# 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)**<br>**PEUGEOT CITROEN AUTOMOBILES S.A. (PSA)**<br>**SICK AG (SICK)**<br>**FUNDATION TECHNALIA RESEARCH & INNOVATION (TECNALIA)**<br>**ROBOCEPTION GMBH (ROBOCEPTION)**<br>**DGH ROBOTICA, AUTOMATIZACION Y MANTENIMIENTO INDUSTRIAL SA (DGH)**<br>**AERNNOVA AEROSPACE S.A.U. (AERNNOVA)**<br>**INTRASOFT INTERNATIONAL SA (INTRASOFT)** |



| | | |
|---|---|---|
| **Title** | : | Improved easy programming framework |
| **Reference** | : | D4.2 |
| **Availability** | : | Public version |
| **Date** | : | 31.03.2018 |
| **Author/s** | : | TECNALIA |
| **Circulation** | : | EU, consortium |


**Summary:**

*This document provides an overview of developed easy-programming modules for THOMAS project (WP4-T4.1). The ongoing developments provides 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

## LIST OF FIGURES

## LIST OF TABLES

**No table of figures entries found.**

# 1. EXECUTIVE SUMMARY

The main purpose of this document is to overview special tools for easy programming of MRP.

Robot's programming method that is going to be implemented in THOMAS project is based on skills and primitives. One robot program consists of a sequence of skills and primitives. The need for links between skills inputs/outputs is pointed out as a requirement for the Skill Base Programming. All skills developed under THOMAS use cases are introduced in D4.3. Executable process files in XML format can be exported automatically using one developed CATIA's GUI. It is an offline Skill Based Programming method using developed CATIA's modules for the linking of setup's information (distances between objects, information about object's geometry etc.) and robot's poses during assembly processes. This CATIA GUI can also be used for the configuration of skills. Robot's programming can also be achieved through one developed easy programming GUI based on a special API. This method consists of two stages:

- Creation of groups of primitives and save them either as independent executable programs or as skills in the Skill Library.
- Combination between loaded skills from Skill library and primitives for the creation of either a process or new skills.

One Generic Dashboard for compatible robot's handling in Cartesian and Joint space has been developed and is going to be presented in this document in parallel with the previous programming GUI (Figure 1). In terms of mobile robot units, one simple joy has been integrated for precise guiding in this Dashboard.

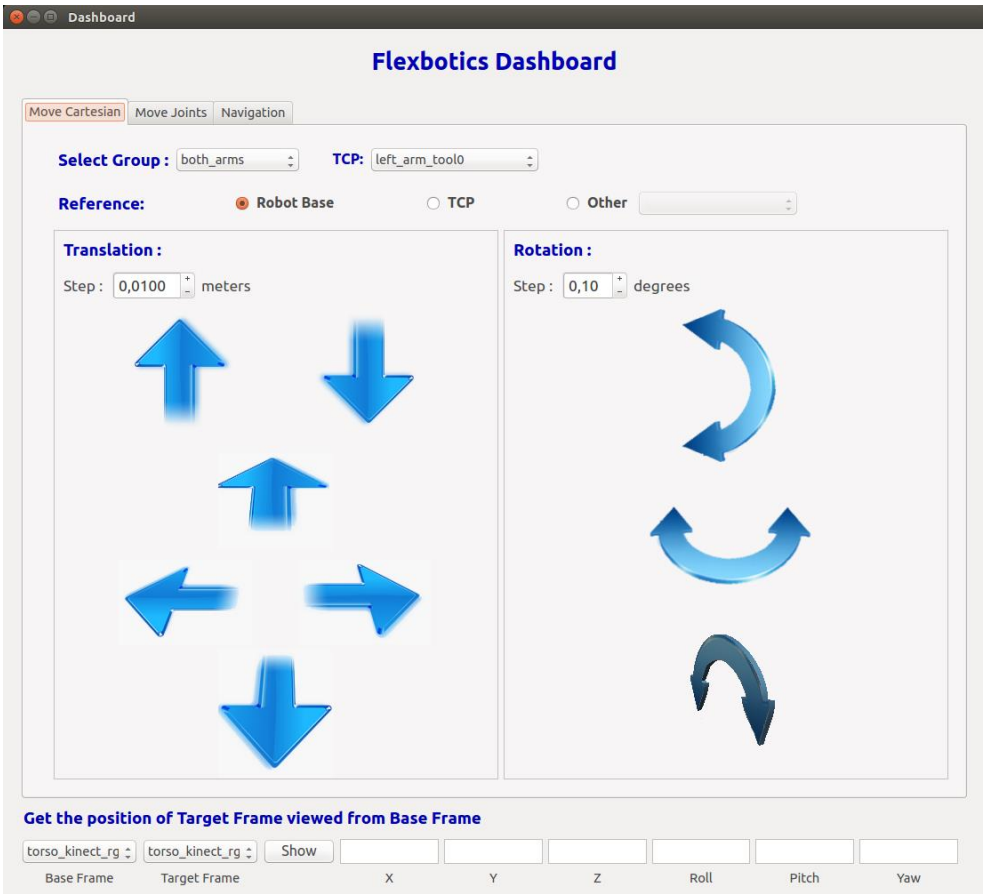


**Figure 1. GUI for Cartesian space**

# 2. INTRODUCTION

This document provides an overview of developed easy-programming modules for THOMAS project. Section 3 consists of a brief description of primitives and skills (as described in D4.1) while the need of remapping their parameters based on their inputs/outputs is pointed out. One developed GUI for robot's programming and skill's configuration process using CATIA software is being introduced in the next section of this document. The output of this GUI is either an executable process file including one sequence of skills or an XML file including 3D scene information. In Section 5, one Generic Dashboard for robot's manipulation and navigation is being introduced. This Dashboard works only with compatible robots inside TECNALIA's Robot Framework. In the same section a drag and drop GUI for easy robot programming is being described. In this way, robot processes can be created either as a group of skills and primitives or as a single skill. All involved skills for drilling, surface inspection, paint sanding and front axle damper assembly (as described in D4.1) can be created as a group of other skills and primitives through this GUI.

# 3. SKILL PARAMETRIZATION

## 3.1. 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 behavior 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 D4.3 the developed skills for THOMAS use cases are described.

## 3.2. 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 current prototype, the process of linking skill inputs/outputs is still manual, and not very user friendly. This is a topic that will be improved in subsequent phases of the project. 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).

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 relevant information of the process (reference frames, object features, object relative positions, etc.) can be easily extracted and linked with the developed skills.

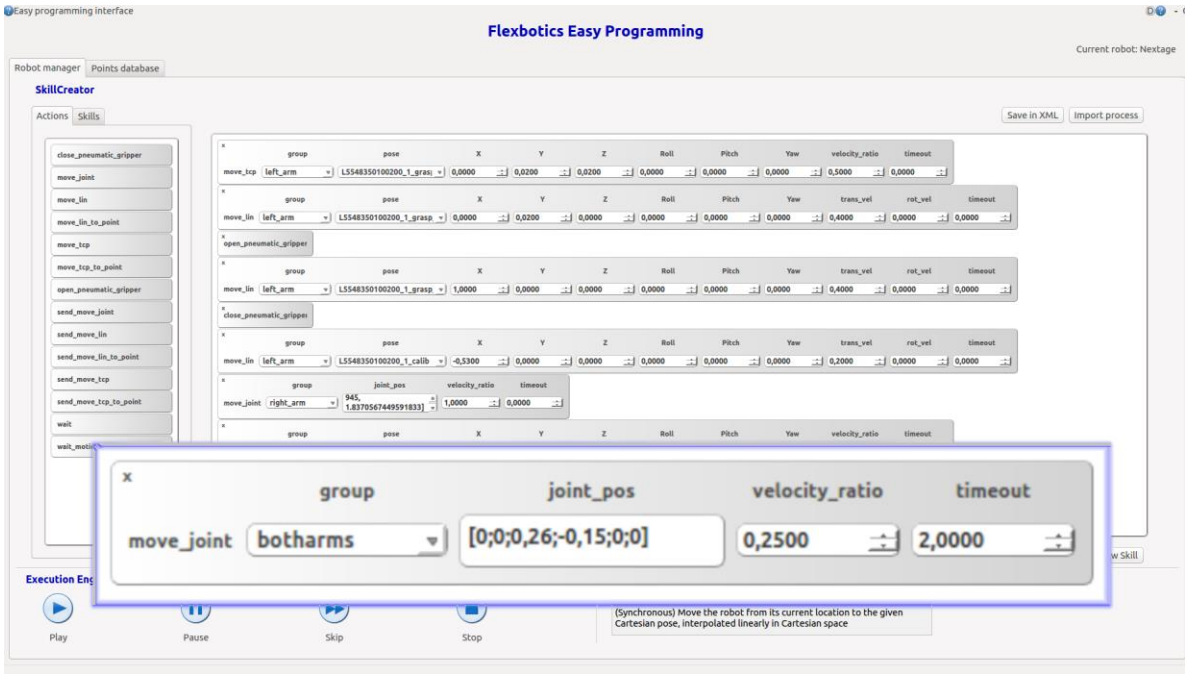


**Figure 2: Skill/Primitive parametrization**

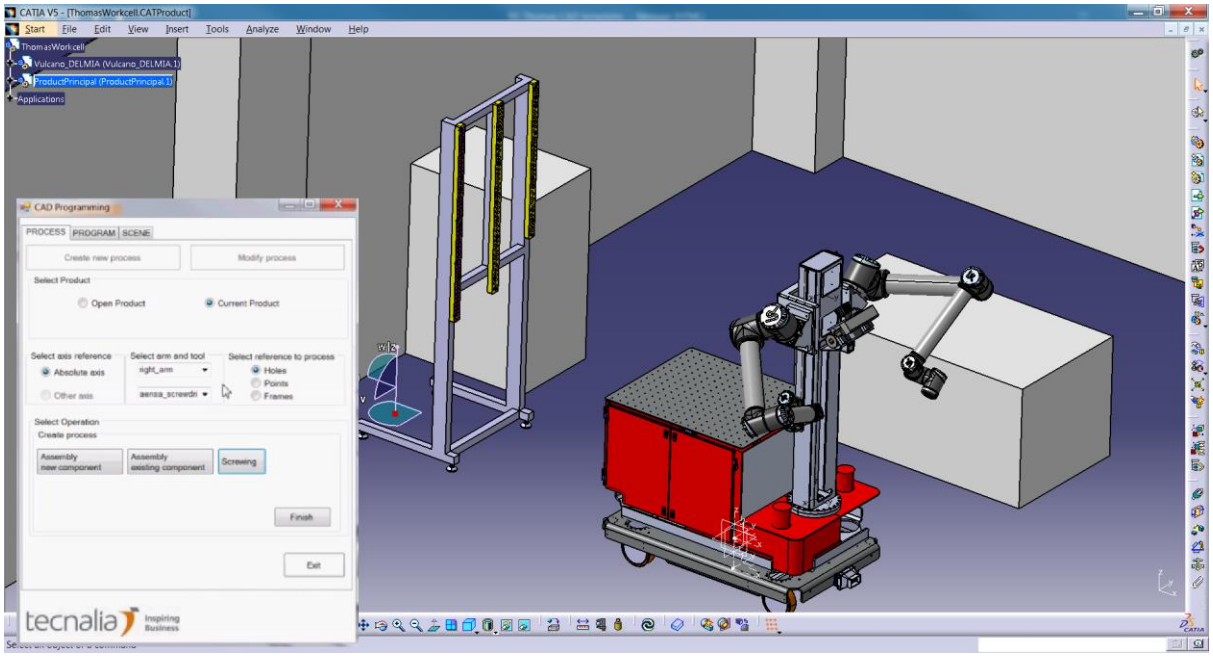


**Figure 3: CATIA scene containing all the involved elements in the process**

# 4. CATIA GUI FOR SKILL CONFIGURATION

## 4.1. 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 elements changes.

## 4.2. 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.
- 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 4), relevant positions, edges, etc.
- Process related information: grasping poses and assembly poses. Currently this information must be added manually following a process that requires specific training.
- Repetitive task parametrization taking advantage of previously prepared skill. Some skills can be prepared for being parametrized from CAD Based Programming.

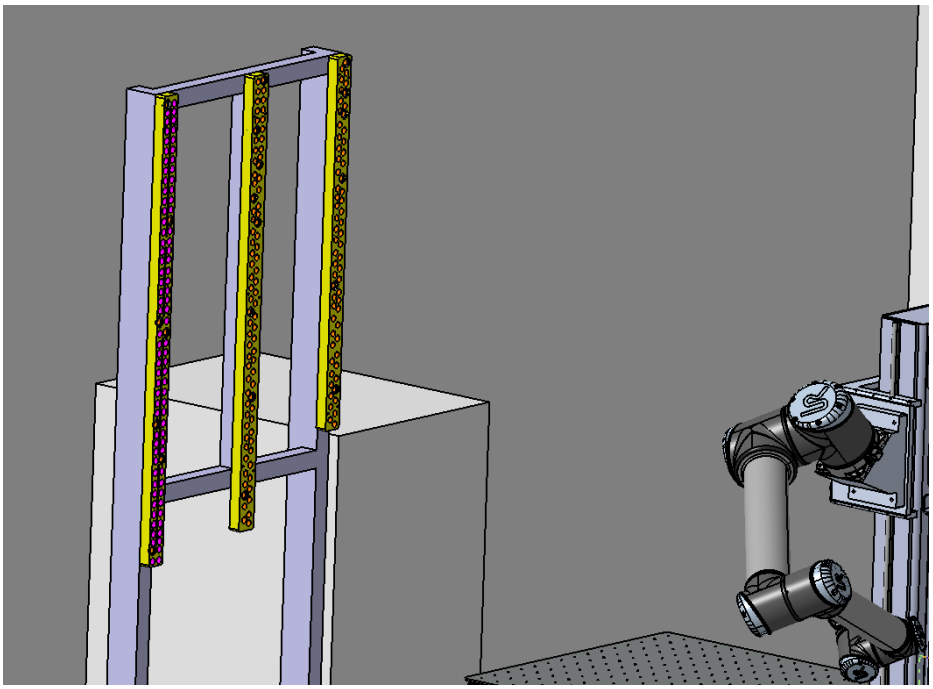

**Figure 4: Hole extraction from CATIA**

## 4.3. 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 file, the format which MoveIt! manages for 3D CAD 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 5.

```
<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 5: 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 6.

```
<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 6: Sequence of skill are stored in process files**

# 5. INTUITIVE PROGRAMMING THROUGH SKILL/PRIMITIVE SEQUENCING

## 5.1. 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 a 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 7) and Joint space (Figure 8). For the robots that have the capability of navigation a simple joy has been integrated for precise guiding (Figure 9).

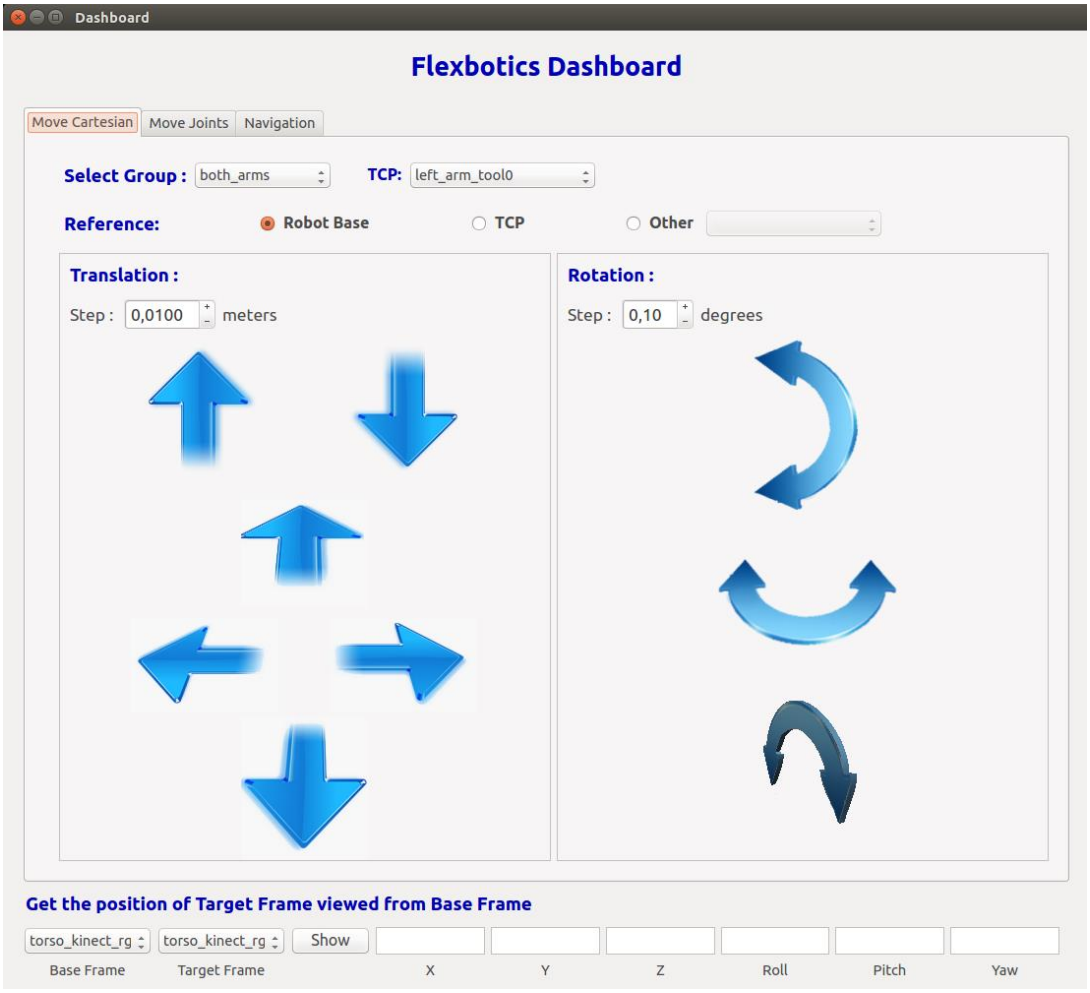


**Figure 7: GUI for Cartesian space**

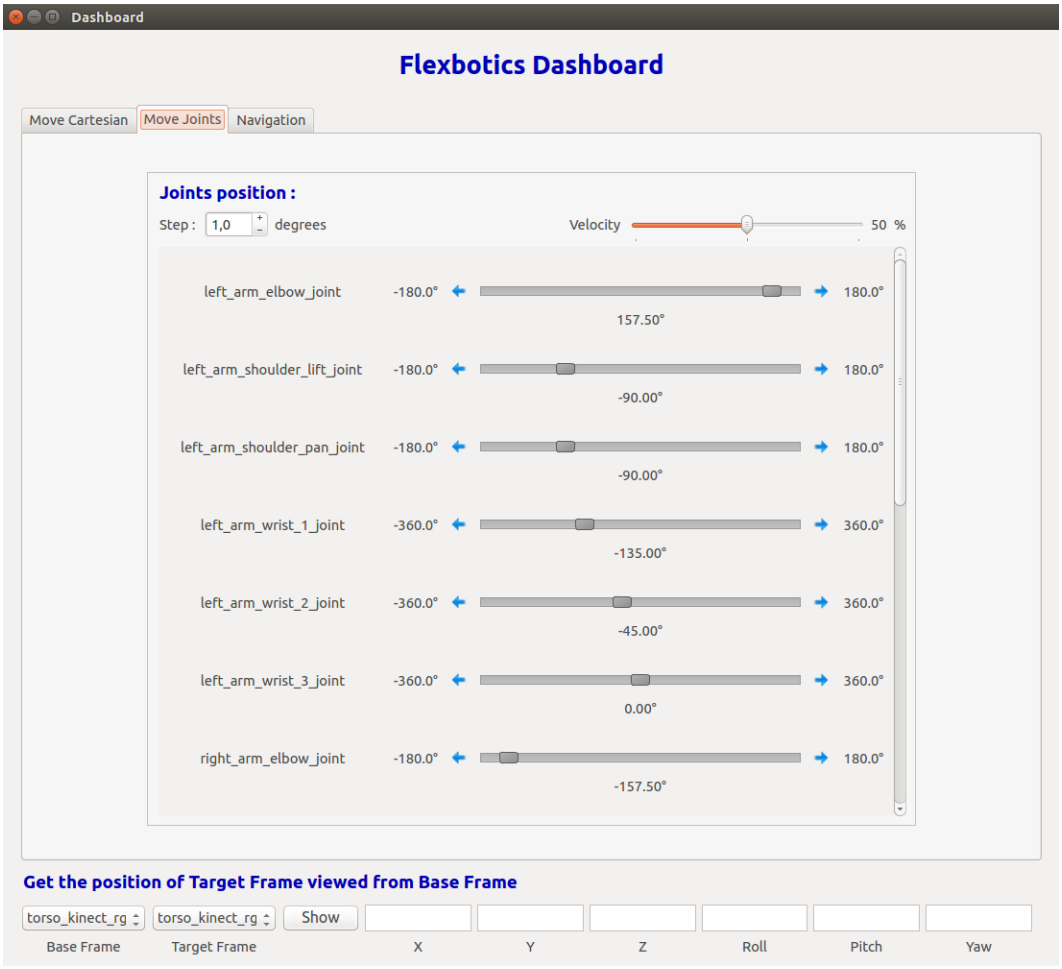


**Figure 8: GUI for Joint space**

## 5.2. 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 knowing as the available primitives for the Skill Based Programming Framework. As can be seen in Figure 10 in the left block previously mentioned primitives can be found. In the main panel a sequence of primitives has 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 11 shows how these 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 12 shows how the collapsed view generates an executable process file.

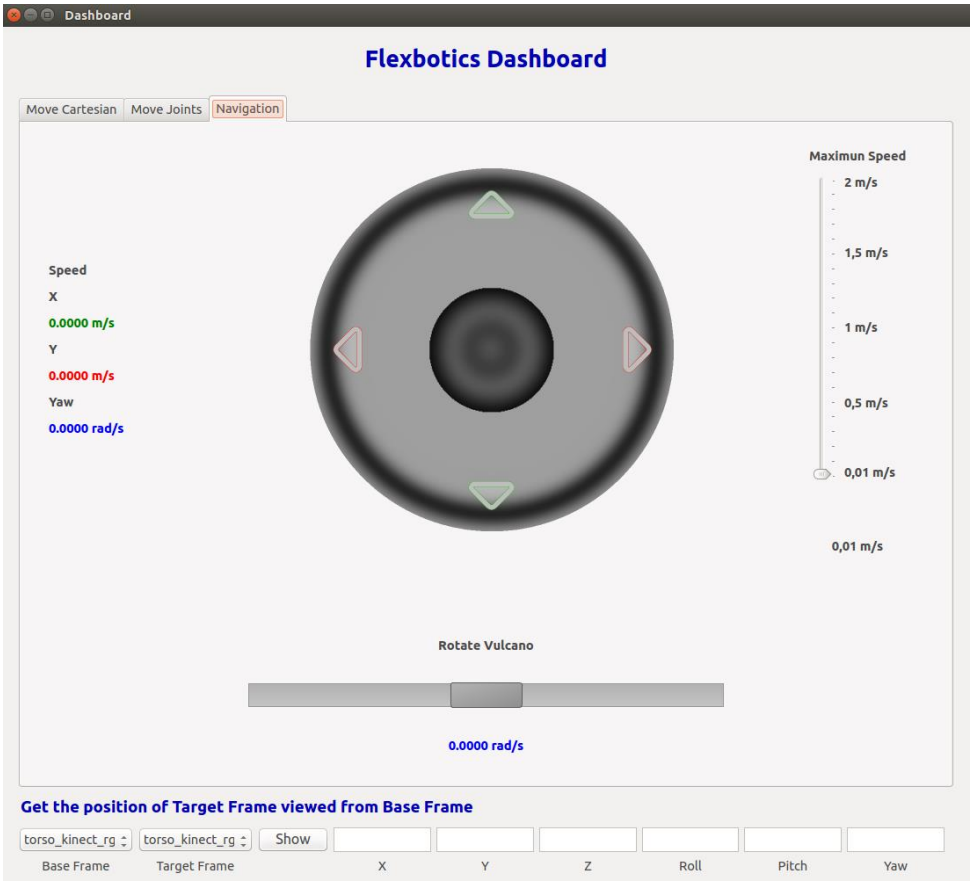


**Figure 9: GUI for the navigation**

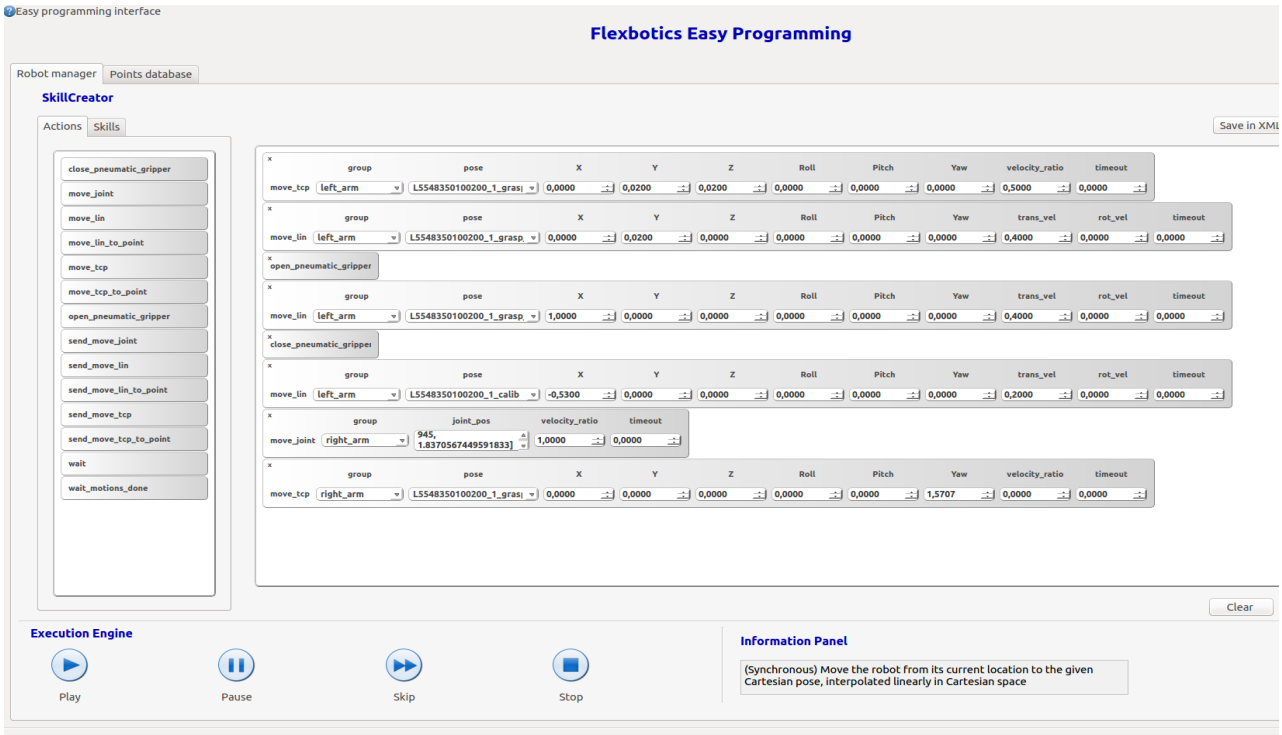


**Figure 10: Easy Programming GUI**

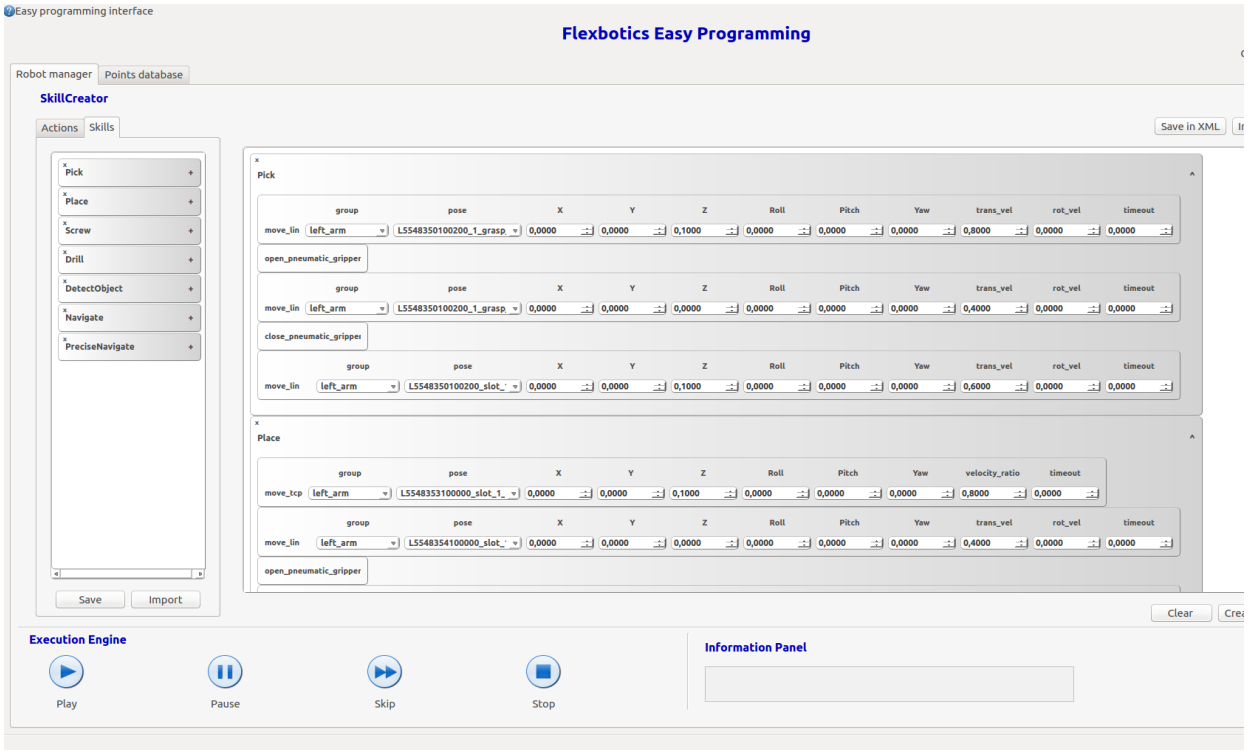


**Figure 11: New skill creation**

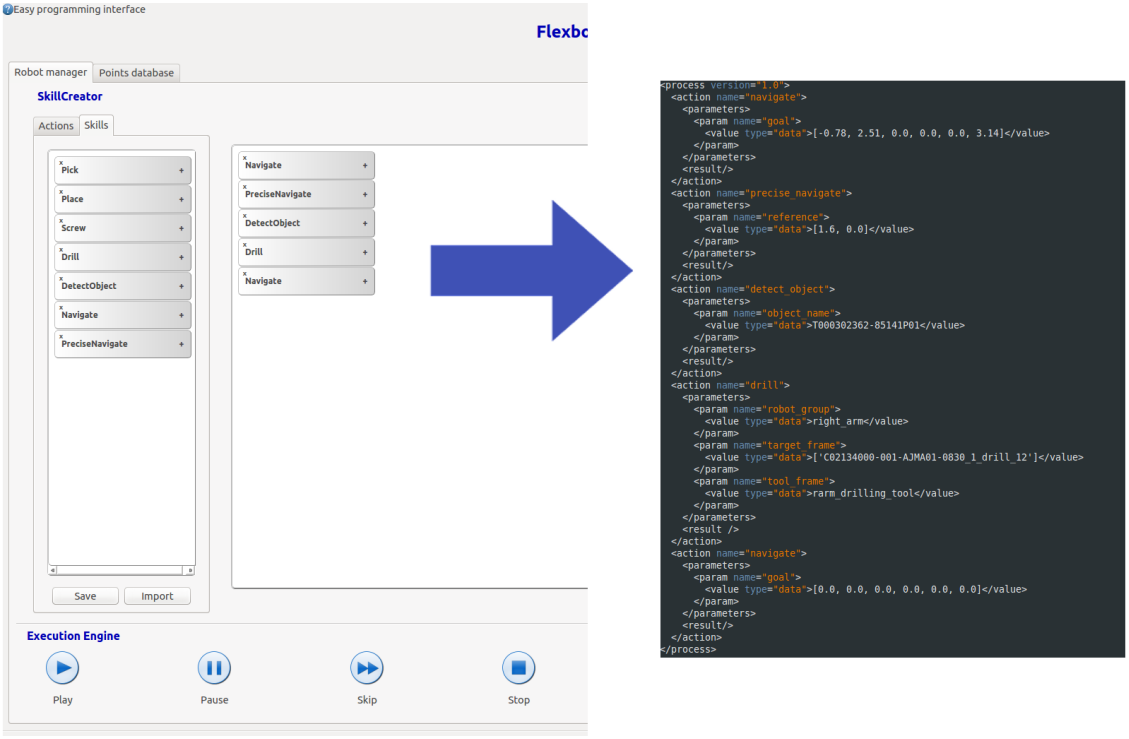


**Figure 12: Process generation from a sequence of skills**

# 6. CONCLUSIONS

This document summarizes the work done in WP4 – T4.1, which concerns about an easy task programming framework for the programming of THOMAS mobile robot systems industrial applications. Under this task, prototypes capable to provide intuitive tools for programming through skills and primitives have been developed and presented. The first prototype is a special GUI based on primitives and skills. Considering the sequence of them, this GUI exports process executable files for the mobile robot platform. Another prototype is being introduced through CATIA software using one GUI for the same purpose. In addition, CATIA GUI can be used as a tool for skill's configuration. In terms of easy handling of THOMAS mobile robot platform, one generic dashboard is being presented. Enabling both robot's manipulation in different ways (Cartesian and Joint space) and robot's navigation, the operator can easily control the MRP.

In the following phases of the project the user experience will be improved making more intuitive the linking of skill input/output parameters.

# 7. 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 |

## 8. REFERENCES

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