A Customizable VR System Supporting Industrial Equipment Operator Training

Franca Giannini[1], Katia Lupinetti[1], Marina Monti[1], Yuanju Zhu[1], Luca Mantelli[2], Sara Anastasi[3], Giuseppe Augugliaro[3] and Luigi Monica[3]

[1]IMATI, Consiglio Nazionale delle Ricerche,
[franca.giannini][katia.lupinetti][marina.monti][yuanju.zhu]@ge.imati.cnr.it,
[2]TPG, DIME, Università degli Studi di Genova luca.mantelli@edu.unige.it,
[3]Istituto Nazionale Assicurazione contro gli Infortuni sul Lavoro,
[s.anastasi][g.augugliaro][l.monica]@inail.it

Corresponding author: Franca Giannini, franca.giannini@ge.imati.cnr.it

Abstract. The use of VR immersive systems for the training of operators in various industrial contexts is becoming more and more common thanks to the possibility of simulating a great variety of situations in safe conditions, which can hardly be experienced in reality without risks. One of the main issues in the development of such training systems is the cost for the creation of alternative learning scenarios, which normally requires the simultaneous involvement of the training expert for its design and the IT specialist for its realization in the VR environment. To reduce this effort, this paper proposes a methodology to allow the training expert to directly specify the training scenarios using familiar tools. The method is devised considering the training of steam generator operators, but it can be easily adapted for other industrial equipment.

Keywords: Virtual reality, smart objects, simulation, training
DOI: https://doi.org/10.14733/cadaps.2023.716-730

1 INTRODUCTION

Regardless of the increasing attention to safety aspects and the development and integration of automation and remotely controlled processes, the number of worker accidents is still high. Work environments involving hazardous materials or high-risk exposure maintain an intrinsic danger that can be reduced but not canceled. The majority of the accidents occurring in the workplace are caused by human errors, mainly due to inadequate maintenance of the working tools, lack of knowledge and underestimation of the risks related to the activities performed. Then it is crucial that workers gain the necessary skills to minimize the probability of error and malfunction of the equipment they use. To this aim, current regulations require participation in specific training courses aimed at creating awareness and knowledge of the procedures to follow. In addition, in various countries it is mandatory to obtain a qualification degree for the management and maintenance of dangerous equipment. This is the case of Italian regulations. In the case of steam and superheated boilers, an on-job learning period is also required. This learning-by-doing activity is inherently dangerous and is generally limited to carry out procedures for the daily operations and periodic maintenance of the equipment. Therefore, it is not appropriate to teach how to deal with critical situations, emergencies, malfunctions, or with the consequences of incorrect operations, being generally (and hopefully) limited to theory and thus lacking effective feedback in practice. Therefore, exploiting the possibilities provided by the most recent technological developments allows the simulation of physical, chemical, and mechanical phenomena, but also the creation of immersive virtual reality (VR) to experiment a great variety of situations in realistic and safe conditions [10].

The use of VR immersive systems for the training of operators in various industrial contexts is becoming more and more common [1,5]. In some contexts (e.g., aeronautics, military, medicine) simulation has been considered for several decades as an indispensable alternative to learn and to experience particular activities, such as dealing with unusual and potentially dangerous emergencies. Simulations offer the advantage of creating a safe environment in which it is possible to actively study and teach by exploiting errors as learning opportunities. In a virtual environment it is possible to simulate situations at different degrees of complexity, thus favoring the realization of a structured learning path, in which it is easy to access demonstrative, technical and regulatory material while carrying out operations. The ability to link the actions and changes of the parameters of the working tools to mathematical algorithms capable of simulating the actual activity of the process drastically expands the set of possible situations and the quality of learning. Despite these advantages, the creation of such a training system is still challenging: developing a VR-based Training System requires a truly multidisciplinary approach [9] demanding expertise in various fields, such as dynamic modelling of processes, programming tools, and subject domain experts. Thus, a fundamental requirement of such a training platform is the ease of configuration to adapt the virtual environment to new learning scenarios, leaving to the expert the possibility to specify new training paths using familiar tools and minimizing the need to reprogram the system. This is particularly crucial when models of different equipment are considered. This is the case, for instance, of steam boiler operators, who must be trained on devices of different capacities to obtain the related qualification degree in Italy. To simplify the configuration of the virtual environment and the definition of the training scenarios, this paper proposes a methodology that allows the expert to directly specify the training scenarios using familiar tools. The method is devised considering the training of steam generator operators and verifiers, but it can be easily adapted for similar industrial equipment.

2 RELATED WORKS

The problem of defining training scenarios and monitoring/evaluating the student's performances in terms of correct operation sequence is a well-known problem in serious gaming development. Various approaches have been applied to specify scenarios and sequences with the aim of reducing the programming effort and making their definition by domain experts possible. Among them, the use of Petri net has been widely adopted [2,14]. It offers the advantage of being a kind of interface between experts and programmers. It is a graphical representation reflecting its formal features, that can be converted into programming code. Major limitations of Petri Nets [2] refers to the fact that they fit well only for stories that are preset evolving in parallel that require the author to specify all branches in advance. For a simpler configuration an approach based on state machine may be more efficient [4,9]. [4] presented a framework for behavior and scenario control based on communicating hierarchical concurrent state machines. The method is free of order dependencies and is demonstrated to be usable for modelling the reactive behavior of autonomous agents and for directing them to produce desired situations. To face the exponential problem of proliferation of specifications in case of alternative configurations, authors in [4] extended the model to include hierarchies of concurrent state machines. [9] claims that modeling the behaviors of 3D objects with advanced state-machines appears to be an efficient solution that allows complex descriptions and the reuse of behaviors. Describing scenarios with constraints and goals is more flexible than a complete description of sequences. It allows for example the re-adaptation of objects and scenarios when a non-predicted event happens. A scenario language, both graphical and textual, named LORA (Language for Object Relation Application) was defined by Mollet and Arnaldi [8]. It gathers a set of hierarchical state machines, composed of a set of variables, steps and links between steps and is interpreted by a multi-agent engine during the Run-Time execution. To enhance the scenario design, a "create-by-doing" and tools metaphor were then devised in [6] to generate and reuse scenarios by a domain expert. Here a scenario is a sequence of actions linked according to their temporal order, which is suitable when the internal states of the objects are not enough to ensure the coherency of the environment.

Semantic modelling is recognized as a suitable mean in VR applications [3,13,15]. Ontologies and excel sheets are used in [15] for the specification of the scenarios. In [14], dynamic objects provide an abstract description of the general data needed to realize each of the components in the real-time environment, while data flows and Petri nets-based state machine are used to implement plant behavior. The trainee's task execution is evaluated in terms of the achieved dynamic object status values. Similarly to this approach, a customizable immersive system for the training of two different categories of users of steam generators (i.e. operators and verifiers) is presented in this paper. The system is based on a model of the equipment composed of reusable smart elements that encapsulate all the information necessary for user interactions and functional simulations. The functional information, expressed as the status of the elements that characterize the different scenarios, are expressed using simple spreadsheet files. The novelty of this approach is the combination of smart objects and external files, directly settable by the domain expert, for the application of the VR simulation system to new equipment, highly reducing the implementation effort.

3 A TRAINING SYSTEM FOR INDUSTRIAL EQUIPMENT OPERATORS

As training effectiveness is enhanced by providing operators with a sense of realism [11], an effective VR training system should guarantee a realistic experience both considering how the learner can interact with the equipment they must get acquainted with and the surrounding environment, and how the equipment itself behaves in response. In addition, the system should provide two different usage modalities: i) learning and ii) verification. In the first one the user should be able to get suggestions on actions to perform to achieve a given task and access to additional didactical material, possibly used during in-class course, or to information allowing them to understand the meaning of the equipment elements. In the second modality, the learner should perform the activities without suggestions, while the system should provide feedback on their correctness. Therefore, it is of outmost importance that the specifications of the training scenarios are easy not only to be defined by experts but also to be handled by the simulator itself.

To guarantee a realistic experience, the simulator exploits two connected models of the equipment: the 3D graphical model, on which the learner can interact within the immersive environment, and the dynamic model, simulating its functional behavior in response to the learner actions.

The user can interact with the equipment using gestures as they would act in reality. Supporting documentation, explanations on equipment components and suggestions on the operations sequence for the specific task are activated in alternative by voice and gaze commands or menu selection in VR.

A VR training system must be flexible to easily adapt to different scenarios involving various equipment and training tasks. Here, the term scenario refers to the specific equipment considered, the task the trainee has to perform, the status of the equipment before and after the conclusion of the task, the correct sequence of operations to complete the task. Examples of tasks are the shutdown and the startup in different conditions, i.e. first time, after fault problems, after a regular shutdown. When performing these tasks, each main operation always causes the status change of one actuator. Therefore, both the initial and the final situations can be expressed in terms of the status of each actuator and control element, whose possible values belong to a limited set. Similarly, there is a limited number of possible types of actuators, control and safety components with a predefined and predictable behavior. This makes it possible to easily adapt the system when changing the equipment on which the operator has to be trained. Based on these assumptions, the core elements to guarantee the desired flexibility are (i) an Object-Oriented organization of data representation of the actuators, control and safety components, as described in the next section, and (ii) the use of files for the specification of the equipment elements statuses and operations. The status values of these components are then exchanged between the VR simulator and the dynamic model reproducing the operation of the considered equipment.

Therefore, the VR simulator can be seen as characterized by the three functional blocks depicted in blue in Figure 1. The Scenario generator has the task of instantiating the configuration of the 3D scene (equipment and environment in which it is inserted) and the status of the equipment in which the learner has to carry out his actions. The configurations must differ in the various work sessions in order to ensure the analysis of multiple scenarios. The VR Environment manages all aspects of visual and sound rendering of the scene and user actions, and allows the user to interact with the virtual environment and with the simulator features; finally, it communicates the changes of status of the actuators to the functional simulator. The Evaluator manages the verification of the correctness of the user's actions in the execution of the assigned tasks.

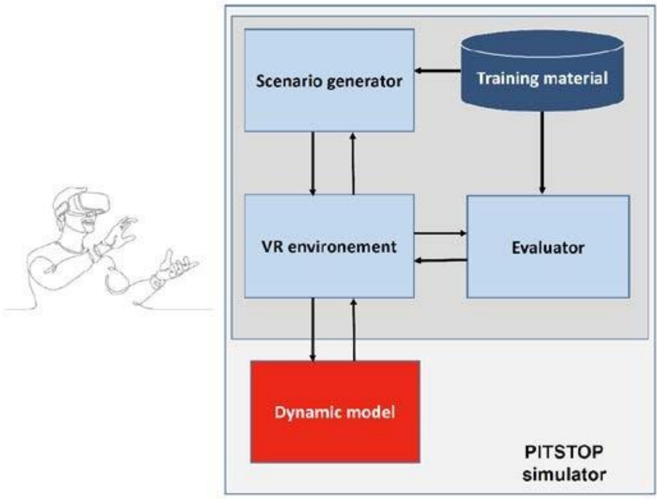


Figure 1: Organization and configurability of the PISTOP training system.

3.1 Modelling Operative Elements of the Equipment for Reuse

In the proposed approach, the key operative elements of the equipment can be active and passive. Passive components are those on which the operator can act to change the behavior of the system, indicated as actuators, or to check their working conditions. Differently, the status of the active elements, such as the control elements, is automatically changed by the system according to its working conditions. Safety elements can be both active and passive. The considered control element classes are: LED, gauge, level indicator, display. The considered actuator classes are: button, valve, switch, slider. The classes of operating elements currently considered for this study may not be exhaustive in general, and new equipment may require the definition of other classes to describe different operating elements.

All the operative elements are characterized by the following classes of data: *training data, status, behavioral data* and *shape*.

Training data refer to the material that the trainer can associate to each element to communicate useful information to the trainee for understanding the function of the element and the related regulations.

Status indicates the conditions that the element can assume, e.g. valve status indicates if they are open or close, while the status of the gauge represents the value of the monitored parameter, e.g. temperature or pressure.

Behavioral data indicate how the object is reacting to the user interaction for passive elements, or how the object must be changed according to the dynamic model results in case of active objects. Thus, behavioral data for active control elements include the status change and the possible color modification and audio effects. Behavioral data for passive elements comprise also the type of applicable user gestures and the consequent position transformation in the space.

Finally, *shape* refers to the 3D model of the element in the 3D space.

An Object-Oriented methodology has been adopted for their implementation since it allows the easy extension and exploitation of already defined characteristics through the creation of hierarchy of elements (hierarchy of classes) inheriting some core properties and functions. The hierarchy of the elements is not reflecting the functional meaning of the operative elements but their behavior in response to the user actions, as shown in figure 2.

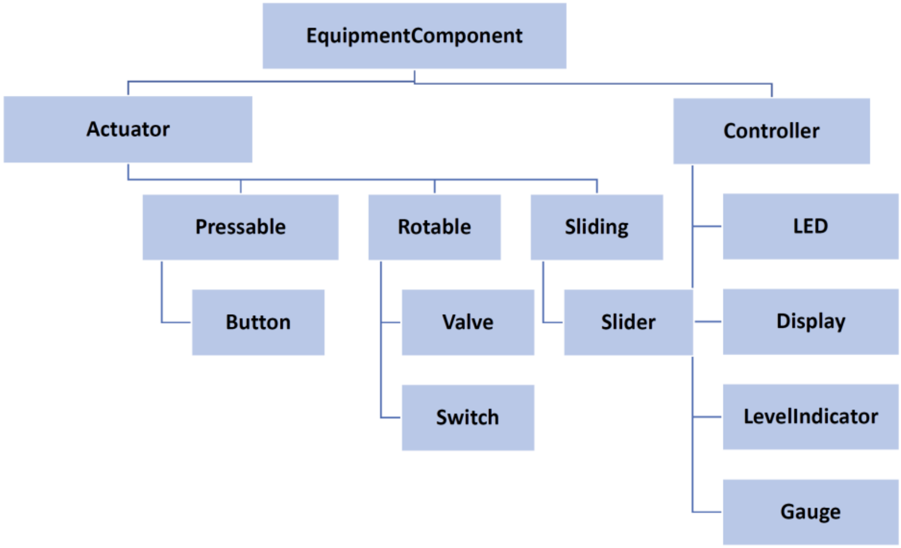


Figure 2: Class hierarchy for the specification of the actuators and control elements in the equipment.

Therefore, the characteristics shared by all the elements of a specific class have been identified and associated to the root class EquipmentComponent, which includes:

- Unique identifier,

- Text describing functional meaning and usage,
- Number of possible statuses,
- Possible status values/ranges,
- 3D digital model.

Properties of operative elements are:

- Pivot, i.e., reference system used to define the movement of the element according to the user action,
- Gestures associated to the possible user action on the element.

Then for each subclass, i.e., for any specific type of operative element, the characterizing properties and methods are specified; for instance, in the case of valve, they are:

- Rotation axis,
- Range of rotation angles,
- Correspondence between angles (i.e., position of the element) and status,
- Sound effect,
- Highlight mode when selected.

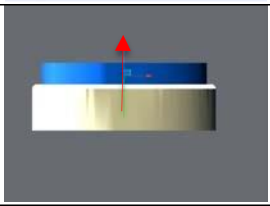
Valve		
Switch		
Button		

Table 1: Example of classes of operative elements with their Pivot.

To facilitate the reuse, predefined behaviors in response to the user gestures, sounds and highlight methods are available to be easily associated with the operation element. Table 2 illustrates the possible user's actions for the passive elements, how they can be accomplished through gestures, and the effect they have on the virtual world through their behavior. These gestures have been selected since they mimic what is normally done in reality when acting on the elements.

Element type	User action	Gestures	Element behavior
Valve	Open/close	Grasp	Change of color to indicate object selection
		Grasp_&_arm rotation	90° rotation around its axis
		Open hand	Back to the original color
Switch	Set the target position	Grasp	Change of color to indicate object selection
		Grasp_&_hand rotation	Rotation around its axis
		Open hand	Change of color to indicate the set mode
Button	Press	Move finger/hand to touch the button	Change of color to indicate object selection
		Finger_push or Hand_push	Change of position (pressed, not pressed)
		Move finger/hand away the button	Change of color to indicate the set mode
Slider	Move up/down	Move finger to touch the slider	Change of color to indicate object selection
		Mover the hand up/down	Change of position
		Move finger away from the slider	Change of color to indicate the set mode

Table 2: Gestures and behavior characterizing user's actions for the passive elements.

Gesture in Fig. 3(a) differs from gesture in Fig. 3(b) in terms of expected movement after the hand has grasped the operation element; in the first case only the hand is rotated, keeping blocked the wrist position; in the second case, the arm is moved while keeping the operation element grasped. The first applies to the change of status of switch selectors, while the second refers to the actions normally applied to valves handles. In the developed system, the grasping action is detected when the user's hand intersects the operation element game object.

In addition to the above illustrated gestures, grasp_&_translate is applicable to any movable object in the scene to grasp it and then change its position, while the entire generator can be moved changing the user's viewpoint by the both_hand_open gesture (opening both hands with their backs facing the user and then moving the hands in the desired direction of displacement).

The considered sound effects are: water_flow, steam_flow, working_engine.

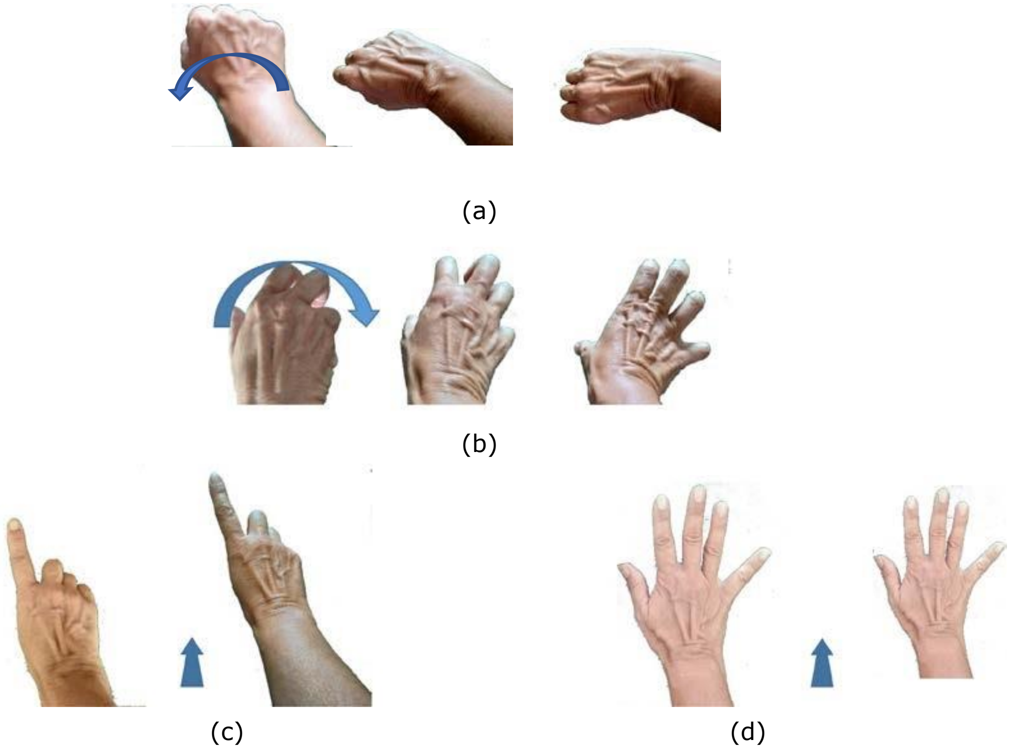


Figure 3: Examples of activating gestures for valves (a), switches (b) and buttons (c, d).

According to the considered approach, changing the equipment simulated within the training system requires: i) the creation of the corresponding 3D model, ii) the association of the pertinent class and properties to each key element 3D model. To reduce manual intervention as much as possible, the association of the 3D model (game object) to an operative element class is automatically done by the system thanks to the use of a configuration file. This file defines the association of <element identifier><name of the 3D object><operative element class>, specifying also the characterizing attributes (e.g., number of states), if possible. The name of the 3D object is the one used in the 3D model. Clearly, this approach requires all the operative elements represented in the 3D model of the equipment to be modelled as single 3D components; in this way, a single game object can be associated with a single operative element, that is a fairly reasonable requirement when defining assets to be included in a Virtual Environment whose appearance and/or position is expected to change.

3.2 Scenario Setting

The scenario generation module (see Figure 1) includes the setup of the virtual environment and the definition of the data necessary for carrying out and verifying the correctness of the target tasks.

The virtual environment setup exploits the set A of the basic elements of the room where the generator is located. In addition, for every target task to be learnt Tj, j=1…m, the inputs for the scenario generation are:

- The sets D_i of the documents required for the execution of the training activity.
- The sets B_i containing possible additional elements of the scene related to the verification of compliance with local regulations.
- The sets C_i of the statuses of the operative elements for the different possible initial configuration.
- The sets S_i of the correct sequences of operations.
- The sets F_i of the expected statuses of the operative elements for the corresponding final configurations.

The sets D_i include documents necessary for the specific training activity. Every set B_i is aimed at introducing specific elements in the equipment room, and includes instances of elements in different, possibly incorrect, positions. The set C_i specifies the values of the operating parameters and of the statuses of the equipment actuators (e.g., closed or open valve) on the basis of which the instances of the actuators must assume a predetermined color and position (e.g., valve lever in position perpendicular or parallel to the reference tube) in the scene. The specification of the initial status of the equipment actuators C_i must include both situations in which the state of the equipment is correct and situations in which one or more actuators are in statuses that cause malfunction/block of the equipment.

Each assignment can be easily described in terms of the statuses of the various key operative elements. Similarly, each action that the trainee has to perform to complete an assigned task modifies the status of one actuator. Therefore, the sequence of actions S_i can be expressed in terms of the statuses of the actuators that have to be operated in the expected order.

The initial configuration is used to provide the input to the dynamic model and to set the initial position and appearance of the actuator in the virtual reality scene. The final values in F_i, instead, are used to verify that the trainee has correctly completed the assigned task, by checking the final status values of the actuators. Similarly, the correctness of the operations the trainee is doing while changing the status of an actuator, is verified by checking if the status of the various operative elements is the one expected when acting on that specific actuator, as described in S_i. In this way, initial and final configuration can be written in the form of a row in a matrix, where the first element indicates the task, all the other elements indicate the status of an operative element (see Figure 4). Similarly, expected sequences can be easily written using (n+2)x(n+1) matrices, where n is the number of active operative elements. In this case, the first row and first column represent the identifier of an operative element, while the second row indicates the order in which the user should act on the operative element to accomplish the considered task, see Figure 5.

$OperativeElement_1$	$OperativeElement_2$	….	$OperativeElement_i$	….	$OperativeElement_n$
status value	status value	….	status value	….	status value

Figure 4: Organisation of C_i and F_i for the i-th task.

	$OperativeElem_1$	$OperativeElem_2$	……	$OperativeElem_k$	….	$OperativeElem_n$
SEQUENCE	**$order_1$**	**$order_2$**	…	**$order_i$**	…	**$order_n$**
$OperativeElem_1$	$status_{1.1}$	$status_{1.2}$	…	$status_{1.k}$	…	$status_{1.n}$
$OperativeElem_2$	$status_{2.1}$	$status_{2.2}$	…	$status_{2.k}$	…	$status_{2.n}$
…..	….	….	…..	….	….	…
$OperativeElem_k$	$status_{k.1}$	$status_{k.2}$	…	$status_{k.k}$	…	$status_{k.n}$
…..	…..	….	…	….	….	…

$OperativeElem_n$	$status_{n.1}$	$status_{n.2}$	…	$status_{n.k}$	…	$status_{n.n}$

Figure 5: Organisation of S_i for the i-th task: the values $status_{j,h}$ indicate the expected status of the j-th operative element when acting on the h-th operative element.

Therefore, the sequences to follow and the list of initial and final configurations of the actuators can be specified in an external file, where the expected status values for each meaningful operative element are indicated. The possibility of specifying setting information in configuration files provides a certain flexibility to the entire training system by allowing the domain expert to easily modify some configurations and/or expand the training scenarios avoiding software modifications when simple changes occur.

The file format should be easy to understand by the domain expert, therefore the .xls file format was chosen, since it naturally organizes content in matrix-like form, as rows and columns.

The following section illustrates the application of the described system to steam generators.

4 THE SIMULATION SYSTEM APPLIED TO THE TRAINING OF STEAM GENERATOR OPERATORS AND VERIFIERS

Within the PITSTOP project [12] the simulation system is used for the training of operators and verifiers on steam generators of various sizes and powers. Two different generators are considered to demonstrate the flexibility of the system.

The first case study of the PITSTOP project is the steam generator installed at the Savona Campus of the University of Genova (see Figure 6). To verify the effective realistic behavior of the simulator, a kind of virtual twin of this system was created.

The "virtual twin" includes a 3D model of the external shape of the equipment assembled with all the components that the trainee must be able to interact with, which are modelled as individual 3D parts. The behavior of these components in response to the trainee's action or to the equipment operation mimics the behavior of their real counterpart. The equipment behavior is simulated thanks to a dynamic model calibrated on real data from the physical equipment. The simulator is developed using the Unity game engine for the VR simulation and Matlab-Simulink for the dynamic model [7]. The simulators communicate with each other by exchanging the status of the actuators, control, and safety elements.

Figure 6: The real and the virtual steam generator used as test case in the project.

Figures 7 and 8 provide a part of the starting and final setting (see Figure 4) for the considered use case "activation of the generator". In the file, the status 0 means closed element, 1 means activated/opened element.

Start handle	Pump Mode	General Water shut-off valve	General water exhaust valve	Generator water shut-off valve	Generator water exhaust valve	Steam shut-off valve	Optical Water level shut-off W	Optical Water level shut-off S	Optical Water level exhaust
0	0	0	1	0	1	0	0	0	1

Figure 7: Part of the specification of the initial value setting for the task "*activation of the generator*".

Start handle	Pump Mode	General Water shut-off valve	General water exhaust valve	Generator water shut-off valve	Generator water exhaust valve	Steam shut-off valve	Optical Water level shut-off W	Optical Water level shut-off S	Optical Water level exhaust
1	2	1	0	1	0	1	1	1	0

Figure 8: Part of the specification of the final values of the actuators for the task "*activation of the generator*".

As an example, to check if the optical water level is correctly functioning, the two *Optical Water level shut-off* valves must be closed and the valve for the discharge should be open till the level is set to the minimum, then the three valves should be repositioned to the original statuses. Thus, the operation of closing the discharge valve and opening the two entering valves should be done after the value of the optical water level parameter is equal 0. Figure 9 shows an example of such a sequence matrix representing part of the required precedencies for turning on the steam generator of Figure 6. The same value in the second row (operation sequence order) indicates that the corresponding actuators (first rows) can be operated at the same time, i.e., there is no relation of precedence among them. This applies, for instance to the various discharge valves (i.e., *General water discharge valve, Generator water discharge valve, Optical Water level discharge*), which are the first to be closed. The NULL string in the first row (sequence specification) indicates that the actuator does not have to be operated to achieve the task, while the NULL string in the other rows indicates that the status of the component indicated in the first column of that row is not meaningful for the specific action, i.e., it does not have to be operated before the element indicated in the first row of that column. As an example, the red circled NULL value in the matrix indicates that, when operating on the *General water shut-off* valve, the status in which is the *Pump mode* (i.e., off=0 automatic=1, manual=2) is not important, while the *General water exhaust* valve should be closed, i.e. status = 0 (blue circled). This approach is suitable when, as in our case, only few actions must be done according to a specific sequence, while for most of them the order of execution is not important. This is the case of the turn-on operation, completed acting on the *Start Handle*, which requires the water line valves to be in a specific configuration. When considering a specific actuator, the cell having as the first element of the column the same first element of the row (colored in yellow) indicates the expected status to be set for that actuator. If it is required to operate more than once on the same actuator to accomplish a specific task, a matrix must be specified for each operation.

Therefore, the verification that the trainee is acting correctly on a given element can be performed by verifying if the not NULL values in the column correspond to the current values of the status of the related actuators and parameters.

	Start handle	Pump Mode	General Water shut-off valve	General water exhaust valve	Generator water shut-off valve	Generator water exhaust valve	Steam shut-off valve	Optical Water level shut-off W	Optical Water level shut-off S	Optical Water level exhaust	Remote control	Resistance Group A	....
SEQUENCE	**4**	**3**	**2**	**1**	**2**	**1**	**8**	**2**	**2**	**1**	**3**	**5**	
Start handle	1	NULL	0	0	0	0	0	0	0	0	NULL	NULL	..
Pump Mode	1	1	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	..
General Water shut-off valve	1	1	1	NULL	NULL	NULL	1	NULL	NULL	NULL	1	1	..
General water exhaust valve	0	0	0	0	0	NULL	0	0	0	NULL	0	0	..
Generator water shut-off valve	1	1	NULL	NULL	1	NULL	1	NULL	NULL	NULL	1	1	..
Generator water exhaust valve	0	0	0	NULL	0	0	0	0	0	NULL	0	0	..
Steam shut-off valve	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	..
Optical Water level shut-off W	1	1	NULL	NULL	NULL	NULL	1	1	NULL	NULL	1	1	..
Optical Water level shut-off S	1	1	NULL	NULL	NULL	NULL	1	NULL	1	NULL	1	1	..
Optical Water level exhaust	0	0	0	NULL	0	NULL	0	0	0	0	0	0	..
Remote control	1	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	1	1	..
Resistance Group A	NULL	NULL	NULL	NULL	NULL	NULL	N	NULL	NULL	NULL	NULL	1	..
....													

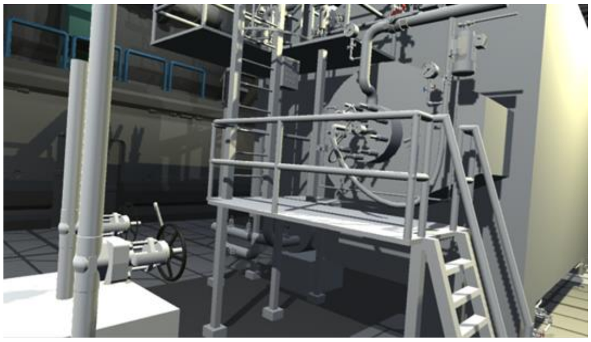
Figure 9: A part of the matrix expressing a correct sequence for the activation of the boiler.

A second case study was considered to verify the flexibility of the PITSTOP simulator to be adapted for different generators: the auxiliary steam generator installed in the power plant of Tirreno Power in Vado Ligure (Savona) Italy, see Figure 10. Keeping in mind the training purpose of this application, the least significant components of this device was not included in the model. The 3D model was generated in AutoCAD 3D modeler at the 1:1 size by using pictures and measurements taken during on-site inspections. Each operative element was created as single CAD part, with opportune reference systems corresponding to their movement pivot (see Table 1). Finally, models were saved in fbx format, readable by Unity engine. The configuration file was created specifying for each operative element i) chosen identifier for the operative element, ii) name of the 3D model, iii) related class, iv) information to be visualized to provide support to the learner, v) number of associated statuses. The file was correctly imported by the system and the corresponding classes were created allowing the expected interaction with the learner (see Figure 10.c). Scenarios were also created and successfully tested considering tasks with the initial, final and operation sequence.

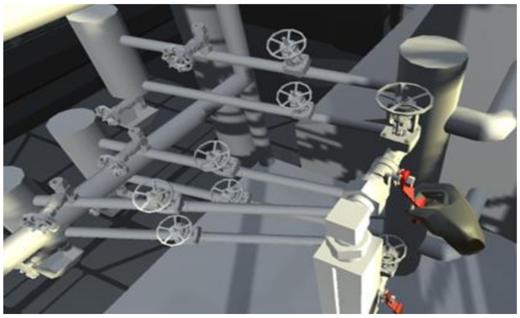
4 CONCLUSIONS AND FUTURE WORK

This paper presented an easily adaptable system for the immersive training of steam generator operators and verifiers, that allows domain experts to define new scenarios without any or little support of the system programmers. It is based on an object-oriented approach for specifying the behavior and interaction modes in VR of the elements that allow bidirectional communication between trainee and equipment; that is to naturally operate on the virtual equipment and to communicate to the trainee information on the equipment operating conditions. This allows an easier re-use of the programmed VR simulation methods when dealing with new equipment. Moreover, the proposed approach includes the use of external files, directly specified by the domain expert, to express initial and final status of the actuators involved in the assigned tasks, and the sequences of operations expected for their accomplishment. Preliminary tests carried out with the operator conducting the small steam generator test case demonstrate that the operator was able to successfully fill in the configuration file after a short explanation of the data organization. The operator also evaluated the experience as very realistic and useful for learning purposes.

(a)



(b)



(c)

Figure 10: The real steam generator at Tirreno Power (a) and its digital counterpart (b), with a zoom on one of the automatically instanced and operable valves (c).

The main strengths and limitations of this approach regard the specification of operations in terms of change of status of the elements used to control the equipment. On the one hand, the expert is not obliged to provide all the acceptable alternative sequences when an operation can be done in different moments, with the only requirement to be performed before a specific one. On the other hand, the system does not support the specification and automatic verification of actions not directly related to the actuators elements. Even if the current development is specific for steam boilers, the approach is applicable to any type of equipment, for which operations can be expressed in terms of conditions on the statuses of their key operative elements. Future work will focus on the development and integration of the dynamic model for the complex steam generator used to demonstrate the system flexibility and on a wider user testing to verify its usability.

ACKNOWLEDGEMENTS

The presented work has been partially supported by the project "PITSTOP: Piattaforma Immersiva per il Training STrutturato dell'OPeratore", funded by INAIL (Italian Workers' Compensation Authority), Bando BRIC 2019. The authors thanks prof. Alberto Traverso and Marco Ferrando for their support in the dynamic model development and the specification of the activities for the operator training and Alessandro Trastullo for the 3D modelling of the two generators.

Franca Giannini, https://orcid.org/0000-0002-3608-6737
Katia Lupinetti, https://orcid.org/0000-0002-0202-4909
Marina Monti, http://orcid.org/0000-0002-1627-3551
Yuanju Zhu, https://orcid.org/0000-0002-3913-5557
Luca Mantelli, https://orcid.org/0000-0002-4866-0302
Sara Anastasi, https://orcid.org/0000-0003-2163-6684
Giuseppe Augugliaro, https://orcid.org/0000-0001-9948-8058
Luigi Monica, https://orcid.org/0000-0001-5702-6912

REFERENCES

[1] Adami, P; Rodrigues, P. B.; Woods, P. J.; Becerik-Gerber, B.; Soibelman, L.; Copur-Gencturk, Y.; Luca, G.s: Effectiveness of VR-based training on improving construction workers' knowledge, skills, and safety behavior in robotic teleoperation, Advanced Engineering Informatics, 50, 2021. https://doi.org/10.1016/j.aei.2021.101431

[2] Brom C.; Šisler V.; Holan T.: Story Manager in 'Europe 2045' Uses Petri Nets. In: Cavazza M., Donikian S. (eds) Virtual Storytelling. Using Virtual Reality Technologies for Story-telling. ICVS 2007. Lecture Notes in Computer Science, 4871. Springer, Berlin, Heidelberg, (2007) https://doi.org/10.1007/978-3-540-77039-8_4

[3] Chevaillier, P.; Trinh, T-H; Barange, M.; De Loor, P.; Devillers,F.; Soler, J.; Querrec, R.: Semantic modeling of Virtual Environments using MASCARET, 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2012, 1-8, https://doi.org/10.1109/SEARIS.2012.6231174

[4] Cremer, J.; Kearney, J.; Papelis. Y.: MACHS: a framework for behavior and scenario control in virtual environments. ACM Trans. Model. Comput. Simul. 5(3), 1995, 242–267. https://doi.org/10.1145/217853.217857

[5] Kumar, V.; Carberry, D.; Beenfeldt, C.; Andersson, M. P.; Mansouri,S. S.; Gallucci, F.: Virtual reality in chemical and biochemical engineering education and training, Education for Chemical Engineers, 36, 2021, 143-153, https://doi.org/10.1016/j.ece.2021.05.002

[6] Lécuyer, F.; Gouranton, V.: Reuzeau, A.; Gaugne, R.; Arnaldi, B.: Action Sequencing in VR, a No-Code Approach, Transactions on Computational Science XXXVII, Special Issue on Computer Graphics Lecture Notes in Computer Science, 12230 LNCS, 2020, 57-76 https://doi.org/10.1007/978-3-662-61983-4_4

[7] Mantelli. L.; Ferrando, M.; Traverso, A.; Giannini, F.; Lupinetti, K.; Monti, M.; Anastasi, S.; Augugliaro, G.; Monica, L.: Integration of dynamic models and Virtual Reality for the training of steam generators, Proceedings of the ASME 2022 Power Conference POWER2022, 2022

[8] Mollet, N.; Arnaldi. B: Storytelling in virtual reality for training. In Z. Pan, R. Aylett, H. Diener, X. Jin, S. Gobel, and L. Li, editors, Edutainment, Lecture Notes in Computer Science, 3942, 334–347. Springer, 2006. https://doi.org/10.1007/11736639_45

[9] Mujika, A.; Oyarzun, D.; Iparragirre, I.; Dominguez, J.; Arambarri, J.: MACHS: An authoring tool to create serious games for machine-tool operator training, 9th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2011, 147-152, https://doi.org/10.1109/ICETA.2011.6112604,

[10] Nazir S., Manca D., How a plant simulator can improve industrial safety, Process Safety Progress, 34,2015, 237-243. https://doi.org/10.1002/prs.11714

[11] Patle, D-S.; Manca, D.; Nazir, S.; Harma, S.: Operator training simulators in virtual reality environment for process operators: a review. Virtual Reality 23(2), 2019, 93–311. https://doi.org/10.1007/s10055-018-0354-3

[12] PITSTOP, http://pitstop.ge.imati.cnr.it/, last accessed 2022/6/15

[13] Repetto, A.; Catalano, C.E.: A semantic layer for knowledge-based game design in edutainment applications, 2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015, 186-194. http://dx.doi.org/10.4108/icst.intetain.2015.259561

[14] Rilling, S.; Wechselberger U.; Mueller, S.: Bridging the Gap Between Didactical Requirements and Technological Challenges in Serious Game Design, 2010 International Conference on Cyberworlds, 2010, 126-133. https://doi.org/10.1109/CW.2010.60

[15] Walczak, K.; Flotyński, J.; Strugała, D.; Strykowski, S.; Sobociński,P.; Gałązkiewicz, A.; Górski, F.; Buń, P.; Zawadzki, P.; Wielgus, M.; and Wojciechowski, R.: Semantic Modeling of Virtual Reality Training Scenarios. In: Bourdot P., Interrante V., Kopper R., Olivier AH., Saito H., Zachmann G. (eds) Virtual Reality and Augmented Reality. EuroVR 2020. Lecture Notes in Computer Science, 12499. 2020, Springer. https://doi.org/10.1007/978-3-030-62655-6_8