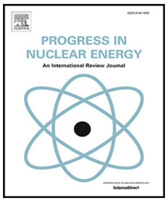
Progress in Nuclear Energy 160 (2023) 104681

Contents lists available at ScienceDirect

Progress in Nuclear Energy

journal homepage: www.elsevier.com/locate/pnucene



# An efficient workflow for virtual reality simulation of maintenance tasks in IFMIF-DONES

Martin Rouret [a,*,1], Kornél L. Varga [a,1], Andrea Benito Fuentes [a], Fernando Arranz [b], Eduardo Ros [a], Jesús A. Garrido [a]

[a] *ValeriaLab, Universidad de Granada, Granada, Spain*
[b] *CIEMAT, Madrid, Spain*

ARTICLE INFO

*Keywords:*
Virtual reality
Nuclear maintenance simulation
Remote handling
Logistics
Task validation

ABSTRACT

An efficient workflow has been developed to generate virtual reality simulations of the maintenance tasks requiring remote handling in DONES (a particle accelerator facility). This framework focuses on the communication between different engineering teams and outlines a Virtualization Task Documentation, which organizes all the necessary information needed to create the simulation. Moreover, this workflow includes a node-based design tool that allows effective translation from natural-language description of the maintenance tasks and actions into virtual reality simulations. The efficiency of the workflow and usefulness of the resulting simulations was demonstrated in a specific use case, the replacement of the high-energy beam transport scraper in the accelerator system in DONES. The simulation of the whole replacement process has been generated. The presented workflow has also been useful for the simulation of maintenance procedures in other areas of the same facility, proving to be easy to adapt to other processes.

## 1. Introduction

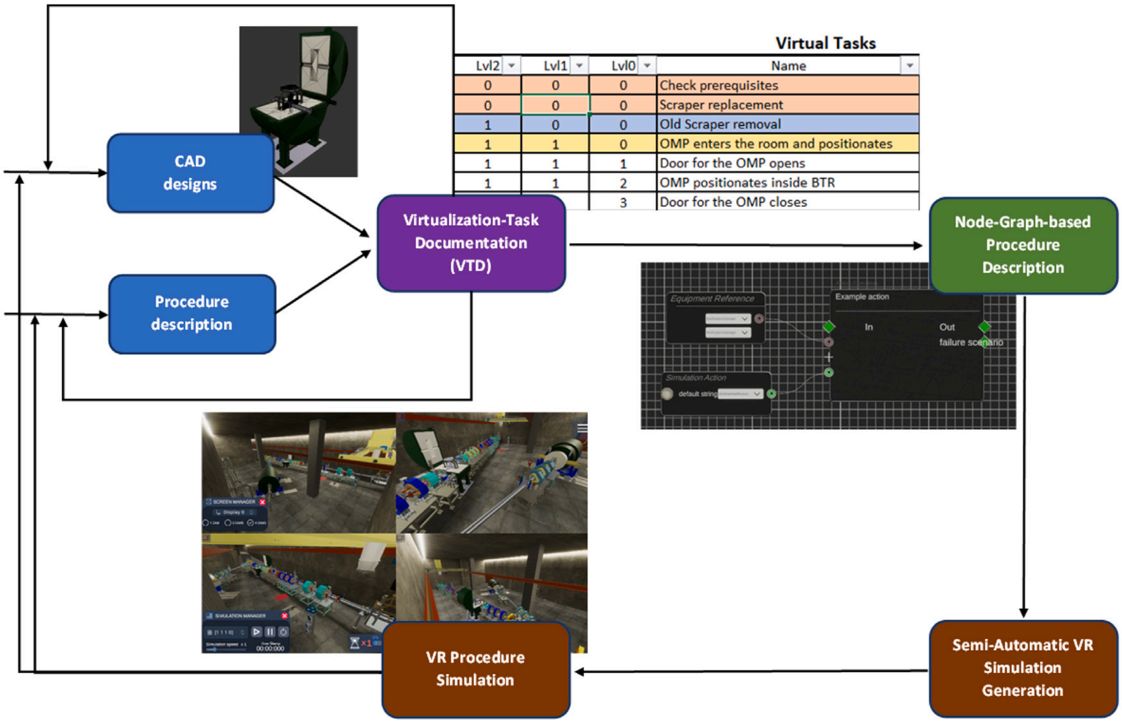
The production of electricity with low-carbon sources is an accepted global need that has pushed the research and development of industrial power stations based on nuclear fusion reactors. The organization coordinating the European effort in this area is Eurofusion. It defined the "Roadmap to the Realization of Fusion Energy" (EUROFusion, 2023). The development path goes through three main stages: operation of the International Thermonuclear Experimental Reactor (ITER), operation of the materials testing facility, the DEMO-Oriented Neutron Source (DONES) (Podadera et al., 2021) and construction and operation of the demonstration power station (DEMO). All these facilities aim to solve important technological issues, being one of them the maintenance of the plant equipment using remote handling procedures. The use of Virtual Reality (VR) tools to assess the processes of such maintenance is a key factor for success, allowing validation of equipment and procedure definition without the need of manufacturing costly prototypes.

The design of this kind of international large scale scientific facilities requires the integration of contributions of many teams and companies from different countries and segments. Furthermore, each scientific facility addresses specific goals that make it singular. This motivates the construction of pathfinder methods that allow preliminary validation of operational and functional concepts. In particular, one of the main aspects that needs to be addressed at the initial stages of the design of infrastructures is the maintenance of the facility. This is the case of installations with irradiated areas which usually call for some of their maintenance tasks to be remotely performed with robotic equipment. Remote handling of maintenance tasks is crucial, as it involves integrating heavy equipment such as cranes, robotic manipulators, and automated vehicles to manipulate large masses through narrow pathways. Hands-on tasks are also necessary and simulation aspects of human intervention are considered as well, such as the estimated time it takes to perform certain tasks.

The DONES facility will include remote handling equipment for carrying out some periodic maintenance operations, due to hazardous environment and difficulty of access (Ibarra et al., 2019). The main areas where remote handling will be used are: (i) the test cell (TC), for the replacement of the target assembly (TA) and high flux test module (HFTM), since both plant equipment will be highly activated, (ii) the lithium systems, where periodic inspection and replacement of traps is needed and (iii) accelerator systems, that will require preventive maintenance including replacement of equipment that gets activated

[*] Correspondence to: Gran vía de Colón 48, Piso 4, Despacho 401, 18010, Granada, Spain.
*E-mail addresses:* martinrouret@ugr.es (M. Rouret), kornel@ugr.es (K.L. Varga), andreabenito@ugr.es (A.B. Fuentes), fernando.arranz@ciemat.es (F. Arranz), eros@ugr.es (E. Ros), jesusgarrido@ugr.es (J.A. Garrido).
[1] These authors contributed equally.

https://doi.org/10.1016/j.pnucene.2023.104681
Received 20 June 2022; Received in revised form 30 January 2023; Accepted 31 March 2023
Available online 11 April 2023



| Lvl2 | Lvl1 | Lvl0 | Name |
|---|---|---|---|
| 0 | 0 | 0 | Check prerequisites |
| 0 | 0 | 0 | Scraper replacement |
| 1 | 0 | 0 | Old Scraper removal |
| 1 | 1 | 0 | OMP enters the room and positionates |
| 1 | 1 | 1 | Door for the OMP opens |
| 1 | 1 | 2 | OMP positionates inside BTR |
|  |  | 3 | Door for the OMP closes |

**Fig. 1.** Schematic diagram of the overall workflow for VR simulation generation of maintenance tasks.

(e.g. high-energy beam transport -HEBT- scraper), or is inaccessible (e.g. target interface room -TIR- modules, which are also activated).

Complex maintenance procedures for scientific facilities can be defined at different levels of detail. The virtualization of these tasks frequently requires the integration of multiple equipment components (CAD designs) which are being designed in parallel and building models that receive constant updates in the layout. However, the feasibility of the resulting procedures need to be assessed, and in some cases, these simulations promote changes in the designs or the operations to be performed. Thus, detailed definition and verification of the maintenance procedures are critical at very early stages, and 3D visualization of these procedures represents a flexible and intuitive tool in order to validate both maintenance operations and models (Boy et al., 2016; Gatto et al., 2013).

The design of scientific facilities and their components has traditionally been addressed with computer-assisted design tools (such as Dassault Systèmes Solid Works or Cadtech Catia/Delmia packages). These software tools are mainly oriented to mechatronic design of the components, but allow rendering the design in order to visually check integration feasibility and inspection of different set ups. In the last years, virtual reality (VR) industry has evolved rapidly, mainly motivated by gaming and multimedia industries, emerging widely used simulation engines such as Unity 3D (Wang et al., 2015). These tools facilitate the creation of virtual environments and the definition of both sequences of events (predefined movies) and interactive simulations. The use of VR reconstructions of remote handling maintenance operations has been used for similar facilities in nuclear fusion, such as JET (Williams et al., 2011), ITER (Tesini and Rolfe, 2009), DEMO (Loving et al., 2014) (Crofts et al., 2022), ESS (Boman and Smisovsky, 2016), CERN (Martínez et al., 2014), IPR (Dutta et al., 2017), IEN/CNEN (Mol et al., 2009).

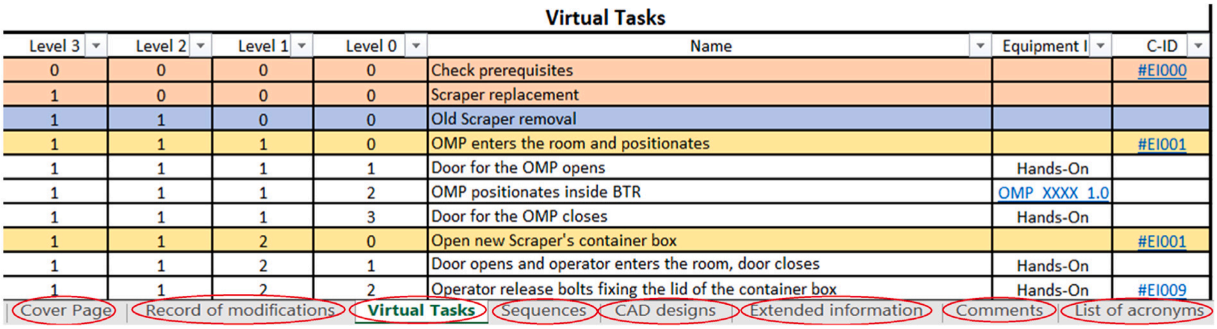
In this article we present an efficient workflow for the development of VR simulations of maintenance procedures at DONES (Fig. 1). This workflow includes structured documentation for definition of the maintenance tasks (by means of the virtualization task documentation), a node-base design of VR simulations as a description language, a graphical tool for semi-automatic generation of VR simulations based on Unity3D engine and its application into the accelerator system of DONES. The proposed methodology facilitates continuous integration of new and updated maintenance tasks, creating interactive applications of virtual reality simulations for their verification. This agile integration of CAD designs in reference repositories, procedure definition and 3D animations motivates the use of Unity that enables efficient creation of verification applications.

Being a standard in video game development the engine Unity3D represents a high value compared to traditional robotic software. It is widely used for creating interactive 3D simulations, which can be more engaging and realistic than traditional robotic simulations, providing the developers the necessary tools to include efficiently as many functionalities as needed. It also offers a wide range of built-in physics engines, animations, and other features that can be used to create detailed and accurate simulations of robotic procedures. It is relatively easy to use, and compared to robotic software simulators, it has a large community of developers and users, making it easier to find help and resources when creating simulations. It is also a cross-platform engine, which means that the simulations can be run on different devices, such as computers, mobile devices, and VR headsets, making it more versatile than traditional robotic simulators (other game development tools such as Unreal Engine would also be a valid alternative).

Along this article, the next sections will describe each of the steps in the workflow. Section 2 will introduce the virtualization task documentation. Section 3 will present how the information in this document can be transferred into a node-based description of the procedures. Section 4 will cover how this description is used to generate VR simulations of the procedures in Unity 3D. Section 5 will describe the particular use case of the High-Energy Beam Transport scraper replacement virtualization, an important element in the accelerator system. Finally, Section 6 will discuss the main advantages and limitations of the proposed workflow.

## 2. Maintenance process description into a virtualization task document

The exhaustive definition of the maintenance steps represents a crucial milestone in order to validate the feasibility of the procedure.



Virtual Tasks

| Level 3 | Level 2 | Level 1 | Level 0 | Name | Equipment I | C-ID |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Check prerequisites | | #EI000 |
| 1 | 0 | 0 | 0 | Scraper replacement | | |
| 1 | 1 | 0 | 0 | Old Scraper removal | | |
| 1 | 1 | 1 | 0 | OMP enters the room and positionates | | #EI001 |
| 1 | 1 | 1 | 1 | Door for the OMP opens | Hands-On | |
| 1 | 1 | 1 | 2 | OMP positionates inside BTR | OMP_XXXX_1.0 | |
| 1 | 1 | 1 | 3 | Door for the OMP closes | Hands-On | |
| 1 | 1 | 2 | 0 | Open new Scraper's container box | | #EI001 |
| 1 | 1 | 2 | 1 | Door opens and operator enters the room, door closes | Hands-On | |
| 1 | 1 | 2 | 2 | Operator release bolts fixing the lid of the container box | Hands-On | #EI009 |

**Fig. 2.** Screenshot of an example VTD with specific sections (as sheets) highlighted with red circles. Note that every maintenance step (rows in the Excel sheet) links the hierarchical organization (initial 4 columns), the identifier of the equipment required (6th column) and related comments (last column, only specified if they exist). Each of the colored lines identify an specific level for easier reading, Level 3 tasks are orange, level 2 blue, level 1 yellow and level 0 white.

In IFMIF-DONES this definition is based on the Virtualization Task Documentation (VTD). The VTD is thought of as a tool to facilitate a clear interaction between teams from different domains (mainly engineers involved in the design of the components and the procedures and virtual reality developers). The VTD includes all the information required to allow any VR designers to reconstruct the 3D simulation of the tasks with no additional input. It includes all the steps required to complete the maintenance procedures based on the specific documents for those procedures, the listing (and the reference) of the CAD designs of the plant, the systems to be maintained and the tools required for the remote handling. Moreover, the VTD has been complemented with comments to clarify the different steps and question/answers to facilitate the interaction between the tracing of the decisions. This document has been implemented as a Microsoft Excel file, so that it allows easy filtering and sorting of the different task levels, providing a faster overview of the levels of detail.

The VTD is composed of 8 different sections included as different Excel sheets (Fig. 2):

1. Cover page: title of the maintenance procedure described in the VTD, specifying authors, co-authors, reviewers and approvers of the document.
2. Record of modifications: simplified record of the changes made in the different versions of the VTD, full details of said modifications are specified in a final technical note.
3. Virtual tasks: the hierarchical listing of the different steps needed to perform the maintenance procedure.
4. Sequence functions: sequences of steps that are repeated among the virtual tasks and are enclosed in one single step to avoid repetition, e.g. *mount robotic arm*. These sequences improve readability of the virtual task steps and facilitate the implementation of the simulation.
5. CAD designs: the list of the CAD models to be used in the simulation, with their specific link to the project repository, the last update date and the identifier of the equipment model for later retrieving from our internal models database.
6. Extended information: some of the procedures need more explanation in order to be fully understandable for everyone involved in the virtualization of the procedures. The extended information page explains specific steps in detail, adding images or links to additional documents (e.g. the task *check prerequisites* has an extended information link with identifier *#EI000* describing which the prerequisites are as shown in Fig. 2).
7. Comments: this section allows the reader to keep track of the different conversations (questions and answers), decisions made and the reasons behind.
8. List of acronyms: the list with the acronyms used in the virtual tasks page. It mainly clarifies the acronyms used for the remote handling equipment.

As shown in Fig. 2 each task of the VTD (rows) has an equipment ID associated to it. E.g. task [1 1 1 2], which reads "OMP positionates inside BTR", has the Equipment ID OMP_XXXX_1.0, representing a particular version of the 3D model. This explains the linking between tasks and 3D models. The actions are the representation of every task (rows in the VTD) in the node system, in order to translate the tasks from the VTD into actual code (or animations) inside Unity. In order to clarify the different procedures for the maintenance processes in the virtual task sheet, the VTD structures the steps into different levels of detail:

1. Level 0. Represented in white. The most concrete tasks, which are defined by themselves (e.g. *moving AMCN (Accelerator Mast Crane North) to a specific point, grabbing some equipment*, …).
2. Level 1. Represented in yellow. Groups of tasks of level 0 that define a general action, such as *test cell cover plate removal*, which contains several level 0 tasks like *heavy rope overhead crane moves to test cell cover plate position or suspended plate down to test cell cover plate*.
3. Level 2. Represented in blue. Groups of actions of level 1 that define a procedure such as *test cell opening* or *heavy-flux test module removal*.
4. Level 3. Represented in orange. Groups of procedures of level 2 that define a whole maintenance process such as *target assembly maintenance and high-energy beam transport maintenance*.

Thus, every step of a maintenance procedure has been included as a new row in the sheet (Fig. 2). In addition to the hierarchical task levels and the step description, in the same table, the following columns have been considered:

1. Equipment ID. Link to information about the version of the CAD models required for this step or link to the explanation of the sequence function.
2. State. The actual state of the procedure step in the simulation. The following states can be considered so far: ongoing (OG) or approved (AP).
3. EI-ID. Link to extended information section regarding this maintenance step (if exists).
4. C-ID. Link to comment section regarding this maintenance step (if exists).

## 3. Node-based definition of the maintenance processes

Aiming to overcome the difficulties to translate human descriptions of the maintenance procedures into a virtual reality simulation we have developed a node-based software tool that facilitates this conversion. The node-base software has been developed as a C# library extending from the Unity3D software which uses the same language, allowing better integration into the framework. Node-based information handling

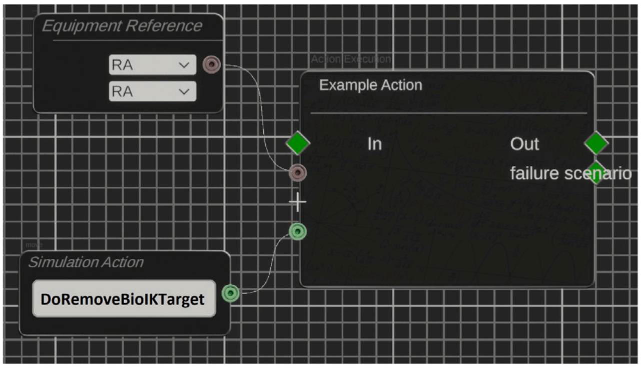


**Fig. 3.** Example action node with incoming connections to action and equipment reference.

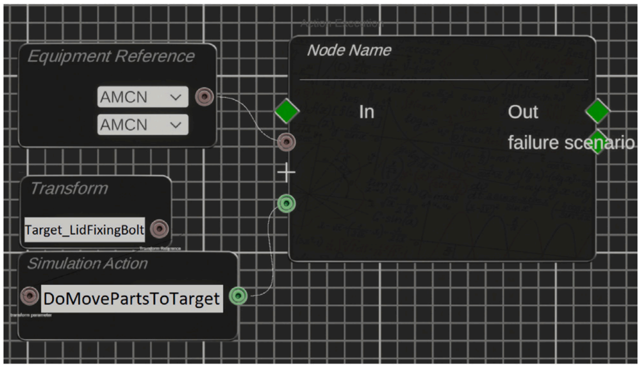


**Fig. 4.** Example of node-based graph to implement the movement of a specific equipment (ACMC) to target.

has become a standard solution in many different fields (Keller et al., 2006). In this article we have applied a node-based representation to describe the different operations required for a particular maintenance procedure. By using the capability of Unity's customization editor interface we have implemented a solution that facilitates the transformation of the procedures defined in the VTD into a 3D simulation of the operations. The node-based representation allows us to modify simulation processes as well as it enhances the visual comprehension of the sequences of actions to perform.

In our implementation, the meta-class of a node, or in other words the abstraction of all nodes incorporates three main components (Fig. 3): (i) the node body, a graphical representation of the node with the title and the type of node, (ii) the incoming slots, where all incoming connections from other nodes are represented and (iii) the outgoing slots, where all outgoing connections to other nodes start.

The flow of progression in a graph always starts with an entry node (a node with no input slots and one output slot), and ends with an exit node (no output and one input slot). The green rhombus shape slots always indicate the input and output of the execution flow. Additionally, aiming to support the workflow of virtualization of remote handling maintenance procedures, the following types of nodes have been implemented:

1. Action-execution nodes. This kind of node represents the abstraction of running actions in the simulation (e.g. *NotificationManager displays a message or moving an object to a target position*). In a later stage, every action will be converted into a scene component that subsequently performs the execution of an action. This type of node corresponds to the "Example action" node in Fig. 3.
2. Checkpoint nodes. Progression control is necessary when it comes to the visualization of processes. Thus we need to mark or highlight when a particular subtask starts or ends. In general, the progression of a process is to be expected to start with an entry node in every node-graph and will end with an exit node. In order to mark junctions between the entry and exit nodes we use checkpoint nodes allowing users to jump between tasks in the simulation.
3. Reference nodes. These nodes are responsible for containing pointers towards equipment, spatial entities or actions of the equipment in the scene. This type of node corresponds to the *Equipment reference* and *Simulation Action* nodes in Fig. 3.

The resulting Unity3D project will include a scene called *Node Editor*. This scene will contain all the nodes describing the maintenance task, and it will reference the behavior of the equipment object's implemented as described in Section 4 below.

By following the VTD description we made in the previous stage, we can create a maintenance simulation. For example, one of the steps included in the VTD could be to move one equipment (the so-called AMCN) to a specific position, so we will need to create a node with this equipment reference and the simulation action will be ACT_MoveToTarget (called as DoMovePartsToTarget in the node), passing as a parameter the target in the transform node, as shown in Fig. 4.

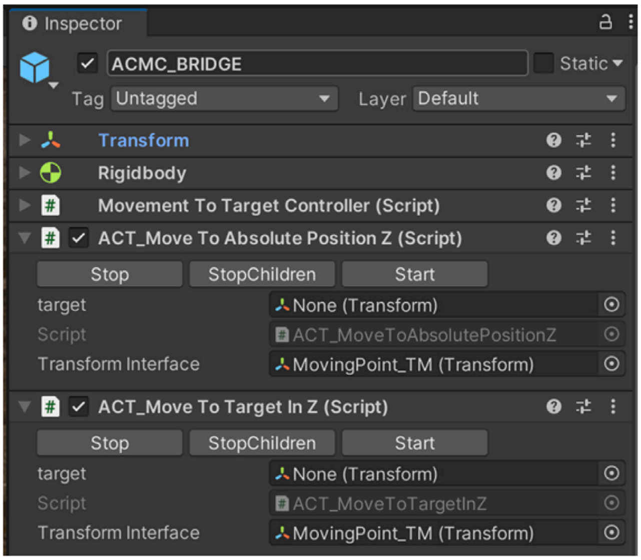
## 4. Equipment behavior implementation and generation of VR simulations based on Unity3D engine

Once the maintenance tasks are described as a VTD and it has been translated into a node-based description, our workflow requires generating an Unity3D scene with all the elements required for this. Each equipment object will implement multiple behavior as by using ACT scripts. There are different types of actions (ACT), some of the fundamentals ACTs are:

1. ACT_AttachObject/ACT_ReleaseObject. With these scripts, any equipment object is able to grab/release other equipment objects.
2. ACT_MoveToAbsolutePosition(X,Y,Z). With this script, an equipment object will be able to move to a specific target position, indicated as the (X,Y,Z) absolute coordinates. This component takes as an input a float value to move to that position in global space.
3. ACT_MoveToTarget (in X, Y or Z). This kind of script will move the end equipment object to a target position along the X, Y or Z axis (only one of these). This script will receive a single value as a target position and the other two coordinates will remain unchanged.
4. ACT_WaitForSeconds. With this script, the resulting simulation will wait for some specified time.

Fig. 5 shows an example of a configuration in an object with two of the behaviors explained before. Similarly, every object in the simulation will include only those actions which this object perform.

By using the node-based implementation we are splitting the creation of the simulation into two different levels of abstraction. The lower level will consist of implementing the specific behavior allowed for each equipment object by means of the ACT scripts. The higher one will describe the whole maintenance process by referencing only the equipment and the behaviors implemented for each of them. Once we finish with the creation of the graph and the implementation of the action behavior for every object we can build the resulting VR simulation of the maintenance task.



☑ ACMC_BRIDGE

Tag Untagged

Layer Default

Transform

Rigidbody

Movement To Target Controller (Script)

☑ ACT_Move To Absolute Position Z (Script)

target: None (Transform)

Script: ACT_MoveToAbsolutePositionZ

Transform Interface: MovingPoint_TM (Transform)

☑ ACT_Move To Target In Z (Script)

target: None (Transform)

Script: ACT_MoveToTargetInZ

Transform Interface: MovingPoint_TM (Transform)

**Fig. 5.** Example of configuration for a specific mast crane equipment in the scene. This object implements two ACT behaviors: ACT_MoveToAbsolutePositionZ and ACT_MoveToTargetInZ.

## 5. Use case: HEBT scraper replacement

### 5.1. HEBT scraper replacement in IFMIF-DONES

The accelerator system of IFMIF-DONES (Podadera et al., 2021) includes an injector that will produce the deuterons to be accelerated in the posterior equipment, where the current of the beam is set at 125 mA. The injector is followed by a radio-frequency quadrupole that increases the energy of the particles up to a level of 5 MeV, and then, the Medium Energy Beam Transport Line (MEBT), containing five magnets and two rebuncher cavities in order to prepare the beam for the next component, the Superconducting Radio-Frequency LINAC (SRF LINAC). The five modules of the SRF LINAC take the energy of the particles up to 40 MeV, which is the optimal value to impact the Lithium curtain at the end of the accelerator in order to produce neutrons with the same characteristics as in a fusion reactor. The beam is transported up to the target by the High Energy Beam Transport Line (HEBT), where it is deflected 9° for protection of the equipment of back scattering radiation. The HEBT has also a scraper that removes particles that have been separated from the core of the beam line, creating a halo that could heat and activate the beam tube. This scraper will become activated because of the reactions of the particles stopped in it, so that it is enclosed into a shielding that stops both neutrons and gamma rays. The effect of the radiation is a deterioration of the materials and therefore a need for periodic replacement with remote handling means.

The HEBT scraper replacement is a procedure composed of two level-two tasks in the proposed VTD: the removal of the old scraper which has been exposed to radiation and the installation of a new scraper. To complete the VTD, twenty-one level-one tasks were defined such as *Open new scraper's container box* which, in turn, is also composed of twelve level-zero tasks such as *Door for the operator opens* and *Telescopic Mast up to moving position*. Note that, as stated above, level-zero tasks are the most simple tasks that can be defined. However, other level-one tasks are simpler, as *OMP (Omnidirectional Mobile Platform) enters the room and positions*, which is only composed of three level-zero tasks such as *Door for the OMP opens* and *OMP positions inside BTR (Beam Transport Room)*. As an overall indicator, ninety three level-zero tasks and twenty-one level-one tasks were defined for this maintenance procedure.

In total, fifteen CAD files with different complexity were used for this 3D simulation, including the CAD model of the main building (composed of walls, ceilings and floors for the different rooms), the different remote-handling equipment pieces, and the container box (a box where the old scraper is introduced after the removal).

### 5.2. Development of the node-based representation of the maintenance task

The creation of the node-based description of the maintenance task represents an intermediate state from the human-readable task definition in the VTD towards the application providing 3D visualization of the maintenance task. According to this approach, based on the existing descriptions of the maintenance processes, we structured all the information according to the VTD layout.

As a way to organize the work, all the level-1 tasks inside the node graph are arranged in rows composed of all the necessary nodes to finish said tasks. Fig. 2 shows a zoomed-out view of a level-3 task in the VTD representation for the HEBT replacement. Note the different levels of task abstraction (level-0 steps are marked with white background cells, level-1 tasks are written on yellow background cells, level-2 tasks are marked with blue cells and level-3 tasks are written on pink background). Then, the information in the VTD was used to create the node-based representation of the maintenance tasks. Fig. 4 shows a zoomed-in view of a single node which represents a level-0 task in the node-based representation. Note that this action execution node receives input from two additional nodes, the first one representing the AMCN model to use (reference-node type) and the second one representing the action script to use (move object to the target)

A total of 517 nodes were needed for this simulation. 136 of those nodes were action-execution nodes. Each one of those represents one or multiple actions over one specific equipment or interface feature (such as showing the name of the current task in the simulation view). 136 reference nodes for equipment were also needed, one per action. Finally, 173 reference nodes for actions and values were needed to define specific parameters of each action, such as moving an equipment piece to a certain global position in the world.

Moreover, 49 reference nodes for transforms were used to define positions or rotations that need to be specified, e.g. *move the AMCN to the new scraper position*. Differently to the action and values nodes, these nodes refer to objects which can alter their position through the simulation, allowing the simulation generator to determine the target position during the simulation realization.

In addition to these, 21 checkpoint-reference nodes were used to allow the simulation to jump from one step to a different one. In this case, we set up as many checkpoints as level-1 tasks. Finally, we created one start node defining where the simulation should start playing and an end node to specify where it ends.

### 5.3. Virtual reality simulation

Fig. 6 shows a snapshot of the resulting 3D simulation generated for the replacement of the HEBT scraper. The movie of the whole maintenance task is provided as Supplementary Material. In order to achieve high quality visualization with low computational cost, different optimization techniques had to be implemented. Regarding the rendering of the CAD models, several triangle meshes were created and were conveniently used based on the distance to the observing camera. Moreover, the culled parts (such as cables inside the machines or screws in the floor) were also removed to reduce the computational requirements. Most of the lights were baked into the scene so we avoided runtime calculation of the lightning. And also occlusion culling got applied, so that objects outside of the actual view do not need to be rendered, reducing computational load.

The resulting simulation is distributed as an interactive application and includes different features that allows the users to have a better understanding of the overall process and facilitate validation of the maintenance tasks. Some of these features are:

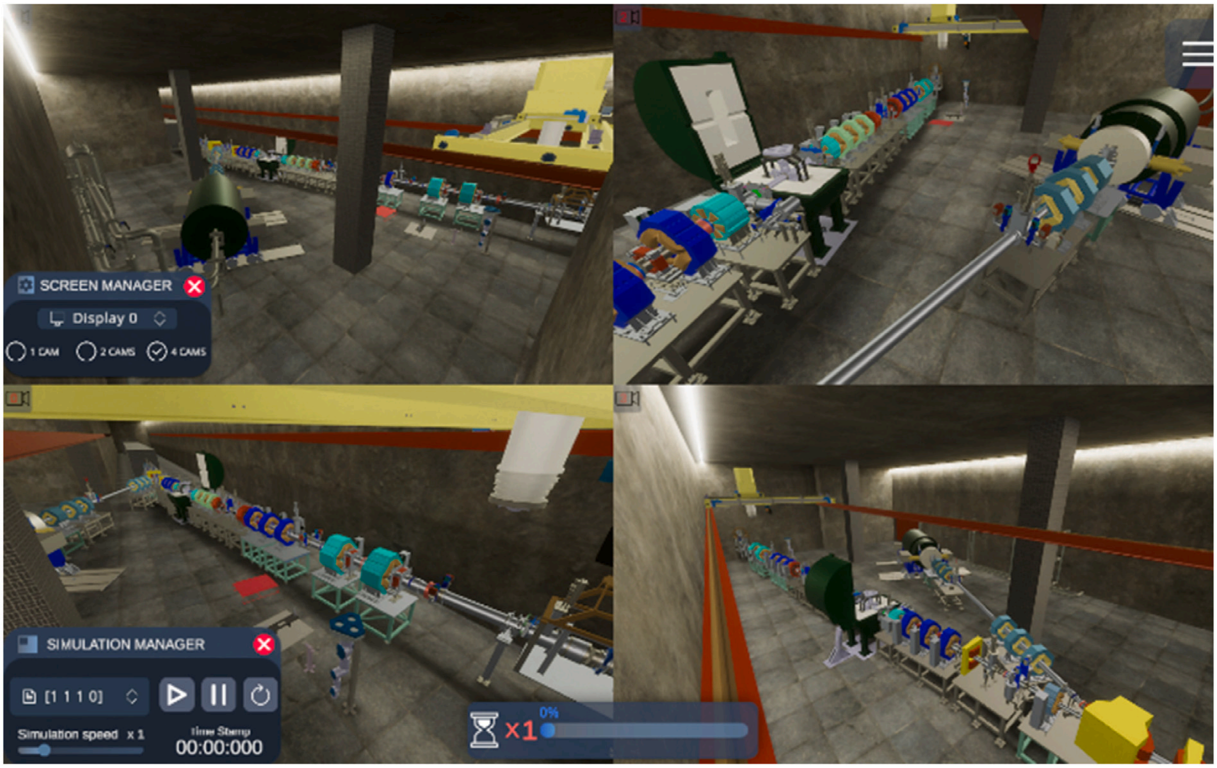


**Fig. 6.** Final aspect of the HEBT scrapper replacement simulation interface. Note that, in this case, the screen was splitted into 4 different camera views. In the small control panel called *simulation manager* the step being executed is indicated in brackets. The simulation manager also allows to change the speed of the simulation.

1. A check-point manager that allows the user to jump between different level 1 tasks as described in the VTD.
2. Simulation speed regulation, allowing to accelerate the simulation up to 5 times normal speed.
3. A multiple camera manager that enables up to 4 cameras per monitor to be visualized. Every monitor area can be assigned to one of the multiple cameras configured for the scene.
4. Free camera: a flying camera that can be voluntarily moved around the scene. This feature allows the exploration of new camera positions, convenient for the correct observation of the actions.
5. Measure camera. This camera allows the user to measure distances between points in an orthographic view. It can also change its position in the plane, letting the user observe inside the objects in the scene.
6. Child camera. This camera follows an object around the scene. It showed especially useful for the observation of the telescopic mast operation and the attaching procedures.

### 5.4. Virtual reality simulation outcomes

The proposed methodology and the resulting simulation have proven to be useful for the design and validation of remote handling maintenance procedures in DONES. Some of the outcomes identified are the following:

1. The robotic arm does not need to store the bolting tool after using it and can be stored with it attached since it is used during the whole procedure of the scraper maintenance, saving operation time.
2. The tools have been moved to the other side of the room for easy storage. This setup optimizes the overall time of operation.
3. The scrapper multi-connector design could be started based on the requirements spotted in the simulation.
4. Changes in the scraper shielding polyethylene part need to be performed in order to operate on the multi-connector with the robotic arm.
5. The bolting tool mockup design needs to consider the distance to the robotic arm once attached to it to avoid collisions as shown in Fig. 7A.
6. The operator needs a stair for certain procedures since the working points are too far from him to reach them as shown in Fig. 7B.
7. The scraper shielding opening orientation had to be switched to the other side since the operator was entering the room from another side and it would make impossible the manipulation of certain objects as shown in Fig. 7C.
8. The robotic arm storage rack height needs to be longer than the robotic arm fully extended plus the maximum size of the different tools that can be attached to it in order to avoid collisions with the floor as shown in Fig. 7D.

These outcomes are then communicated to the responsible designers for decisions regarding necessary changes.

## 6. Conclusions

This article describes a workflow implemented in order to efficiently generate VR simulations of the maintenance operations in DONES, a particle accelerator facility. Our workflow splits the generation of the VR simulation into two different stages: the definition of the equipment behavior by means of action scripts and the description of the sequence of actions by using a node-based representation. This strategy has resulted in increased re-usability of the work in two different fronts. As equipment is used through multiple maintenance procedures with repeated behavior, the nodes can be easily copied and pasted into new projects tweaking only a few parameters like target points. On the one hand, different parts of the node-based representation can be duplicated and modified as required for different parts of a maintenance process.

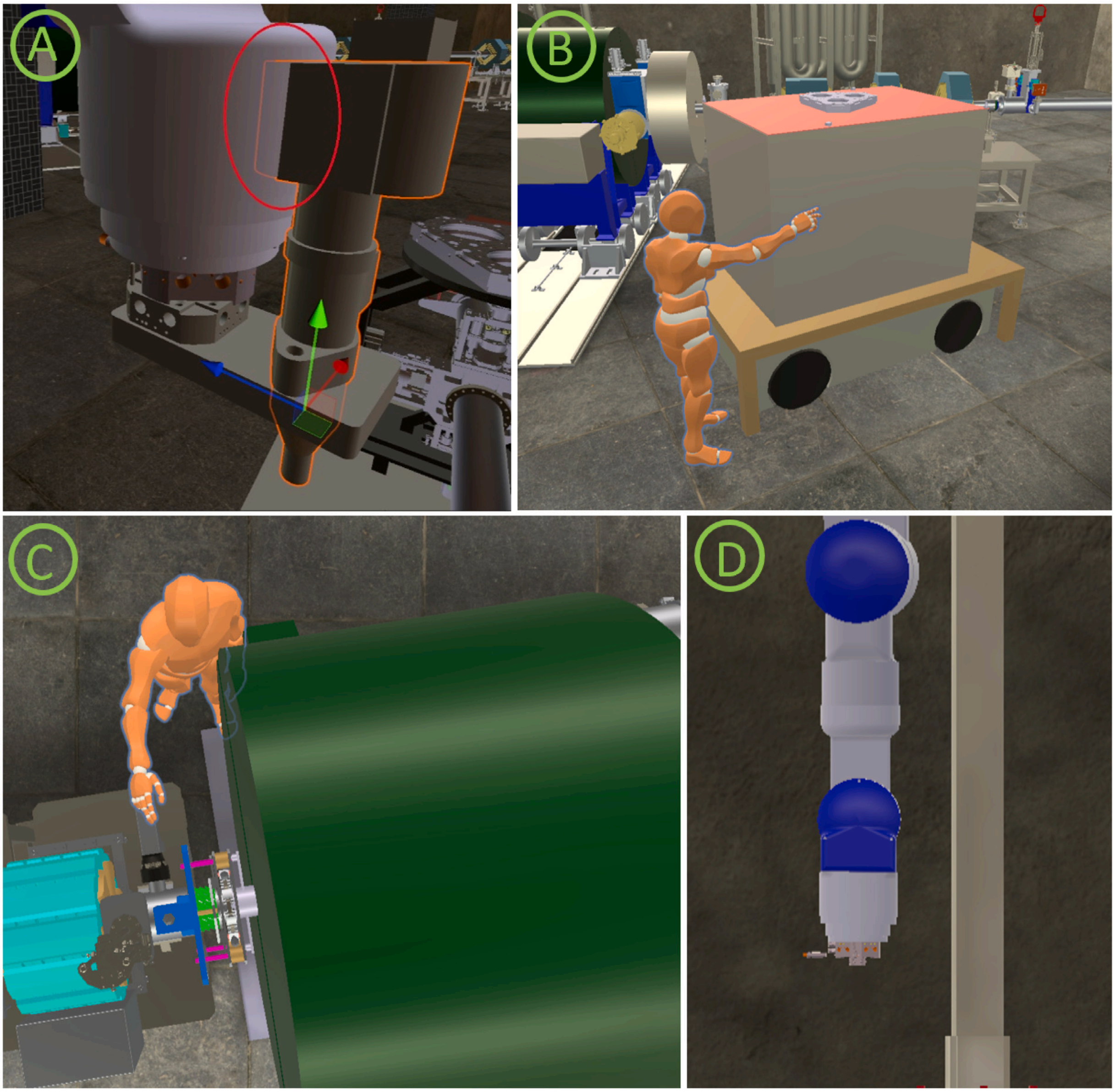


**Fig. 7.** Outcomes examples of the HEBT scraper maintenance procedure. From left to right and top to bottom: (A) Collision between robotic arm and mocked up bolting tool (B) Stairs needed for the worker to get to the working point (C) Scraper shielding opening orientation needed to change (D) Height of the arm storage rack needed to be longer.

On the other hand, the action scripts can be easily adapted from one piece of equipment to another. The same strategy has also been applied for the generation of VR simulations of the maintenance of other areas in the same facility, such as the test cell.

The presented workflow includes several advantages over the alternative of doing direct implementation of the VR simulations from the description of the procedures, such as:

1. A structured way of communication and detailed description of maintenance operations (VTD) for the designers of the maintenance procedures and the researchers developing VR simulations, linking CAD models, actions and parameters of the actions. This document also represents a unique point of reference (VTD) to reconcile all the information required to develop simulations.
2. Faster development of the simulations even considering a framework where CAD models and procedures are still under design.
3. Full set of steps to conceptualize and translate natural language description of actions into an interactive graphical 3D simulation of tasks.

This workflow allows the verification of the definition of the different maintenance tasks and the early detection of possible inconsistencies in the procedures and designs. It enables the integration of links to virtual reality sequences (that help in feasibility analysis and validation) and facilitates the evolution of the design during the development stages. The VTD for the HEBT scraper replacement has been added as supplementary material for better illustration.

The main limitation of this method is not being able to reach a fully automated procedure, mainly due to the inclusion of natural language in the description of maintenance tasks that has to be interpreted by the developers of the VR simulations. At this stage, the presented work is aimed at offline simulations, not allowing the user to manually manipulate the equipment through the simulation, nevertheless, interactive simulations are work in progress to facilitate deeper validation processes.

Finally, the generation of VR simulations of maintenance operations by using VR-oriented engines, such as Unity 3D has the potential advantages of allowing us to export the generated environments into more interactive frameworks (Henrique da Silva et al., 2015; Katsumi et al., 2020). A video presenting this maintenance has been added as supplementary material. In particular, the presented scenarios will be used to facilitate training of the operators of the maintenance tasks once the plant will be active. In addition to this, a highly immersive version of the maintenance has also been generated where the user wears a head-mounted display and freely moves around the scene while the maintenance is taking place. However, further studies regarding depth perception of users in virtual space through gaze tracking should be taken into account in order to avoid motion sickness and other problems that VR might lead to Liu and Qin (2022). Therefore, all these new capabilities will require additional exploration in the future.

## CRediT authorship contribution statement

**Martin Rouret:** Software, Methodology, Visualization, Investigation, Writing – original draft, Writing - review and editing. **Kornél L. Varga:** Conceptualization, Software, Methodology, Writing – original draft. **Andrea Benito Fuentes:** Software, Writing – original draft. **Fernando Arranz:** Supervision, Writing – original draft. **Eduardo Ros:** Supervision. **Jesús A. Garrido:** Supervision, Project administration, Writing – original draft.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors are unable or have chosen not to specify which data has been used

## Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training program under grant agreement No. 101052200. The views and opinions expressed herein do not necessarily reflect those of the European Commission. Moreover, this study has been (partially) financed/supported by the Regional Government of Andalusia through the project “TAN19_UGR_IFMIF-DONES” and European Regional Development Fund (ERDF) “A way to make Europe”/“Andalusia moves with Europe”.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.pnucene.2023.104681.

## References

Boman, E., Smisovsky, L., 2016. Remote Handling within the Active Cells Facility at the European Spallation Source, Using Digital Reality Techniques (Master’s thesis). Lund University.

Boy, G.A., Jani, G., Manera, A., Memmott, M., Petrovic, B., Rayad, Y., Stephane, L., Suri, N., 2016. Improving collaborative work and project management in a nuclear power plant design team: A human-centered design approach. Ann. Nucl. Energy 94, 555–565. http://dx.doi.org/10.1016/j.anucene.2015.12.039.

Crofts, O., Loving, A., Torrance, M., Budden, S., Drumm, B., Tremethick, T., Chauvin, D., Siuko, M., Brace, W., Milushev, V., Mittwollen, M., Lehmann, T., Rauscher, F., Fischer, G., Pagani, P., Wang, Y., Baars, C., Vale, A., 2022. EU DEMO remote maintenance system development during the pre-concept design phase. Fusion Eng. Des. 179, 113121. http://dx.doi.org/10.1016/j.fusengdes.2022.113121.

Dutta, P., Rastogi, N., Gotewal, K.K., 2017. Virtual reality applications in remote handling development for tokamaks in India. Fusion Eng. Des. 118, 73–80. http://dx.doi.org/10.1016/j.fusengdes.2017.03.047.

EUROFusion, 2023. EUROFusion European Research Roadmap to the Realisation of Fusion Energy. https://www.euro-fusion.org/eurofusion/roadmap/. (Accessed: 2023-04-10).

Gatto, L.B.S., Mól, A.C.A., Luquetti dos Santos, I.J., Jorge, C.A.F., Legey, A.P., 2013. Virtual simulation of a nuclear power plant’s control room as a tool for ergonomic evaluation. Prog. Nucl. Energy 64, 8–15. http://dx.doi.org/10.1016/j.pnucene.2012.11.006.

Henrique da Silva, M., Cotelli do Espírito Santo, A., Marins, E.R., Legey de Siqueira, A.P., Mol, D.M., Carlos de Abreu Mol, A., 2015. Using virtual reality to support the physical security of nuclear facilities. Prog. Nucl. Energy 78, 19–24. http://dx.doi.org/10.1016/j.pnucene.2014.07.004.

Ibarra, A., et al., 2019. The European approach to the fusion-like neutron source: the IFMIF-DONES project. Nucl. Fusion 59, 065002. http://dx.doi.org/10.1088/1741-4326/ab0d57.

Katsumi, H., Yuuki, K., Masashi, T., 2020. Simplified virtual reality training system for radiation shielding and measurement in nuclear engineering. Prog. Nucl. Energy 118, http://dx.doi.org/10.1016/j.pnucene.2019.103127.

Keller, R., Eckert, C.M., Clarkson, P.J., 2006. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? Inf. Visual. 5, 62–76. http://dx.doi.org/10.1057/palgrave.ivs.9500116.

Liu, H., Qin, H., 2022. Perceptual self-position estimation based on gaze tracking in virtual reality. Virtual Real. 26, 269–278. http://dx.doi.org/10.1007/s10055-021-00553-y.

Loving, A., Crofts, O., Sykes, N., Iglesias, D., Coleman, M., Thomas, J., Harman, J., Fischer, U., Sanz, J., Siuko, M., Mittwollen, M., 2014. Pre-conceptual design assessment of DEMO remote maintenance. Fusion Eng. Des. 89 (9), 2246–2250. http://dx.doi.org/10.1016/j.fusengdes.2014.04.082.

Martínez, H., Fabry, T., Laukkanen, S., Mattila, J., Tabourot, L., 2014. Augmented reality aiding collimator exchange at the LHC. Nucl. Instrum. Methods Phys. Res. A 763, 354–363. http://dx.doi.org/10.1016/j.nima.2014.06.037.

Mol, A., Jorge, C., Couto, P., Augusto, S., Cunha, G., Landau, L., 2009. Virtual environments simulation for dose assessment in nuclear plants. Prog. Nucl. Energy 51, 382–387. http://dx.doi.org/10.1016/j.pnucene.2008.04.003.

Podadera, I., et al., 2021. The accelerator system of IFMIF-DONES Multi-MW facility. In: 12th International Particle Accelerator Conference. http://dx.doi.org/10.18429/JACoW-IPAC2021-TUPAB211.

Tesini, A., Rolfe, A., 2009. The ITER remote maintenance management system. Fusion Eng. Des. 84, 236–241. http://dx.doi.org/10.1016/j.fusengdes.2008.11.029.

Wang, J., Phillips, L., Moreland, J., Wu, B., Zhou, C., 2015. Simulation and visualization of industrial processes in unity. In: Proceedings of the Conference on Summer Computer Simulation. Society for Computer Simulation International, pp. 1–7.

Williams, A., Sanders, S., Weder, G., Bastow, R., Allan, P., Hazel, S., JET EFDA contributors, 2011. Evolving the JET virtual reality system for delivering the JET EP2 shutdown remote handling tasks. Fusion Eng. Des. 86, 1898–1902. http://dx.doi.org/10.1016/j.fusengdes.2011.01.127.