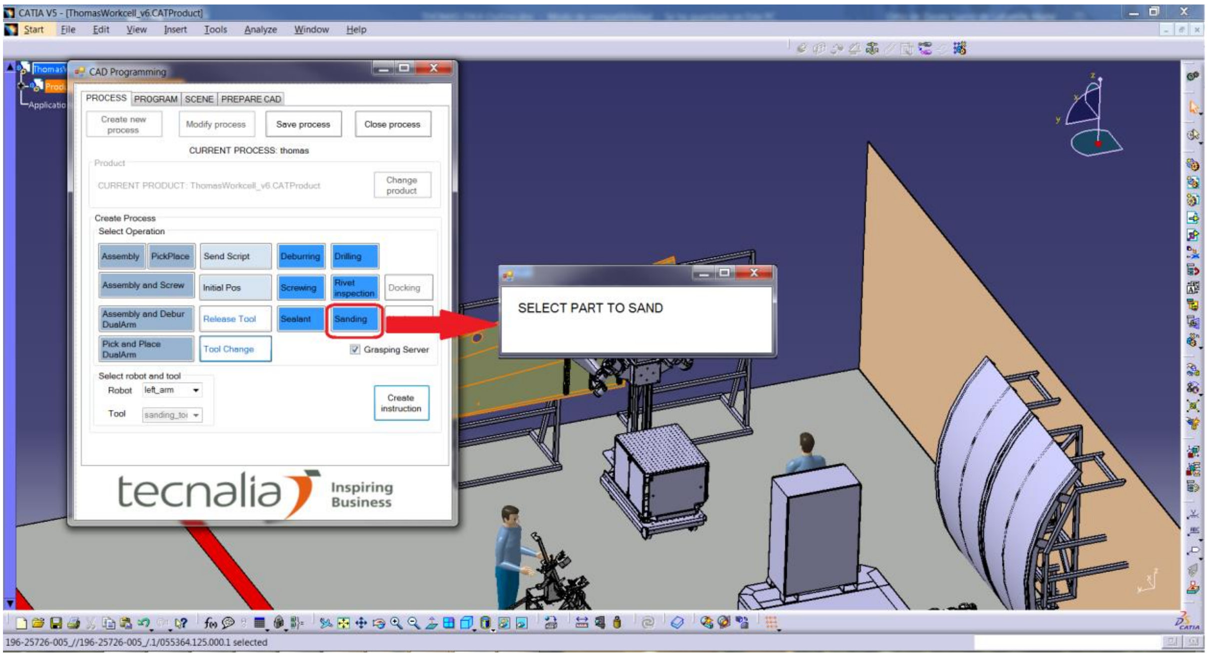

Figure 12: CAD Programming: Sanding

6.2.6. Rivet inspection

The *rivet inspection* skill button produces the required instruction for executing a rivet inspection process.

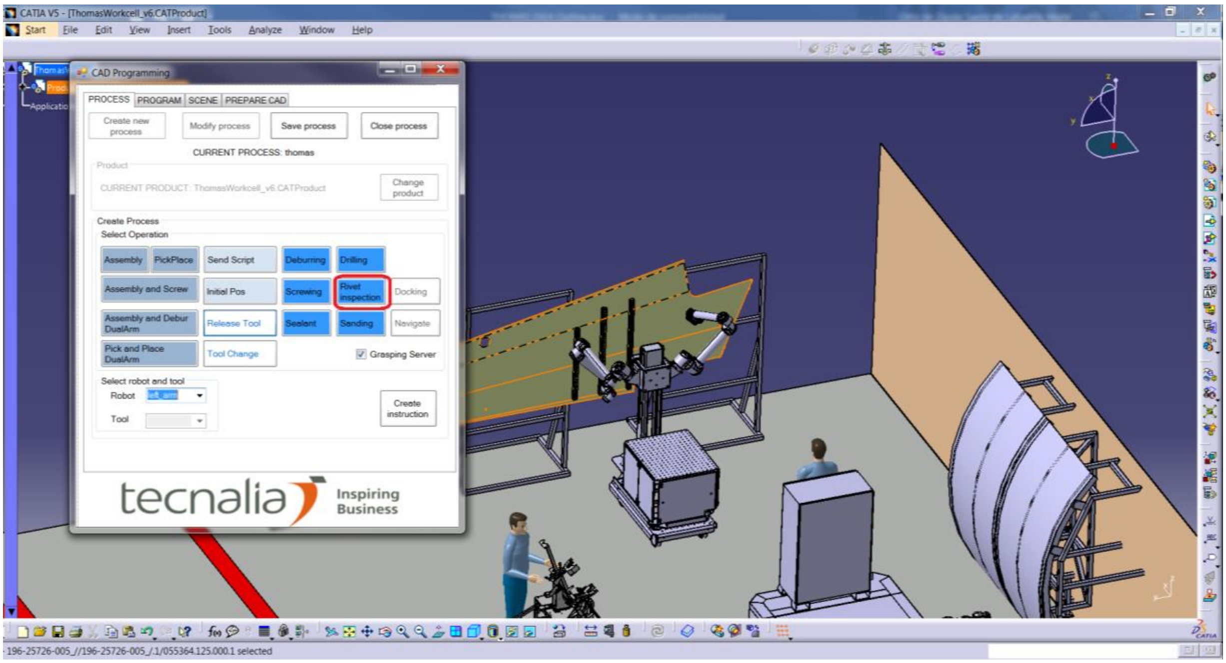

Figure 13: CAD Programming: Rivet inspection

6.3. Provided output

The extracted information is exported in different formats depending of the nature of the data. For example, the 3D scene information follows a specific process that allows being compatible with ROS, and more concretely, with MoveIt! [1] package. The cell elements are exported to STL format and after that are stored in a *.scene* format file, the format which MoveIt! manages the 3D models.

Regarding the reference frames, object features and process related information, this information is exported in XML files following an ad-hoc convention that looks as can be seen in (Figure 14).


```
<transform_list xmlns="http://www.flexbotics.com/Schemas">
 <transform>
  <parent_frame>base_link</parent_frame>
  <child_frame>element_1</child_frame>
  <position>
   <x>-0.7553</x>
   <y>0.4724</y>
   <z>-0.0503</z>
  </position>
  <orientation>
   <rpy>
    <r>1.56903332</r>
    <p>0.02335458</p>
    <y>1.55715990</y>
   </rpy>
  </orientation>
 </transform>
</transform_list>
```


Figure 14: CATIA extracted information convention

The CATIA GUI can also export the executable process files (the sequencing of skills) with the TECNALIA's Robot Framework format, as can be seen in Figure 15.

```
<process version="1.0">
 <action name="approach_to_template_detection_pose">
            <parameters>
    <param name="robot_group">
     <value type="data">left_arm</value>
    </param>
    <param name="template_vision_approach_frame">
     <value type="data">template_vision_approach</value>
    </param>
    <param name="tool_frame_id">
     <value type="data">left_arm_tool0</value>
    </param>
   </parameters>
   <result />
 </action>
 <action name="detect_template_pose">
   <parameters />
   <result />
 </action>

...

</process>
```

Figure 15: Sequence of skill are stored in process files

7. Intuitive programming through skill/primitive sequencing

7.1. Generic dashboard for robot handling

Even though the development of a dashboard for robot handling was not the main objective of this WP, the complexity of the MRP (a mobile platform + 2 DOF torso + 2 UR robot + pan/tilt) has required the development of a tool that allows handling the complete platform easily. In order to provide an additional added value, the developed dashboard is generic and can be re-used for all compatible robots inside of TECNALIA's Robot Framework. Through an utility that takes the URDF and SRDF of the robot, the dashboard adapts to each robot and exposes their available joints.

The Generic Dashboard allow handling robots in Cartesian space (Figure 16) and Joint space (Figure 17). For the robots that have the capability of navigation a simple joy has been integrated for precise guiding (Figure 18).

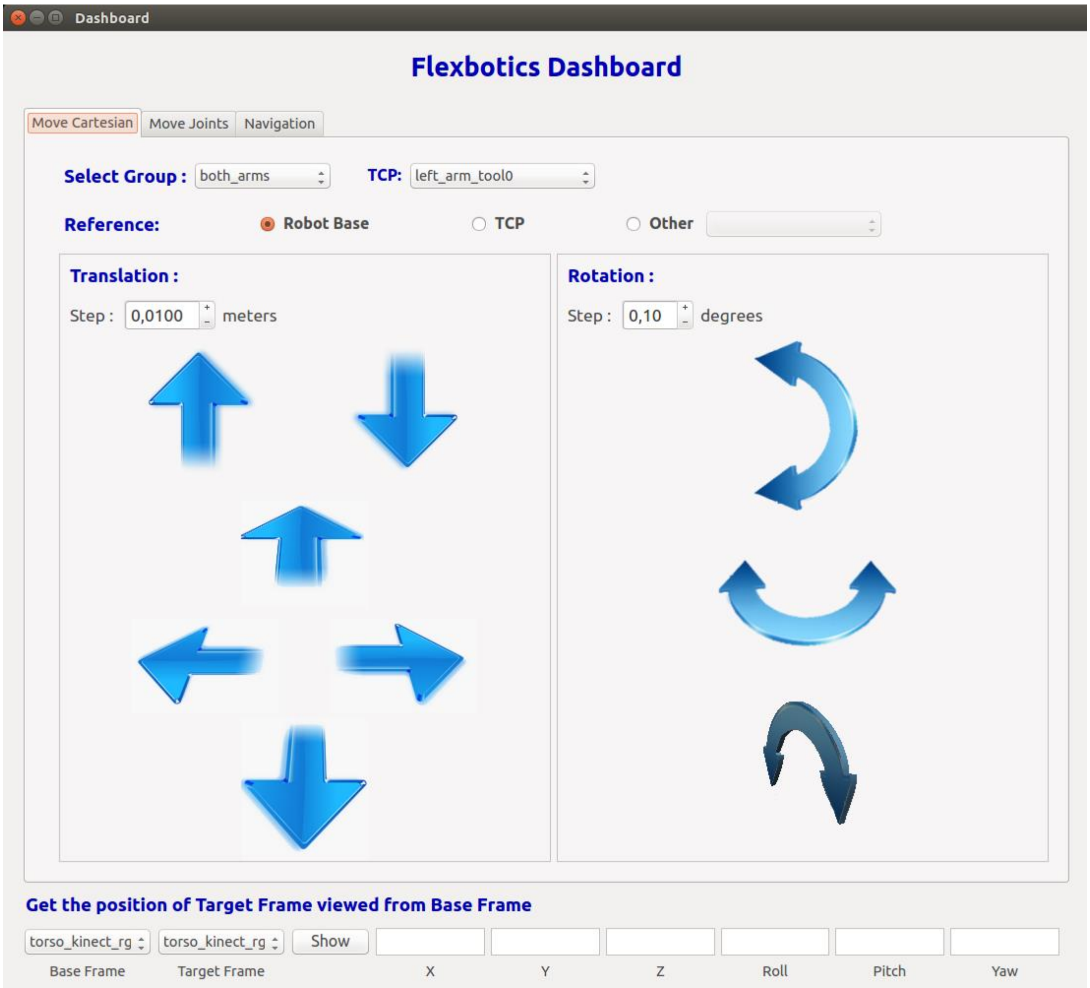

Figure 16: GUI for Cartesian space

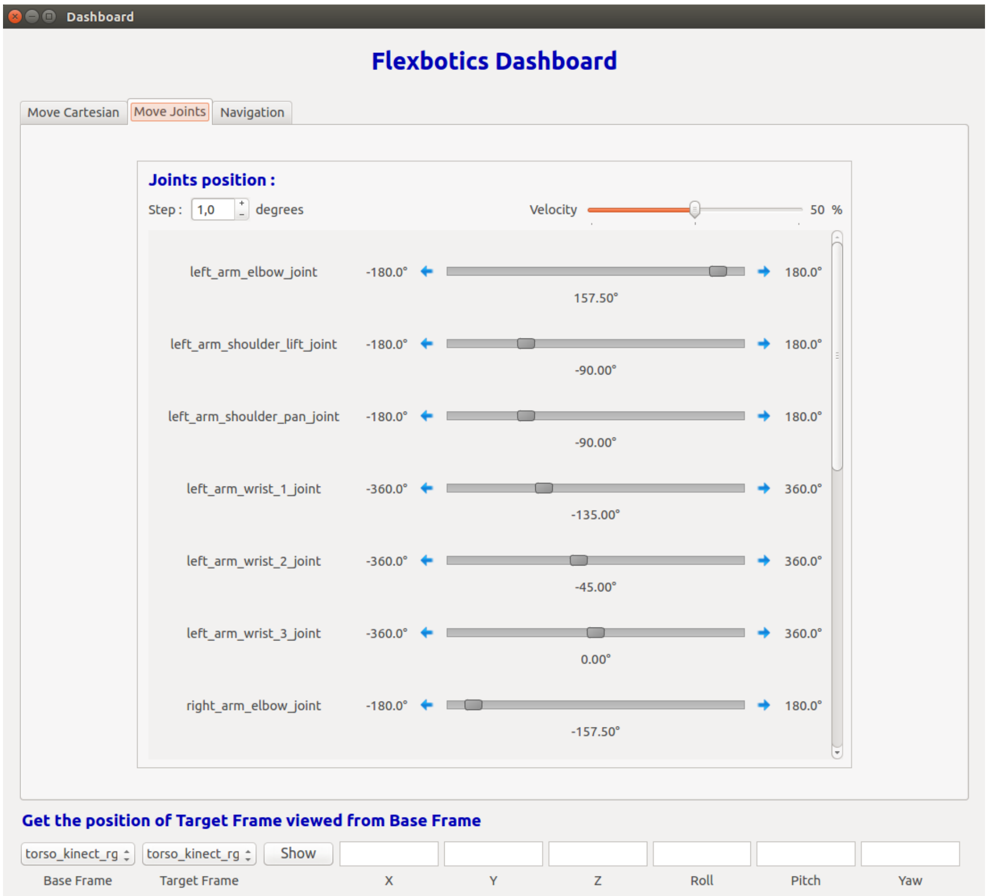

Figure 17: GUI for Joint space

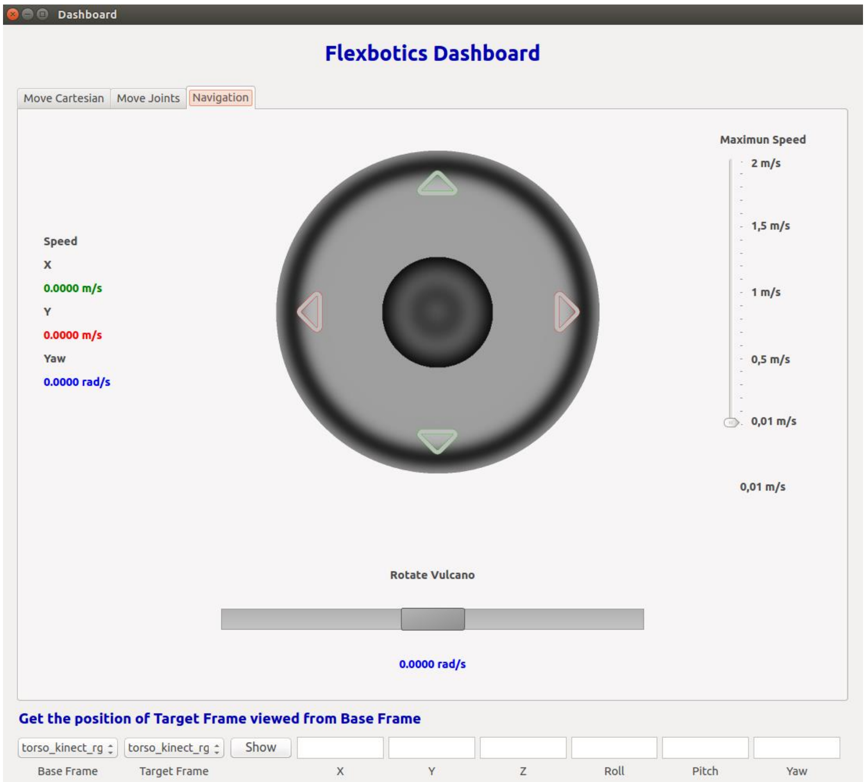

Figure 18: GUI for the navigation

7.2. Drag & drop easy programming GUI for skill/primitive sequencing

The MRP has been integrated within TECNALIA's Robot Framework, taking advantage of all previously developed modules, libraries and skills. Basically, the framework provides an API with a series of methods that can be seen as the capabilities of the robot. These methods of the API are also known as the available primitives for the Skill Based Programming Framework. As can be seen in the following figure (Figure 19) in the left block previously mentioned primitives can be found. In the main panel a sequence of primitives have been drag & dropped, and each of them has been parametrized appropriately as has been mentioned in Section 5.2. This sequence of primitives can be stored either as an executable program (introduced process file at Section 6.3 and D4.1) or as a new skill (see D4.1).

When a sequence of primitives is stored as skill, the resultant skill is stored in the Skill Library. In the *Skills* tab of the left block the available skills can be found. Figure 20 shows how this skills can be drag and dropped to the main panel for sequencing them and combining with other skills or primitives. In the same way than before, these sequences can be parametrized and stored either as process or as a new skill. The skills in the main panel can be collapsed in order to simplify the presentation. Figure 21 shows how the collapsed view generates an executable process file.

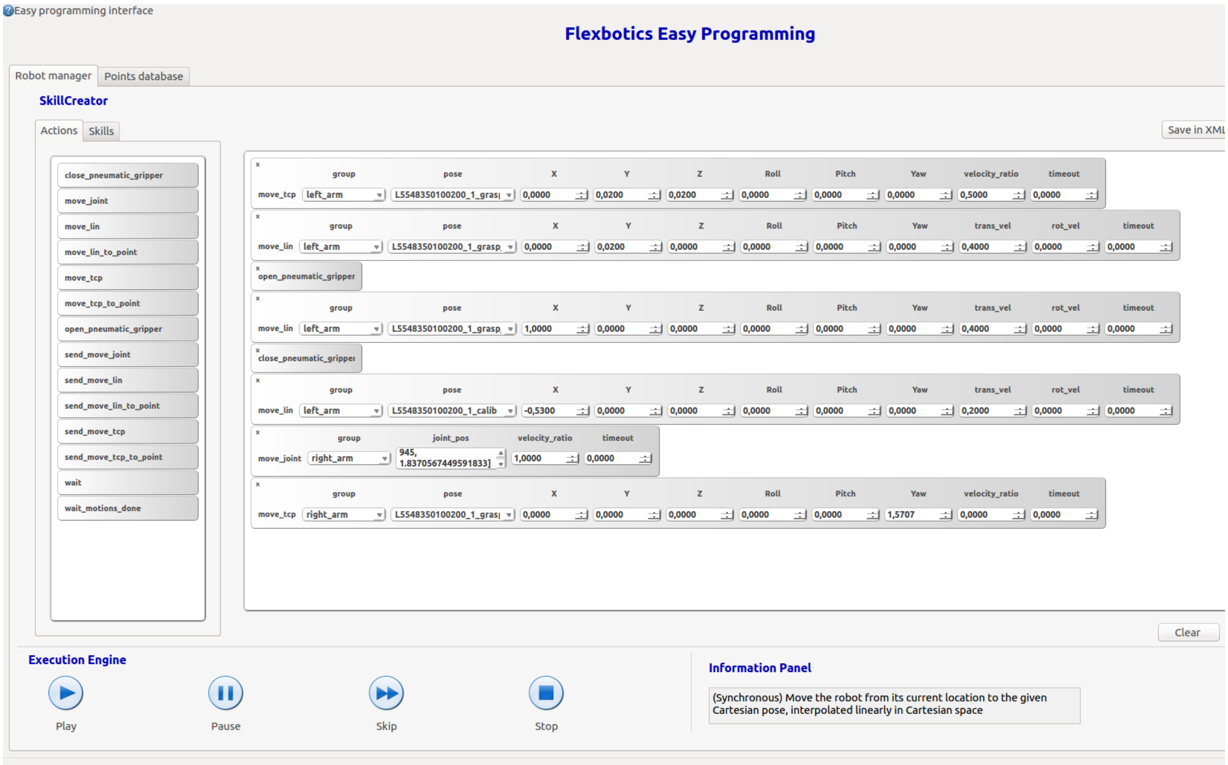

Figure 19: Easy Programming GUI

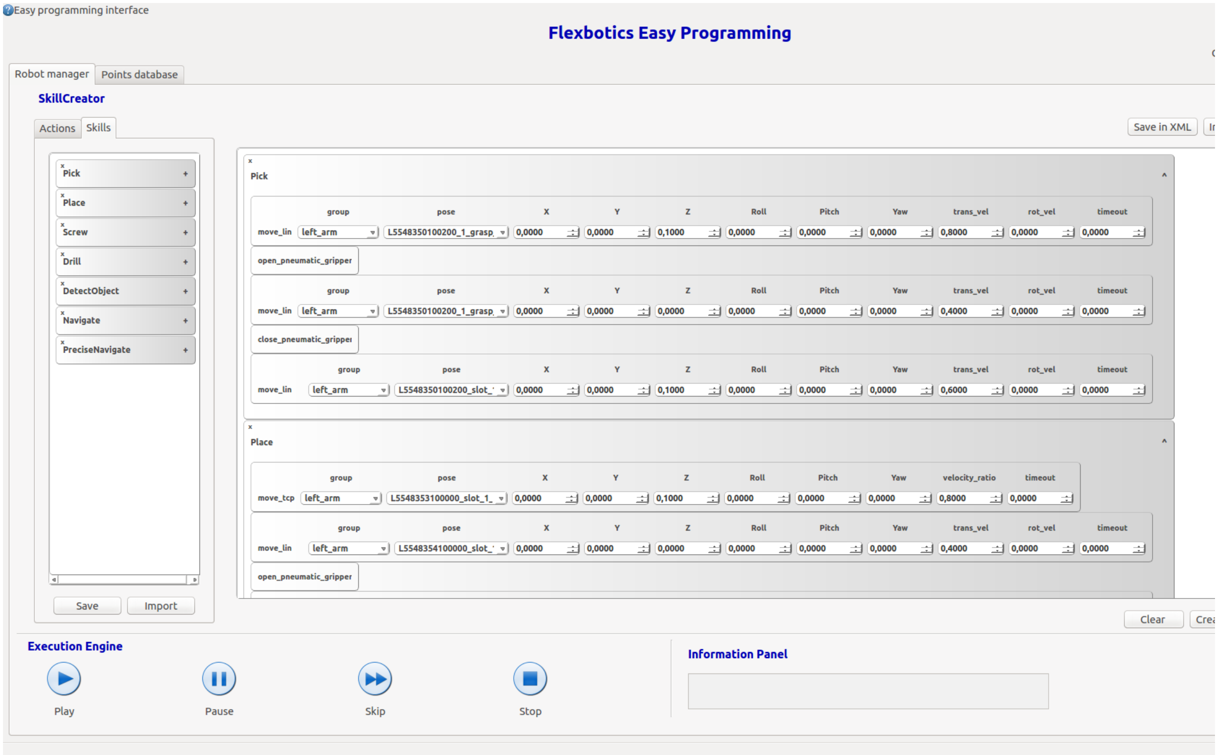

Figure 20: New skill creation

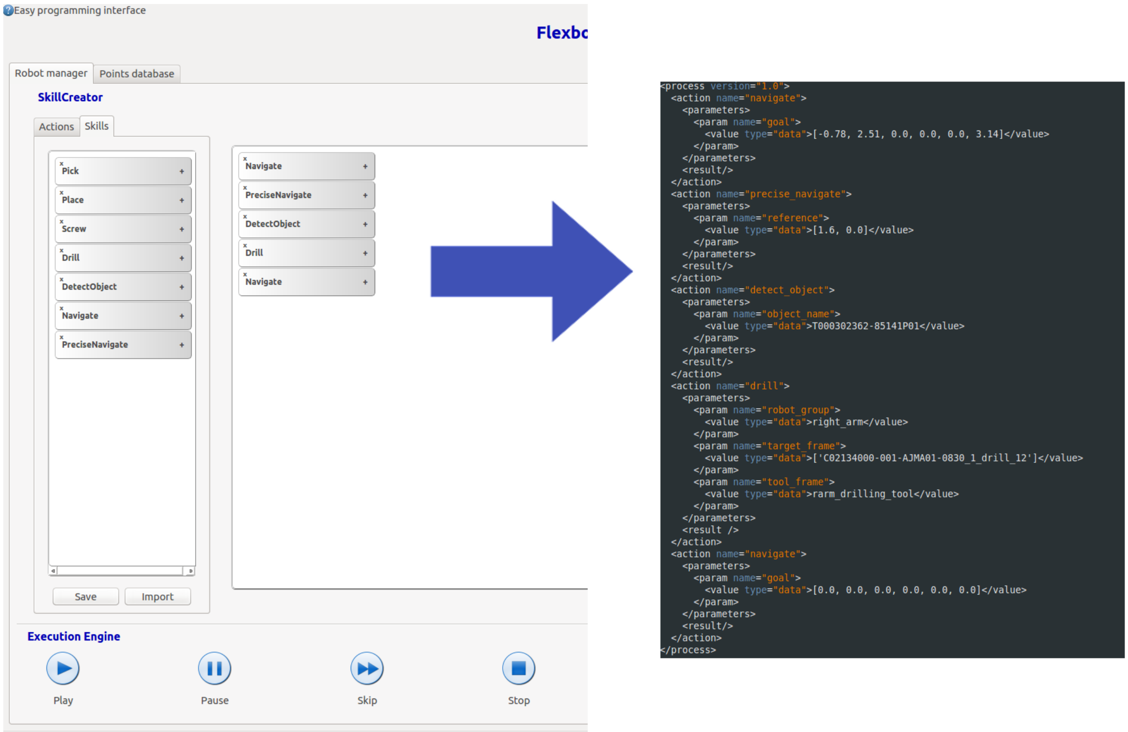

Figure 21: Process generation from a sequence of skills

8. Conclusions

This document summarizes all the actions of the consortium around the fourth Work Package of THOMAS project and especially inside the Task 4.1. This task aims to introduce new methods for mobile dual arm robots' programming inside any industrial shopfloor. As already presented in the previous section, the first programming method based on skills and primitives but also on a custom GUI designed by TECNALIA for the CATIA software. Multiple executable files for robots' actions may be generated through this GUI automatically. Using this GUI, the operator is capable to parameterize, store and retrieve any generated executable file for robot actions' execution. Skill based programming use a user-friendly GUI for the initialization and connection of skills.

The second programming method based on a generic dashboard for robot's controlling. This dashboard enables both platform's (navigation) and arms' (motion) controlling through 3 available GUIs. These GUIs have already presented in the last section of this document.

Both methods integrated and tested inside THOMAS Aeronautic use case using the Mobile Robot Platform of TECNALIA partner.

9. GLOSSARY

XML	eXtensible markup language
ROS	Robot Operating System
GUI	Graphical User Interface
CAD	Computer-aided Design
MRP	Mobile Robot Platform
DOF	Degree of freedom
URDF	Unified Robot Description Format
SRDF	Semantic Robot Description Format

10.REFERENCES

[1] Ioan A. Sucan and Sachin Chitta, "MoveIt!", [Online] Available: http://moveit.ros.org