REMODEL - Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects

Deliverable 3.1 – CAD Platform Interface

Project acronym: REMODEL
Project title: Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects
Grant Agreement No.: 870133
ObjectsTopic: DT-FOF-12-2019
Call Identifier: H2020-NMBP-TR-IND-2018-2020
Type of Action: RIA
Project duration: 48 months
Project start date: 01/11/2019
Work Package: WP3
Lead Beneficiary: TAU
Authors: TAU, UNIBO, TECNALIA, PUT
Dissemination level: Public
Contractual delivery date: 28/02/2023
Actual delivery date: 27/02/2023
Project website address: https://remodel-project.eu/

Table of contents

1. Task description: T3.1 CAD Platform interface

Leader: **TAU**; Participants: **UNIBO, IEMA, TECNALIA, ELIMCO, TUM, PUT, ELVEZ, VW**

A software package to extract the information about the wiring sequence, cables routing and component nominal location (fitting) from the CAD design files will be provided by the WIRES experiment. This task will permit the extraction of the information about the wiring harness geometry and structure from the EPLAN Harness prod software and will be interfaced with the robot task planners and the User Interface. The possibility of selecting during the post-processing of the product design for the generation of the robot activity plan which part of the manufacturing can be executed by the robot will be considered. This selection can be eventually edited by means of the User Interface developed in T3.2.

To TRL 4: The ROS package available for switchgear design will be extended to extract the information about the wiring harness geometry and structure. The integration with the robot task planners and the User Interface will be carried out. Laboratory tests will be executed on a couple of design samples from all the use cases. The methodology will be extended to wiring harness manufacturing and assembly.

To TRL 5: Integration with the design chain carried out by the companies and validation against their design procedures. Execution of tests with new products along the whole design chain for the automatic generation of the robot task list considering the constraints and the interfaces of the CAD software.

To TRL 6: Demonstration of the CAD Platform interface package in the design procedure carried out by the industrial partners. Execution of tests considering timing and constraints of the real production environment.

2. INTRODUCTION

Task 3.1 aims to provide the REMODEL ROS system with a modular means to translate physical hardware, system, product, and process data into compatible digital formats, which can be extracted and used as required by the various subsystems of REMODEL. This module is a software component (composed of at least one ROS package) termed CAD Platform, as some of the input files from which this information is extracted are CAD design files (such as 2D drawings of the wiring harnesses configuration or 3D models of the assembly/mounting platforms). However, this package doesn't only deal with CAD files, it also manages information of other file types and formats about the properties of the cables, the assembly operations, the dimensions of tools and other elements of the platforms, cable connections, etc. This package processes all this information, constituting the basis for building the virtual world of the robot platforms in ROS and providing key hardware and process information at various stages of the use cases execution (where applicable).

The deliverable aims to summarize in a concise manner, the primary objectives of the CAD Platform and the common structure and functionalities taken into consideration during its development. It also focuses on the differences on the utilization of the generic system layout of the CAD platform in the different use cases, based on the formats of various data available/ provided by the use case owners; the different hardware elements to be modeled/ translated; and the variation in processes depending on the nature of the use case. The document also explains the implementation of the CAD Platform, along with a brief summary of existing usage. The above-mentioned information is sequentially documented in the subsequent sections of the report.

3. COMMON ARCHITECTURE

Three different CAD Platforms have been developed for the IEMA, ELIMCO, and ELVEZ use cases. They all share a common overall architecture, but there are differences between them as the processes, the platforms, and the handled components of each use case are disparate. Consequently, the data used to define the process and the system are different. Additionally, the formats of the information files provided by the companies are different in most cases, and the same happens with the outputs required by the REMODEL system to perform each of the use cases.

The following steps can be differentiated in the CAD Platform (see Figure 1):

- **Input files generation:** In some cases the input files for the CAD Platform ROS package require the use of additional software to preprocess or generate them. This additional software is also part of the CAD Platform module.
- **Information extraction:** Once the input files are prepared, their information is extracted and introduced into the ROS system. This applies to any kind of input file, whatever its format.

- **Information processing and storing:** The data extracted from all the input files are processed, merging their information whenever required, as some cases have different files containing complementary information. Then, the required useful information obtained as outcome, is stored using different structures (mostly dictionaries) that can be accessed in the next two steps.
- **ROS services definition:** A set of ROS service servers are defined to provide information about the physical hardware, handled materials, process, etc. These services can be called at any point from any other module of the REMODEL ROS system. Whenever any of these services are called, it gets the required information from the storing structures created in the previous step and provides it to the service client making the call.
- **RVIZ visualization:** The stored useful information is also accessed to visualize interesting information in RVIZ. For instance, **tf** frames to represent points of interest, **markers** to visualize the 3D mesh of the platform in which the robot is working, etc.

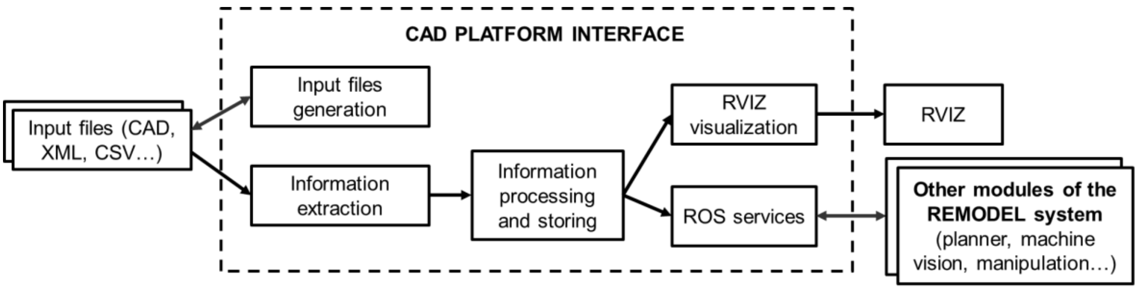


Figure 1. Block diagram of the CAD Platform

However, there are exceptions with the VWP and the ENKI use cases, as they do not use dedicated CAD files or other previously mentioned process/product descriptor files, so the structure of their CAD Platform's packages is simpler. Instead they use pre-existing proprietary information systems of a similar nature, which are described in this deliverable as well (Section 4.4 and 4.5). The objective of these systems is the same — to introduce information about the physical platform, the process, the manipulated components in the ROS system, and provide it to any other ROS nodes whenever it is required.

4. Use Case Specific Implementation

4.1. IEMA Use Case

The IEMA Use Case involves the automatization of the cabling operation of an electrical switchgear. The task, described through several files exported from a softwares for switchgear technical designs, is performed through three robotic manipulators that handles the insertion, screwing and routing operations for each connection in the production plan. In particular a switchgear is characterized by a high number of electromechanical components of different morphologies that require a high degree of flexibility both from the vision system to correctly perform the detection and from the manipulation system to handle the different kind of connectors and insertions. The REMODEL database for this use case, on the base of the general structure common to the other use cases, aims at providing an ordered list of all the operations to be performed to complete the use case task, together with a detailed description of each component involved to properly allow the dynamic planner (T5.1) to define the specific tools and subtask sequence required to properly handle each connection and routing.

The database information is based on several files produced from the project modeling software (Eplan for IEMA use case) which are defined as follows:

- **System Modelling:** a CAD file (X3D or WMRL format) allows a complete definition of the shape and positioning of all the components defining the switchgear. In particular, it provides a description of the main case, the supporting rails, and all the ordered components for each rail. Morphology and color description allows the production of a complete visualization of the switchgear through STL meshes as shown in Figure 2.

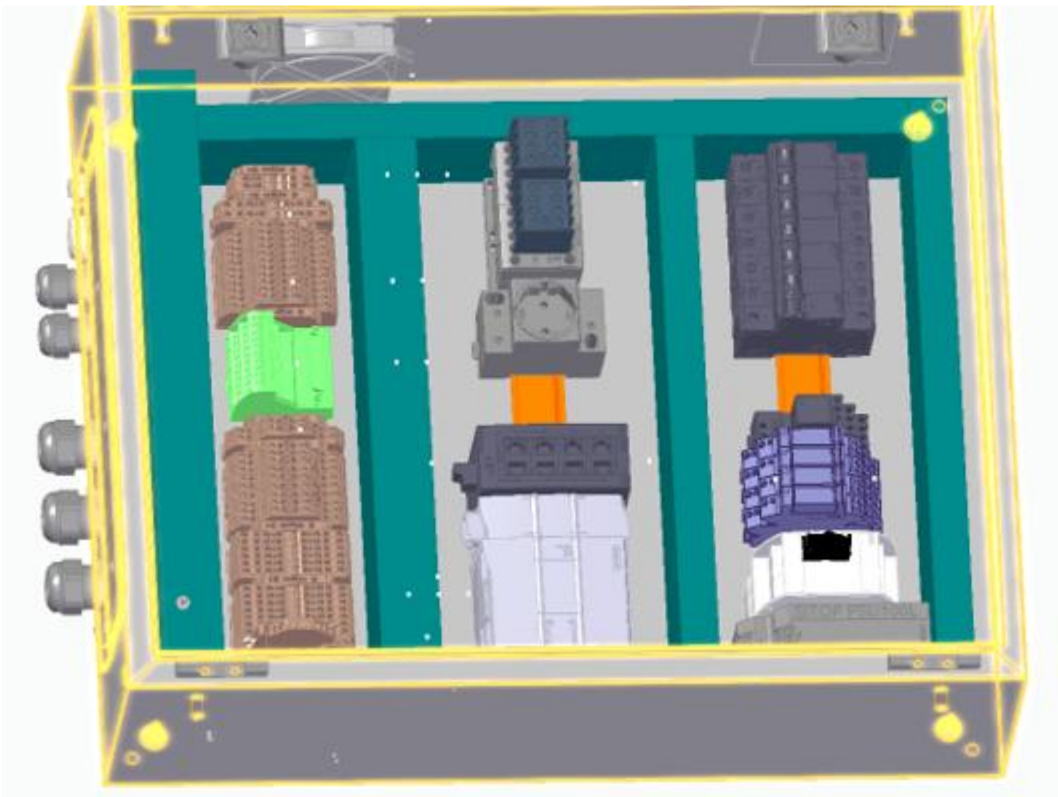


Figure 2. CAD representation of the IEMA switchgear

Since the morphology data are provided by the manufacturer as a general representation and the positions of the components are still prone to error due to their placement performed by hand, the spatial information provided by these files are rarely accurate and should be integrated with addional files and verified through a proper vision system.

- **Process description file:** a tabular file (CSV format) providing all the information concerning the task to be executed. For use case 1 the file provides the ordered list of connections to be performed with specifics on the type of cable and components involved in each cabling operation, along with the expected path as shown in Figure 3.

N. Pagina	Zona	Componente:pin	N. Pagina	Zona	Componente:pin	Numero filo	Sezione	Colore	Tipo pot.	
+1500/1.2	Q	-F1500.1:14	+1500/1.2	Q	-Q1500.1:A1	1500.51	0.75	BU	+	746
+1500/1.2	Q	-F1500.2:14	+1500/1.2	Q	-Q1500.2:A1	1500.53	0.75	BU	+	836
+1500/1.2	Q	-F1500.3:14	+1500/1.2	Q	-Q1500.3:A1	1500.55	0.75	BU	+	926
+1500/1.2	Q	-F1500.1:13	+1600/1.1	Q	-X1600:2:a	1500.50	0.75	BU	+	1055
+1500/1.2	Q	-F1500.2:13	+1600/1.1	Q	-X1600:3:a	1500.52	0.75	BU	+	1005
+1500/1.2	Q	-F1500.3:13	+1600/1.1	Q	-X1600:4:a	1500.54	0.75	BU	+	955
+0030/1.2	Q	-X0030.1:1L+.2:a2	+1600/1.1	Q	-X1600:1:a	1L+	1	BU	+	
+1500/1.2	Q	-Q1500.1:A2	+1500/1.2	Q	-Q1500.2:A2	0V	1	BU	-	320
+1500/1.2	Q	-Q1500.2:A2	+1500/1.2	Q	-Q1500.3:A2	0V	1	BU	-	320
+1500/1.1	Q	-Q1500.1:2	+1500/1.1	Q	-X1500:1:a	1500.04	1.5	BK	L	394
+1500/1.1	Q	-Q1500.1:4	+1500/1.1	Q	-X1500:2:a	1500.05	1.5	BK	L	391
+1500/1.1	Q	-Q1500.1:6	+1500/1.1	Q	-X1500:3:a	1500.06	1.5	BK	L	387
+1500/1.1	Q	-Q1500.2:2	+1500/1.1	Q	-X1500:5:a	1500.10	1.5	BK	L	370
+1500/1.1	Q	-Q1500.2:4	+1500/1.1	Q	-X1500:6:a	1500.11	1.5	BK	L	366
+1500/1.1	Q	-Q1500.2:6	+1500/1.1	Q	-X1500:7:a	1500.12	1.5	BK	L	363
+1500/1.1	Q	-Q1500.3:2	+1500/1.1	Q	-X1500:9:a	1500.16	1.5	BK	L	345
+1500/1.1	Q	-Q1500.3:4	+1500/1.1	Q	-X1500:10:a	1500.17	1.5	BK	L	341
+1500/1.1	Q	-Q1500.3:6	+1500/1.1	Q	-X1500:11:a	1500.18	1.5	BK	L	338
+1500/1.1	Q	-F1500.1:2	+1500/1.1	Q	-Q1500.1:1	1500.01	1.5	BK	L	258

Figure 3. Connection list data file example

- **Component characterization:** several data files (XML or CSV formats) providing detailed information on the component's characteristics (connectors type and location, terminal block characterization and components descriptions) as show in Figure 4.

=SSB-20B4	ZONE OPENED	IFM IGS232	1
=SSB-33CAM1	MODULO DI INTERFACCIA	33CAM1	1
=SSB-10F1	PILZ	PILZ.772000	1
=SSB-10F1	PILZ	6402292	1
=SSB-10F1	PILZ	6402293	1
=SSB-75G1	GUAINA D23	GUAINA D.23__66	5
=SSB-75G2	GUAINA D23	TUBO_ARIA_6/4_	10
=SSB-75G2	GUAINA D23	GUAINA D.23__66	8
=SSB-75G3	GUAINA D23	GUAINA D.23__66	8
=SSB-08H1	LAMPADA PRESENZA TENSIONE	6154492	1
=SSB-08H1	LAMPADA PRESENZA TENSIONE	6154374	1
=SSB-08H1	LAMPADA PRESENZA TENSIONE	6153805	1
=SSB-08H1	LAMPADA PRESENZA TENSIONE	6158742	1
=SSB-40I1	PANNELLO OPERATORE	6122780	1
=SSB-10KA1	LASER SUPPLY	SIE.3RH2122-1FB4	1
=SSB-10KA1	LASER SUPPLY	6503531	1
=SSB-10KA2	LASER SUPPLY	SIE.3RH2122-1FB4	1
=SSB-10KA2	LASER SUPPLY	6503531	1
=SSB-10KA3	LASER ENABLE	FIND.55349024000	1
=SSB-10KA3	LASER ENABLE	6400168	1

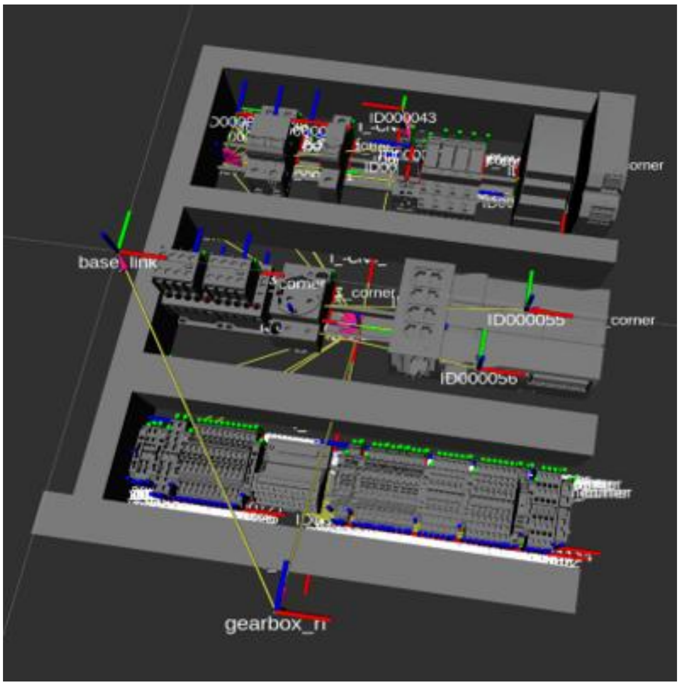
```
<data>
    <code name="SIEMENS 5SL4202-8" commercial="SIEMENS
5SL4202-8" xdim="0.0898" ydim="0.0359" zdim="0.07448" >
        <couple name="1">
            <subitem name="screw">
                <pos name="x">0.0085</pos>
                <pos name="y">0.009</pos>
                <pos name="z">0.04496</pos>
                <rot name="R">0</rot>
                <rot name="P">0</rot>
                <rot name="Y">0</rot>
            </subitem>
            <subitem name="hole">
                <pos name="x">0</pos>
                <pos name="y">0</pos>
                <pos name="z">0</pos>
                <rot name="R">0</rot>
                <rot name="P">0</rot>
                <rot name="Y">0</rot>
            </subitem>
        </couple>
        <couple name="2">
            <subitem name="screw">
```

Figure 4. Component characterization files examples (CSV on the left, XML on the right)

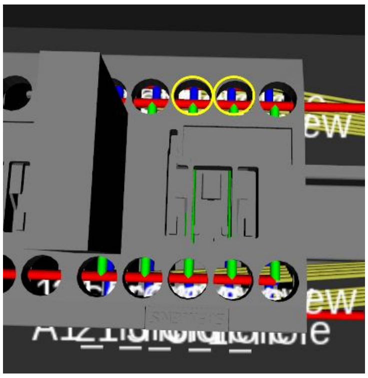
- **Semantic interconnection description:** a tabular file (WRI format) matching together the different nomenclatures concerning the same component in the different files as shown in Figure 5.

Cad ID	Label	Commercial
ID000041	X00005.2	6013277
ID000045	F1500.1	6503372
ID000046	F1500.1	6503373
ID000047	F1500.2	6503373

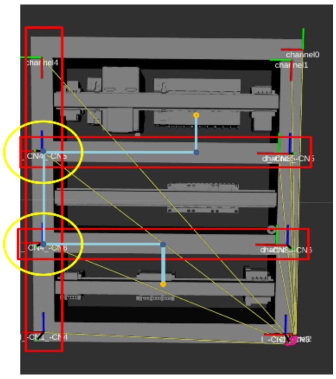
Figure 5. Semantic description WRI file table example



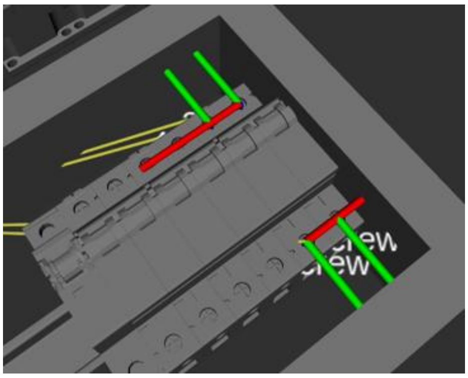
a.Remodel Switchgear



b. Connectors characterization



c. Cable routing path generation



d. Connector approach description

Figure 6. tfs and Markers of the IEMA switchgear, published by the CAD Platform

The system extracts all data in different dictionaries to produce a complete product database able to provide all the information required for the task through ROS services as summarized in Figure 9.

By inheriting from a baseline class describing a generic component all subsystems composing the switchgear are characterized as objects of different classes according to their characteristics. By exploiting the semantic description file (WRI) CAD data are enhanced by the additional characterization provided through commercial component descriptor files.

```
<Transform DEF='ID000038'  translation='59.1 22.675 73' rotation='-1 -0 -0 1.5708'>
  <Shape>
    <Appearance>
      <Material diffuseColor='0.360784 0.380392 0.431373'>
      </Material>
    </Appearance>
    <IndexedFaceSet coordIndex='378 379 359 -1 383 362 359 -1 383 382 362 -1 382 386 38
      <Coordinate point='3.9 14.1 -65.2117 3.9 15.7616 -65.2116 13.9 15.7616 -65.2116 1
      </Coordinate>
    </IndexedFaceSet>
  </Shape>
  <Shape>
    <Appearance>
      <Material diffuseColor='0.360784 0.380392 0.431373'>
      </Material>
    </Appearance>
    <IndexedFaceSet coordIndex='57 58 56 -1 60 61 62 -1 63 61 60 -1 63 64 61 -1 65 64 6
```

a. X3D CAD file

```
ID000036    =-BD3   WEI.0236500000
ID000037    =-05U2  6032226
ID000038    =-05Q2  6508704
ID000039    =-05Q1  5SL42028
ID000040    =-05Q3  5SL42038
ID000041    =-5Q12  SIE.5SY51027
ID000042    =-10KA3 6400168
ID000043    =-10KA3 FIND553490240000
ID000044    =-44KA1 FIND381170240050
ID000045    =-44KA2 FIND381170240050
ID000046    =-44KA3 FIND381170240050
```

b. CSV file with IDs

```
<code name="SIEMENS 5SL4204-8" commercial="6508704" xdim="0.035"
    <couple name="1">
        <subitem name="screw">
            <pos name="x">0.009</pos>
            <pos name="y">0.0815</pos>
            <pos name="z">0.050</pos>
            <rot name="R">3.141593</rot>
            <rot name="P">0</rot>
            <rot name="Y">3.141593</rot>
            <dat name="depth">0.01</dat>
            <dat name="diameter">0.007</dat>
            <dat name="type">0</dat>
        </subitem>
        <subitem name="hole">
            <pos name="x">0.009</pos>
            <pos name="y">0.0895</pos>
```

c. XML file with the models' description

Figure 7. Merging process of cross referenced data

Moreover by cross referencing the information about the sequences of operation to perform and the component files advanced morphological information are produced as the routing path keypoints, the sequence of components along a rail for component detection and the point cluster and occludency grid due to the presence of other cables for all connection points in the components.

A database class groups together all the objects characterizing the switchgear and handles the information query from external nodes.

A common trasform tree structure is kept among all objects to keep track of their relative position and to be able to map each object position with respect to the task world reference frame.

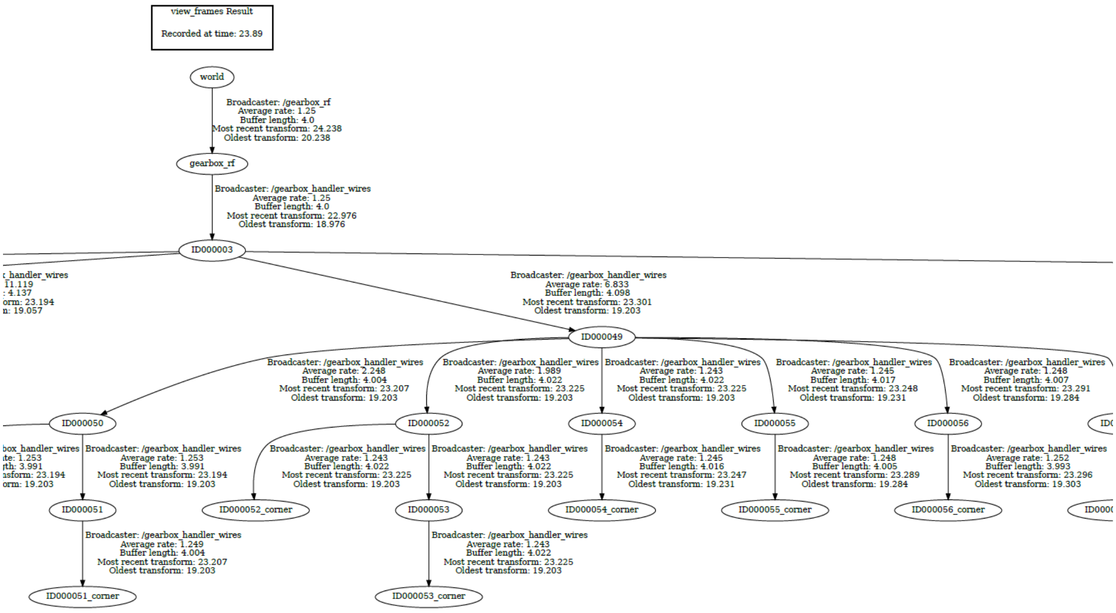


Figure 8. Hierarchical transform tree between the components of the switchgear

Eventually a dedicated service class acts as a bridge between the database and the ROS network defining all the service routines and providing to the planner the required informations.

Table 1. IEMA CAD Platform ROS services

Category	Service name	Description and applications	Input (request)	Output (response)
Component database	/gearbox_handler/cad_service	Provides morphological informations about a component. Used to get a complete description of the system. Active for analysis also if process informations are missing.	• label (string)	• Success (bool) • Target_pose (Pose) • Corner_pose (Pose) • Original_pose (Pose) • Dimensions (Vector3) • Pins (cad_pins[])
	/gearbox_handler/full_cad_service	Provides functional information about a component. This is useful to understand the expected motions for collision checking in the component neighborhoud.	• label (string)	• Success (bool) • Target_pose (Pose) • Corner_pose (Pose) • Original_pose (Pose) • Dimensions (Vector3) • Connection (cad_connection[])
	/gearbox_handler/get_item_screw_service	Provides information about component connectors given their detected position. Useful to perform the cable insertion.	• label (string) • cad_pose (Pose)	• Success (bool) • Pins (string[]) • screws (Pose[]) • holes (Pose[]) • screw_data (pyn_type[]) • hole_data (pyn_type[])

Category	Service name	Description and applications	Input (request)	Output (response)
	/gearbox_handler/ get_terminal_data	Provides specific information about terminal blocks.	• label (string)	• Success (bool) • Pins (string[]) • Terminal_start (Pose) • screws (Pose[]) • holes (Pose[]) • length (float) • height (float)
Process database	/gearbox_handler/ full_connection_ provider	Provides information about the next routing operation on the list. Used by the system planner to determine which functions to call.	• Occupancy (bool)	• End (bool) • Missing_pin_data (bool) • List_position (int) • source (string) • label (string) • pin (string) • direction (bool) • component_pose (Pose)
	/gearbox_handler/ group_connection _provider	Provides information about the next routing operation enriched by data on all components involved.	None	• End (bool) • Missing_pin_data (string) • List_position (int) • Origin (cad_pins) • Destination (cad_pins) • Wire (string) • Wire_path (string) • Path_keypoints (Pose[])
	/gearbox_handler/ reset_connection_ list	Resets the operation sequence to the starting point	None	• Success (bool)
	/gearbox_handler/ reset_to_ connection_list	Resets the operation sequence to the requested lines	• Data (float) • Source (string)	• Output (string)
Detection database	/gearbox_handler/ rail_scanner	Provides the characteristics of all the components along a single rail of the switchgear for detection purposes	None	• Rail_start (Pose) • Component_poses (Pose[]) • Components_dimensions (Vector3[]) • Components_label (string[]) • End (bool)
	/gearbox_handler/ reset_rail_scanner	Resets the rail list	None	• Success (bool)
	/gearbox_handler/ cluster_service	Provides the position of all connectors of a component grouped as horizontal or vertical cluster of points for detection purposes.	• Label (string)	• Horizontal (cluster[]) • Vertical (cluster[])

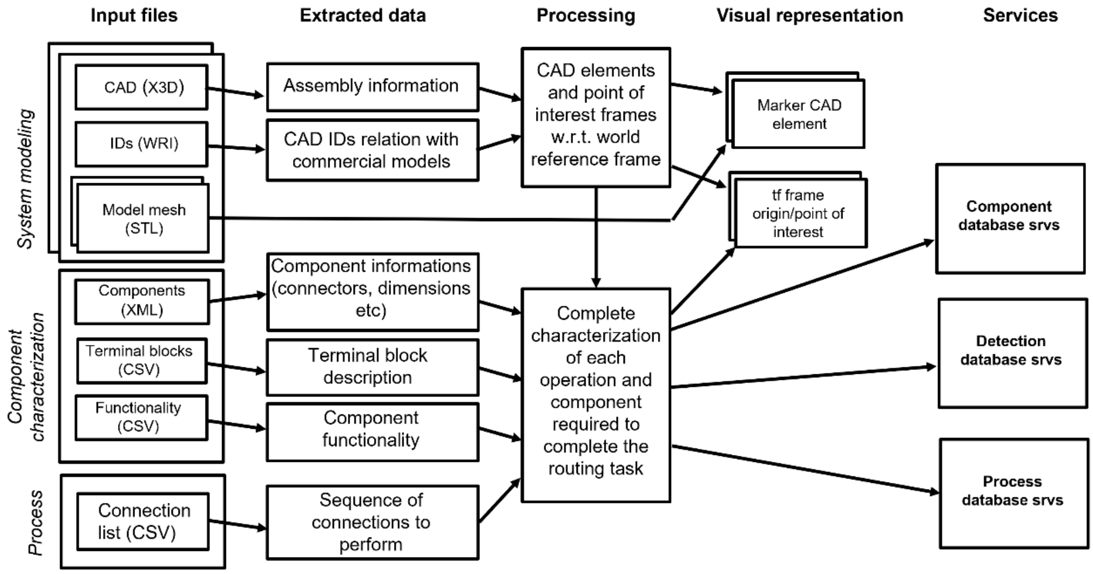


Figure 9. Database evaluation structure

4.2. ELIMCO Use Case

In ELIMCO's use case, a dual-arm robot manufactures medium to large-size wiring harnesses for the aeronautic sector. The key point of the posed scenario is the use of CAD drawings of the wiring harnesses as input for the robotic system to automatize the robot program generation. In order to achieve this goal, a developed custom application allows the generation of user-defined wiring routes from 2D planes, including additional process information such as pin placement on the workbench. A ROS node has also been developed for the management of this extracted information, a node that executes high-level skills that adapts the robot's behavior based on the defined routes.

ELIMCO uses CATIA software to create 2D drawings, the standard CAD software in the aeronautical sector. To create a functional approach for the current ELIMCO's workflow, it was decided to develop software able to interact with CATIA and extract process information from the CAD drawing. The application has been developed in Visual Studio, offering an interface that assists the user in creating cable routes by easily clicking on the CATIA 2D drawing. This application will generate XML files that will be loaded afterward by ROS to generate robot trajectories automatically.

As stated previously, the initial point of ELIMCO's workflow is a CATIA file with the 2D drawing of the wiring and an Excel file with the information on each cable. Both files are currently used in the manual assembly process at ELIMCO. The developed Visual Studio application is able to load and manage these files, offering the next functionalities to the user:

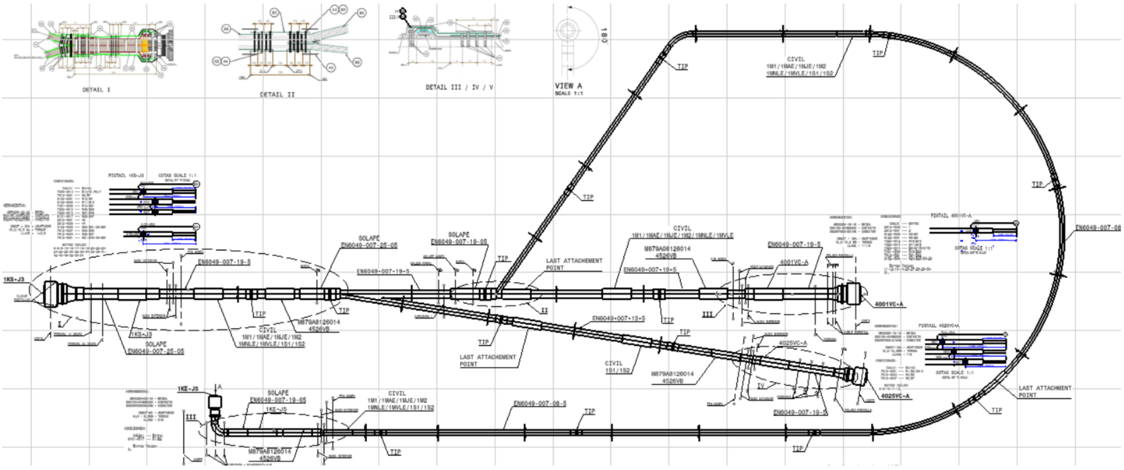
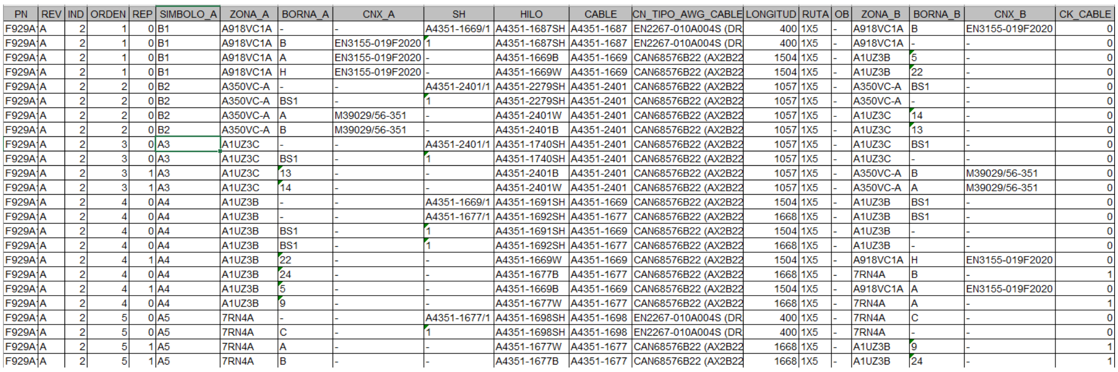


Figure 10. CATIA drawing of ELIMCO



PN	REV	IND	ORDEN	REP	SIMBOLO_A	ZONA_A	BORNA_A	CNX_A	SH	HILO	CABLE	CN_TIPO_AWG_CABLE	LONGITUD	RUTA	OB	ZONA_B	BORNA_B	CNX_B	CK_CABLE
F929A	A	2	1	0	B1	A918VC1A	-	-	A4351-1669/1	A4351-1687SH	A4351-1687	EN2267-010A004S (DR	400	1X5	-	A918VC1A	B	EN3155-019F2020	0
F929A	A	2	1	0	B1	A918VC1A	B	EN3155-019F2020	1	A4351-1687SH	A4351-1687	EN2267-010A004S (DR	400	1X5	-	A918VC1A	-	-	0
F929A	A	2	1	0	B1	A918VC1A	A	EN3155-019F2020	-	A4351-1669B	A4351-1669	CAN68576B22 (AX2B22	1504	1X5	-	A1UZ3B	5	-	0
F929A	A	2	1	0	B1	A918VC1A	H	EN3155-019F2020	-	A4351-1669W	A4351-1669	CAN68576B22 (AX2B22	1504	1X5	-	A1UZ3B	22	-	0
F929A	A	2	2	0	B2	A350VC-A	-	-	A4351-2401/1	A4351-2279SH	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A350VC-A	BS1	-	0
F929A	A	2	2	0	B2	A350VC-A	BS1	-	1	A4351-2279SH	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A350VC-A	-	-	0
F929A	A	2	2	0	B2	A350VC-A	A	M39029/56-351	-	A4351-2401W	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A1UZ3C	14	-	0
F929A	A	2	2	0	B2	A350VC-A	B	M39029/56-351	-	A4351-2401B	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A1UZ3C	13	-	0
F929A	A	2	3	0	A3	A1UZ3C	-	-	A4351-2401/1	A4351-1740SH	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A1UZ3C	BS1	-	0
F929A	A	2	3	0	A3	A1UZ3C	BS1	-	1	A4351-1740SH	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A1UZ3C	-	-	0
F929A	A	2	3	1	A3	A1UZ3C	13	-	-	A4351-2401B	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A350VC-A	B	M39029/56-351	0
F929A	A	2	3	1	A3	A1UZ3C	14	-	-	A4351-2401W	A4351-2401	CAN68576B22 (AX2B22	1057	1X5	-	A350VC-A	A	M39029/56-351	0
F929A	A	2	4	0	A4	A1UZ3B	-	-	A4351-1669/1	A4351-1691SH	A4351-1669	CAN68576B22 (AX2B22	1504	1X5	-	A1UZ3B	BS1	-	0
F929A	A	2	4	0	A4	A1UZ3B	-	-	A4351-1677/1	A4351-1692SH	A4351-1677	CAN68576B22 (AX2B22	1668	1X5	-	A1UZ3B	BS1	-	0
F929A	A	2	4	0	A4	A1UZ3B	BS1	-	1	A4351-1691SH	A4351-1669	CAN68576B22 (AX2B22	1504	1X5	-	A1UZ3B	-	-	0
F929A	A	2	4	0	A4	A1UZ3B	BS1	-	1	A4351-1692SH	A4351-1677	CAN68576B22 (AX2B22	1668	1X5	-	A1UZ3B	-	-	0
F929A	A	2	4	1	A4	A1UZ3B	22	-	-	A4351-1669W	A4351-1669	CAN68576B22 (AX2B22	1504	1X5	-	A918VC1A	H	EN3155-019F2020	0
F929A	A	2	4	0	A4	A1UZ3B	24	-	-	A4351-1677B	A4351-1677	CAN68576B22 (AX2B22	1668	1X5	-	7RN4A	B	-	1
F929A	A	2	4	1	A4	A1UZ3B	5	-	-	A4351-1669B	A4351-1669	CAN68576B22 (AX2B22	1504	1X5	-	A918VC1A	A	EN3155-019F2020	0
F929A	A	2	4	0	A4	A1UZ3B	9	-	-	A4351-1677W	A4351-1677	CAN68576B22 (AX2B22	1668	1X5	-	7RN4A	A	-	1
F929A	A	2	5	0	A5	7RN4A	-	-	A4351-1677/1	A4351-1698SH	A4351-1698	EN2267-010A004S (DR	400	1X5	-	7RN4A	C	-	0
F929A	A	2	5	0	A5	7RN4A	C	-	1	A4351-1698SH	A4351-1698	EN2267-010A004S (DR	400	1X5	-	7RN4A	-	-	0
F929A	A	2	5	1	A5	7RN4A	A	-	-	A4351-1677W	A4351-1677	CAN68576B22 (AX2B22	1668	1X5	-	A1UZ3B	9	-	1
F929A	A	2	5	1	A5	7RN4A	B	-	-	A4351-1677B	A4351-1677	CAN68576B22 (AX2B22	1668	1X5	-	A1UZ3B	24	-	1

Figure 11. Excel with cable information

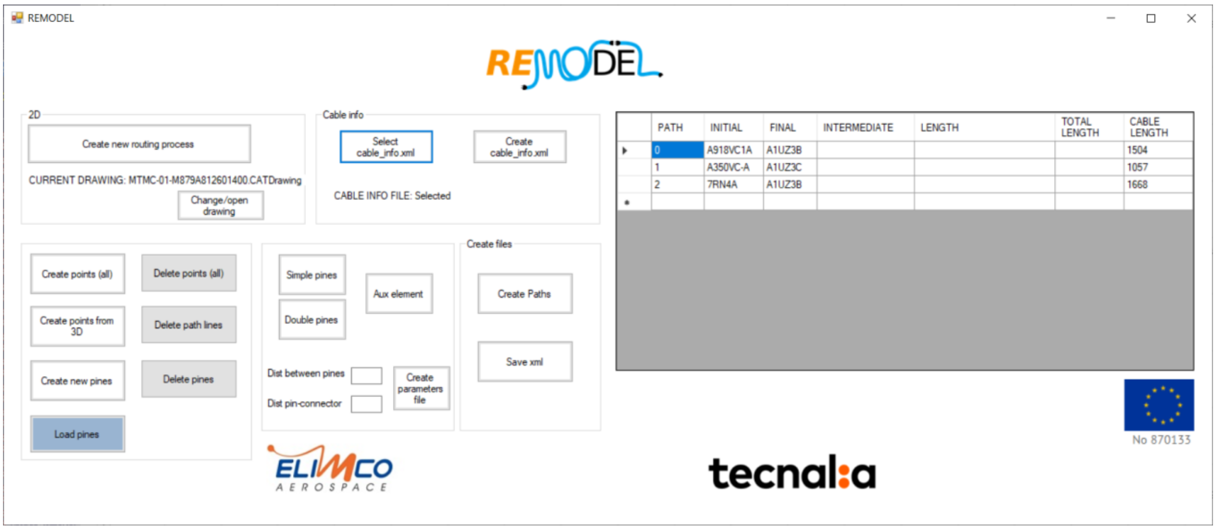


Figure 12 - Visual Studio application for ELIMCO's use case

- **Create new routing process:** The application reads the file opened in CATIA. It is possible to change the CAD model using the "Change/open drawing" button.

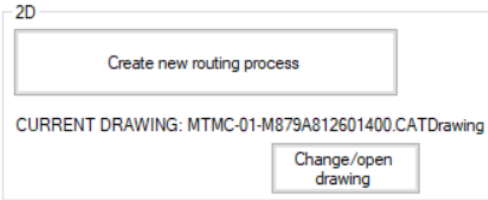


Figure 13. Create new routing process step

- **Cable info:** In this section, the user can choose between two possibilities; 1) select the XML file with the cable information, created beforehand with the application, or 2) create a new file based on the input Excel file. The XML file contains the most relevant information extracted from Excel.

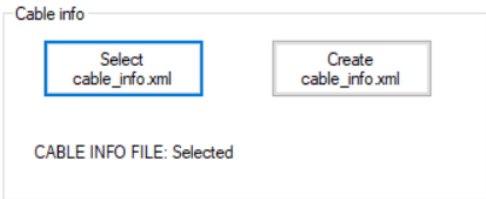


Figure 14. Cable info step

Once the XML is selected, the information for each path is automatically loaded into the interface.

	PATH	INITIAL	FINAL	INTERMEDIATE	LENGTH	TOTAL LENGTH	CABLE LENGTH
▶	0	A918VC1A	A1UZ3B				1504
	1	A350VC-A	A1UZ3C				1057
	2	7RN4A	A1UZ3B				1668
*							

Figure 15. Path information loaded in the interface

- Creation and positioning of pins (physical pins used to guide the wires) in the drawing to create the routes on the workbench.

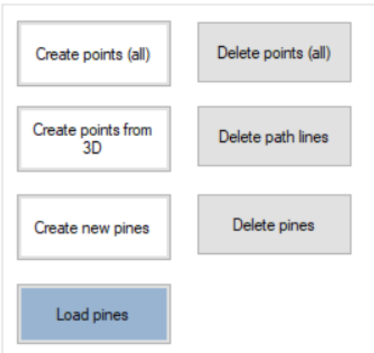


Figure 16. Creation of pins step

This functionality allows for creating the points where the pins will be placed, offering different options to the user. It is possible to create points using the 3D model where the initial and final points of each path are automatically generated and positioned. It is also possible to create them manually or load the pins from a previous process. These points are the nominal positions of the pins that the robot will place on the workbench, creating different wiring routes along the table. In addition, the different points created are distinguished by colors to help the user identify them more easily. Red points identify the initial points of the routes, blue points indicate the intermediate route points and finally, the green ones indicate the end point of the different routes.

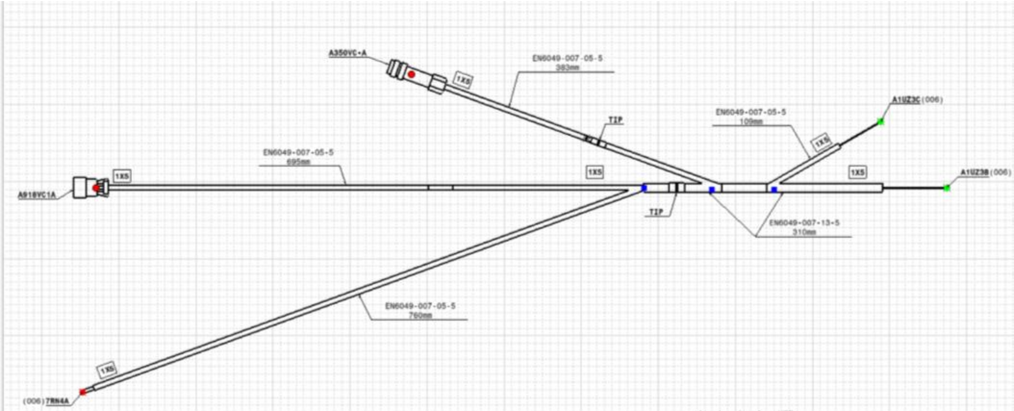


Figure 17. Pin placement interface

- It is possible to modify the properties of the defined positions and add auxiliary information about the pins (e.g. define simple or double pins, the distance between pins in double pin configuration, and the distance between the connector and first pin...). This information will add parameters to the pin placement ROS skill of the robot.

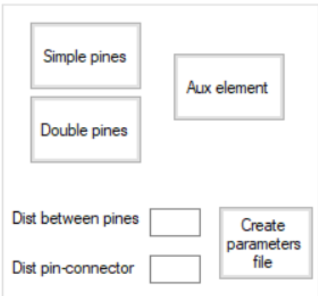


Figure 18. Pins information step

- Generate routes: The next step is the definition of different routes based on the different pins defined in the drawing. Once a route is selected on the list, the application will ask the user to click the intermediate points. The created routes are displayed in CATIA for user verification.

	PATH	INITIAL	FINAL	INTERMEDIATE	LENGTH	TOTAL LENGTH	CABLE LENGTH
	0	A918VC1A	A1UZ3B				1504
	1	A350VC-A	A1UZ3C				1057
▸	2	7RN4A	A1UZ3B	[pin_1_doble]	7RN4A - pin_1_doble: 775 pin_1_doble - A1UZ3B: 393	1168	1668
*							

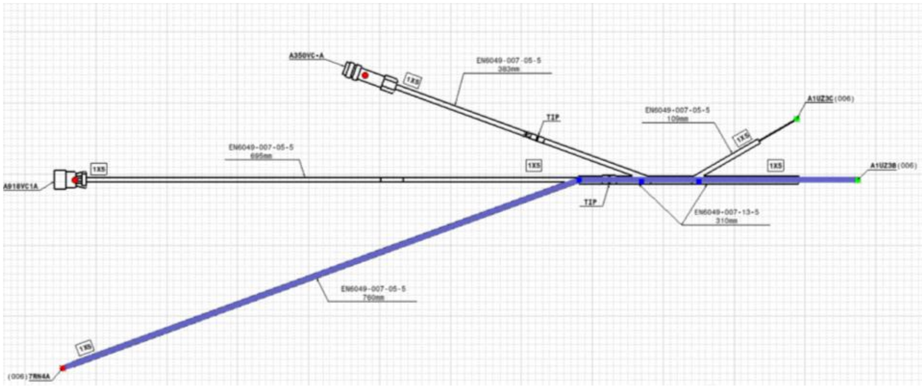


Figure 19. Routes generation step. Top: routes information visualization, Bottom: routes visualization in drawing

- Finally, when the paths and routes are created, the user can save the information in different XML files: pin_info.xml, cable_info.xml, and path_info.xml. These files are the input for the ROS node developed for the management of this extracted information.

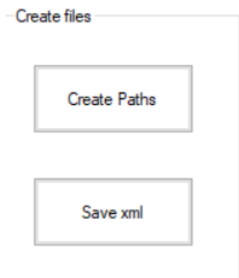


Figure 20. Files generation step

An additional ROS node has also been developed, Elimco CAD Information Manager node, to make all this information available for the ROS framework. The main purpose of the node is to load the XML files generated in CATIA and offer all the geometrical information through ROS services. Additionally, the node also offers the capability to carry out some geometrical calculations on the data to provide the pin and route positions in a different coordinate system as in many cases the CAD files' reference frame is on the plane's coordinate system, meters away from the pin positions.

Specifically, the ROS node offers the next services:

- **Load XML files:** Service to load XML files dynamically, allowing an online application change.

- **Get pin positions:** Service to retrieve the list of pin positions on the workbench. There are some parameters to define if it is necessary to calculate the centroid as the new reference frame, or to define the distance between pins and connectors.
- **Display pins in tf:** Display the position of the pins in tf for verification purposes. This can be seen in Figure 21.
- **Get cable list:** Retrieve the list of all the cables of the current wiring harness.
- **Get path positions:** Get the list of path positions of a given cable.

These services offer all the CAD information to the different modules developed within workpackages WP3 and WP5, providing geometrical information in a simple ROS interface. The whole CAD Platform functionality is shown graphically in Figure 22.

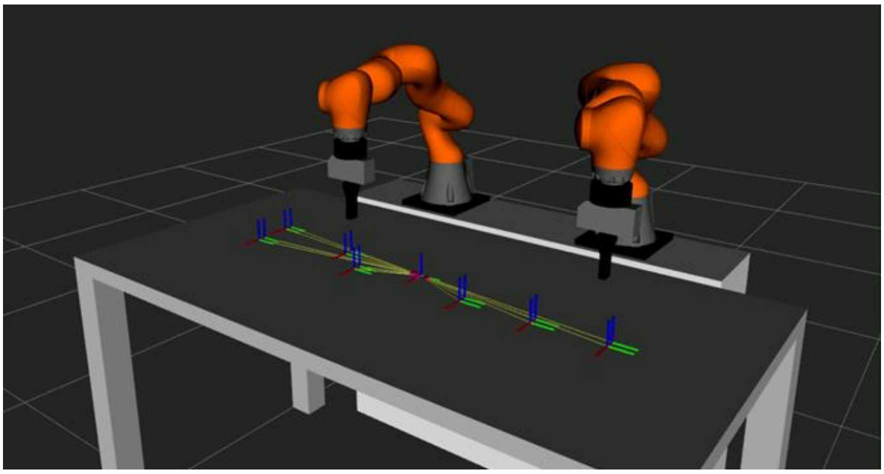


Figure 21. Workbench layout and pins tfs displayed in RVIZ

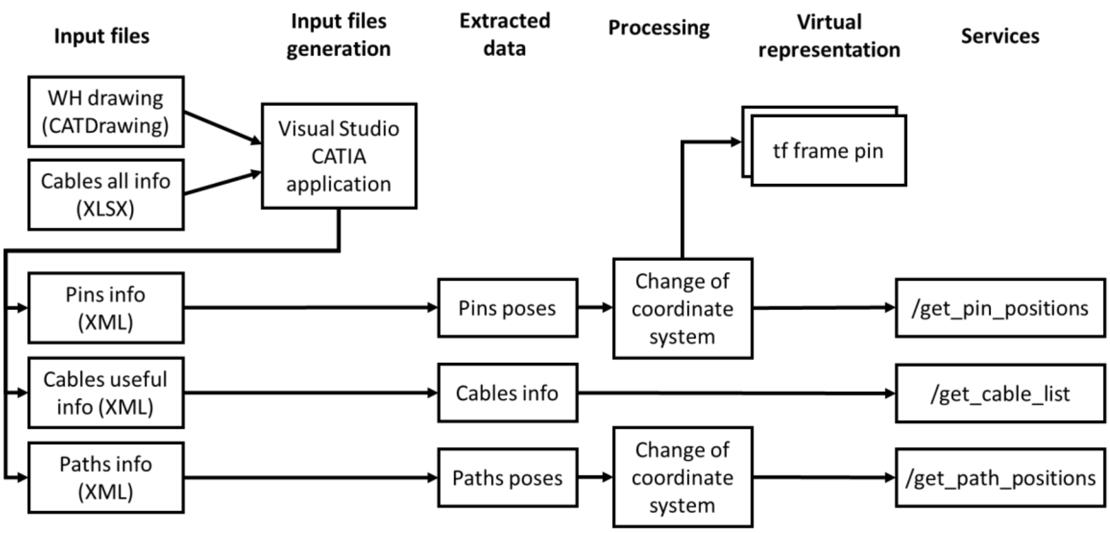


Figure 22. ELIMCO CAD Platform functionality block diagram

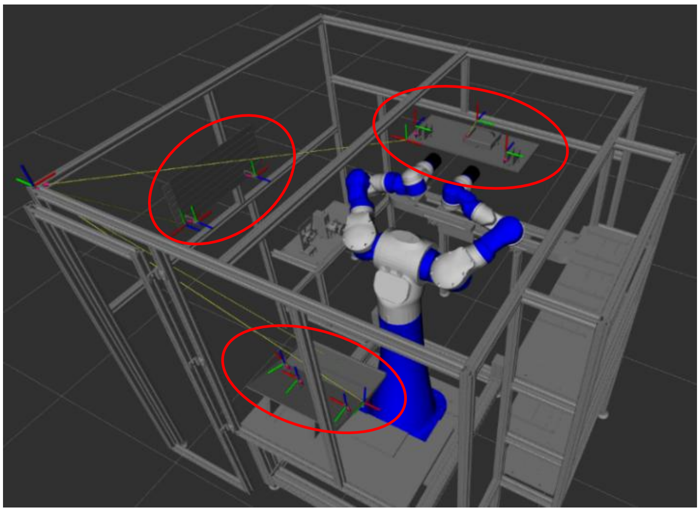
4.3. ELVEZ Use Case

The ELVEZ use case entails the automatic assembly of three complex wire harness, composed of six, ten and eleven cables of different colors and sizes. To perform this assembly the different cables of the wire harnesses have to be routed through a set of guides following a certain distribution, and then, they have to be taped together in several spots. To achieve this, a dual-arm robot is used, equipped with two parallel grippers and an automatic taping gun. These tools can be switched with the help of an Automatic Tool Changer (ATC). Additionally, a redesigned assembly platform, provisioned with guides, is used to facilitate the correct cables distribution. In order to make the system highly reconfigurable, all the actions of the robot are parameterized and depend on the information of all the presented elements, which is provided by several input files. Due to this, the CAD Platform is a crucial element of the system, as it automatically extracts and processes all the data in these files, providing the necessary data whenever is requested, which allows a full process reconfiguration just by updating these input files.

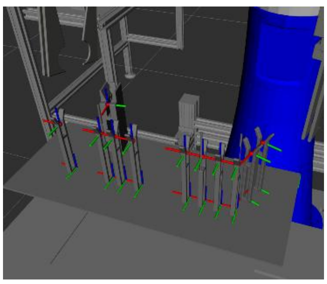
The ELVEZ CAD Platform's input files can be classified into four groups according to the items they are describing:

- **Physical platforms:** Three physical platforms are used to perform the use case. The first one is the "assembly platform" (Figure 23.b) where several guides are mounted; which is used to route the cables of the wire harnesses as per process requirement. The second one is the "cable-holding platform" (Figure 23.c) which consistst of a different set of mounted guides on a dedicated platform, from which the wire harnesses are picked initially. Finally, the last platform is the "ATC platform" (Figure 23.d), which has different slots for picking and placing the robot's tools. Each of these platforms are defined by three files: the CAD 3D model of the platform assembly, in X3D format; an XML file with the description of the geometry and keypoints of all the different elements (i.e., one description per element model) of the platform; and a WRI file that relates the ID of each element of the CAD assembly with its model number and provides a label name for it. Additionally, the position and orientation of each platform within the scene (with respect to the cell's top corner, as can be seen in Figure 23.a) can be specified from the launch file with a static transform publisher. Besides these three files, the STL mesh of each element model of the platform is also necessary to visualize the physical platforms in RVIZ. The name of each STL file must be the commercial number of the commercial model.
- **Manipulated objects:** Two files are used to describe the manipulated wire harnesses. One of them is a CSV file that contains the main attributes (dimensions, color, type, etc.) of all the components of the manipulated harnesses, i.e., cables, connectors, and capacitors. The second file is in XML format and it describes the structure of the harnesses, specifying how the previous components are arranged in each of them.

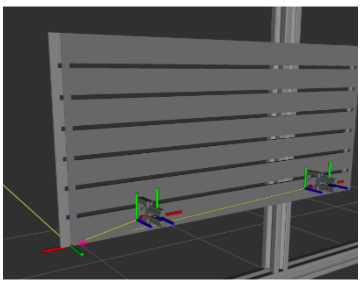
- **Tools:** One XML file is used to describe all the tools available for the robot i.e., the grippers and the automatic taping gun. This description includes the label name of its slot in the ATC platform, the transformation frame from the tool base to the tool changer frame (attached to the robot's wrist), and the transformation frame from the tool changer frame to the actuation frame (i.e., the finger pads of the grippers and the nozzle of the taping gun).
- **Process:** One CSV file defines the high-level action plan for the robotic system. This plan contains the sequence of operations to be executed. For each operation, it indicates the type of operation (i.e., insert connector, route cables, or tape), the component involved (e.g., a certain connector, or a group of cables), and the location (i.e., the label name of the platform elements where the task needs to be performed).



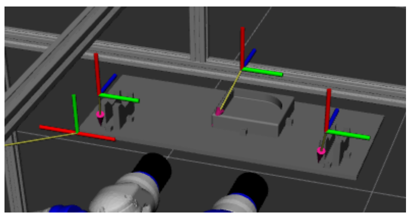
a. Robotic cell



b. Cable Routing Platform



c. Cable Holder Platform



d. Automatic Tool Changing Platform

Figure 23. tfs and Markers of the ELVEZ robotic platform, published by the CAD Platform

The CAD Platform ROS package extracts the relevant information from all these files. The data extraction can be divided into two groups, the X3D files and the rest of the files. For every X3D file, a class instance is created containing information about all the elements in the assembly in a hierarchical structure. Each element is defined as an instance of another class, whose attributes are the position and orientation of its origin frame with respect to the origin frame of its parent element, the type of element (i.e., a guide, an ATC slot, a flat surface, or any other category that may be added in the future), which is extracted from the label name of the WRI file, its model name, its parent element (if any), and its children elements (if any). Regarding the rest of the files, the information of each file is stored in a dictionary.

After the data is extracted and stored in the system, some of the files' information needs to be processed and merged as they provide complementary information. This is the case of the three files that describe the physical platforms. The X3D file provides the relative pose of each element of the platform with respect to its parent element, however, the XML file is the one that defines the pose of the points of interest of each element model with respect to their own origin frame. Therefore, the information extracted from these two files has to be combined, using the WRI file to relate them (see Figure 24). The final result is a dictionary with the pose of each point of interest with respect to the robot's base frame.

```
<Transform DEF="ID000005"
        translation="2.000000 -217.500000 16.940001"
        scale="1.000000 1.000000 1.000000"
        rotation="-0.999998 0.000000 0.000000 0.261800"
        >
                <Shape>
                        <Appearance>
                        </Appearance>
                        <IndexedFaceSet solid="false
                                coordIndex="
34 -1 7 6 34 -1 42 47 14 -1 47 48 14 -1 48 33 14 -1 48 0 33
12 54 -1 112 113 54 -1 113 97 54 -1 107 114 54 -1 114 56 54
180 57 -1 17 179 57 -1 180 181 57 -1 181 182 57 -1 182 183 5
```

a. x3d CAD file

ID000001	T1	0011
ID000002	G1	0008
ID000003	G2	0009
ID000004	G3	0009
ID000005	G4	0009
ID000006	G5	0006
ID000007	G6	0008
ID000008	G7	0009
ID000009	G8	0005
ID000010	G9	0009

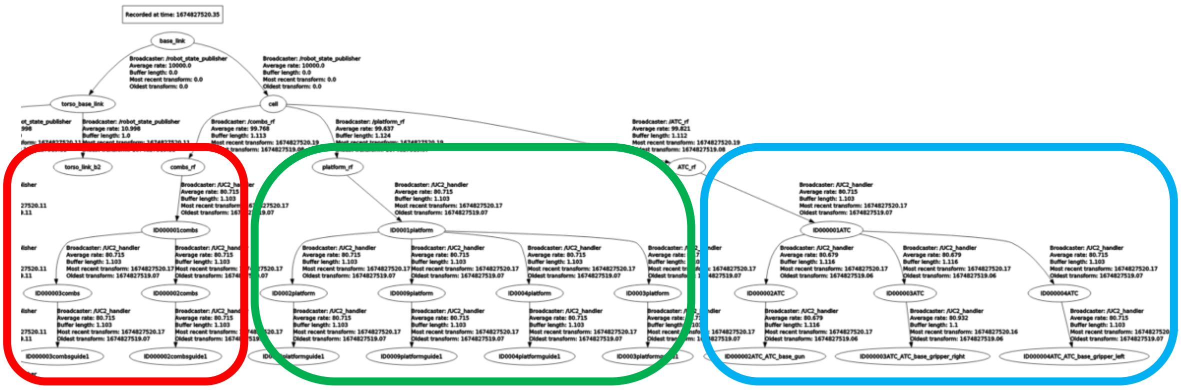
b. csv file with IDs

```
-<guide model="0009" xdim="10" ydim="22.5" zdim="75">
  -<key length="10" gap="12.5" height="30">
      <pos name="x">0</pos>
      <pos name="y">5</pos>
      <pos name="z">45</pos>
      <rot name="R">0</rot>
      <rot name="P">0</rot>
      <rot name="Y">0</rot>
   </key>
 </guide>
```

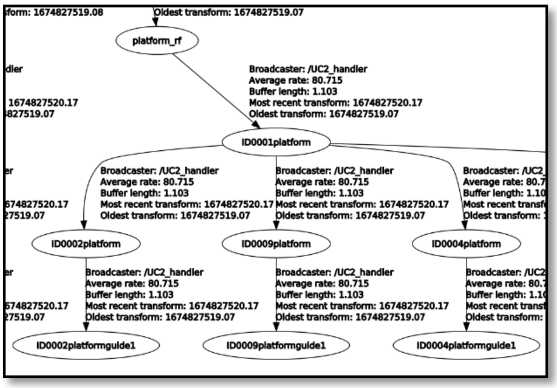
c. xml file with the models' description

Figure 24. Relation between the files describing the assembly platform. In the X3D file (a), it specifies the ID of the component (ID000005), the CSV file (b) relates this ID with a label (G4) and a model number (0009), whose geometry and points of interest are described in the XML file (c).

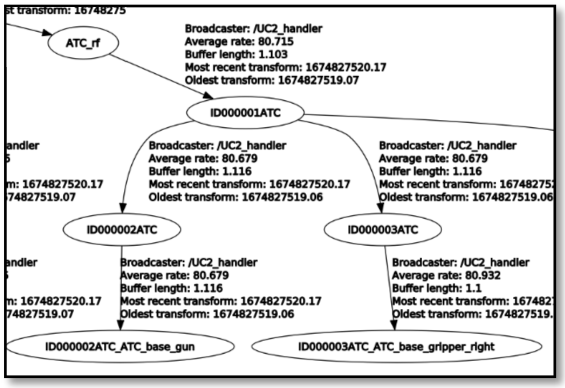
The stored data is then accessed to provide information to other REMODEL modules through ROS services, and to create the virtual representation of the robotic platform in RVIZ. Eight services servers are defined in the CAD Platforms to access to different kind of information, which can be classified into four groups (see Table 2). Regarding the RVIZ representation, a tf frame is defined for the origin of each platform element and their points of interest, following a hierarchical structure (see Figure 25), and markers are created to visualize the STL meshes of all the elements of the physical platforms within the scene (see Figure 23). The whole CAD Platform functionality is shown graphically in Figure 26.



a. Simplified tf tree of the physical platforms. Red: cable holder platform, Green: assembly platform, Blue: ATC platform.



b. Simplified tf tree of the assembly platform (green box in (a))



c. Simplified tf tree of the ATC platform (blue box in (a))

Figure 25. tf tree of the ELVEZ physical platforms generated by the CAD Platform

Table 2. ELVEZ CAD Platform ROS services

Category	Service name	Description and applications	Input (request)	Output (response)
Physical platform elements	/elvez/guide_info	Provides information about a guide. Used to configure the optimal robot trajectories.	• Guide name (string)	• Success (bool) • ID (string) • Dimensions (float[]) • Origin_frame (Pose) • Key_frame (Pose) • Key_corner_frame (Pose) • Key_length (float) • Key_gap (float) • Key_height (float)
Manipulated wire harnesses	/elvez/cable_info	Provides information about a cable. This is useful for their recognition using vision (color, diameter, connector pin index…),	• Cable name (string)	• Success (bool) • Color (int[]) • Length (float) • Diameter (float) • WH (int) • Connector (string)

Category	Service name	Description and applications	Input (request)	Output (response)
		for its manipulation, etc.		• Con_index (int[])
	/elvez/con_info	Provides information about a connector. Useful for its manipulation and vision recognition.	• Connector name (string)	• Success (bool) • Color (int[]) • Dimensions (float[]) • WH (int) • Cables (string[])
	/elvez/wh_info	Provides information about all the elements of a wire harness.	• WH number (int)	• Success (bool) • Connector (string) • Branches: ○ Connector (string) ○ Cables (string[])
Tools	/elves/all_tools	Get the name of all the tools.	None	• Success (bool) • Names (string[])
	/elvez/tool_info	Provides information about a tool. Useful for tool changing and trajectory generation.	• Tool name (string)	• Success (bool) • ATC_spot (str) • Base_wrist_frame (Pose) • Wrist_action_frame (Pose) • Dimensions (float[])
Process	/elvez/all_op	Provides information about all the operations of the action plan. Used by the system planner to determine which functions to call.	None	• Success (bool) • List of: ○ Type (string) ○ Spot (string[]) ○ Label (string[])
	/elvez/index_op	Provides information about a specific operation of the action plan.	• Operation index (int)	• Success (bool) • Type (string) • Spot (string[]) • Label (string[])

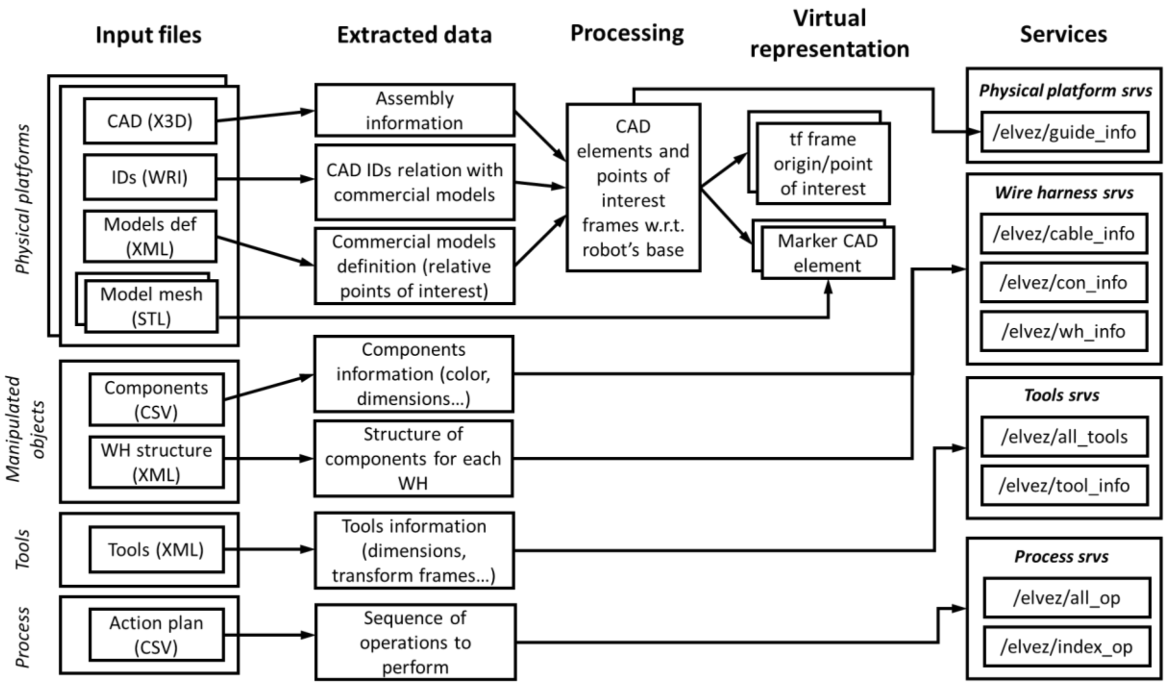


Figure 26. ELVEZ CAD Platform functionality block diagram.

4.4. Volkswagen Use Case

In UC3, the CAD interface is largely simplified since the wiring harness is always the same. For this reason the description of the wiring harness is contained in a text file that is read by the wiring harness detection system in order to create a semantic representation of the actual wiring harness state using the GraphML formalism. The details of the strategy for the detection and the manipulation of the wiring harness are described in D5.4 “Wiring Harness Manipulation”. This approach enable future extension to the cases in which the structure of the wiring harness can change among the different products.

4.5. ENKI Use Case

In UC4, the CAD interface consists simply in the communication to the robotic platform of the actual diameter of the hose to be manipulated. However, the system is open to future extensions in which more details about the products can be passed to the robotic cell to adapt the manipulation to different product requirements.

5. Conclusion

This deliverable documents the CAD Platform Interface module developed for the REMODEL project. This software module extracts data from several files of different natures and types (including CAD design files), and processes it to obtain useful information about the layout, handled components, processes, etc. Additionally, it provides this information in a set of ROS services, so any other node of the REMODEL ROS system can request it at any point. A CAD Platform has been developed for all the use cases of the project. The CAD Platforms of the ENKI, ELIMCO, and ELVEZ use cases have a similar high-level structure, however, a different implementation was required for each of them as the input data provided by each company, and the outputs required by each system are different. In the other two use cases (i.e., WVP and ENKI) the CAD Platforms were largely simplified due to the nature of the process data. All the CAD Platforms have been integrated and tested with the rest of REMODEL modules and their software is ready for deployment.