Systems Engineering Approach for the Development of a Virtual Training Platform: Case Study in the Missile Systems Sector

Giuseppe Di Gironimo[1](✉), Sara Buonocore[1], Antonio Fariello[1], Fabrizio Carpentiero[2], Maria Rosaria Lanza[2], and Andrea Tarallo[1]

[1] Department of Industrial Engineering, JL IDEAS, University of Naples Federico II, Naples, Italy
{giuseppe.digironimo,andrea.tarallo}@unina.it, {sara.buonocore2, ant.fariello}@studenti.unina.it

[2] MBDA Italia SpA, Roma, Italy
{fabrizio.carpentiero,maria.lanza}@mbda.it

Abstract. The present work has two main objectives: the realization of a Virtual Training system about assembly operations in two immersive virtual environments and the comparison between the features offered. The first solution proposed is the use of commercial software IC.IDO, produced by ESI Group. It is designed for applications throughout all the PLM, offering support both in the design and production phases, as well as after-sales assistance. At the opposition, the second solution proposed is the use of the Open-source software Unity. This graphics engine has found significant success in gaming field in the recent years, as it allows more flexibility to the developer for the implementation of the features but requires greater competence in terms of programming. The following paragraphs firstly address the path outlined from the collection of customer's requirements to the virtual prototypes' release, following the Systems Engineering approach. Secondly, a benchmark between IC.IDO and Unity's features is illustrated to point out their main differences, strengths and weaknesses.

1 Introduction

Personnel training is a topic theme, especially in the Aerospace sector, which is subjected to particularly stringent tolerances. Personnel must be not only high-qualified, but also adequately trained with respect to the critical issues of a given assembly cycle. Therefore, several companies are currently investing many resources to find an effective and efficient training method, trying to overcome the limits of traditional methods as classroom lectures and on-the-job training. As part of a training class, the participants generally receive paper manuals with textual instructions and any explanatory 2D images. In more advanced cases, users can partially interact with video recordings of the operations or applications for tablets or computers [1]. However, these methods require a considerable effort of imagination to the participants, who are unable to fully understand

L. T. De Paolis et al. (Eds.): AVR 2021, LNCS 12980, pp. 655–669, 2021.
https://doi.org/10.1007/978-3-030-87595-4_47

the geometric and aesthetic characteristics of the components, struggling to recognize their three-dimensionality. These limits are intensified when the object of study is characterized by very low annual production volumes because it becomes strongly difficult to organize on-the-job training sessions. In addition, on-the-job training would require a slowdown or a halt to ongoing activities, causing delays and waste of resources. Once defined the technology and the added value of the project, having already clarified the involved stakeholders, it is fundamental to establish which approach will be applied to achieve the set objectives. In this case, it was found that the best solution was the Systems Engineering approach, as it supports the design team in the path that goes from receiving the customer's needs into a product that fully satisfies them.

2 State of the Art

Over the past thirty years, several experiments have been conducted to verify if there is a significant improvement with the use of Virtual Reality to implement an innovative training system. In the 90s, the research started its long journey with VR[1] technology focusing on its advantages and feasibility. In 1995 a significant project named "VADE"[2] was started by the NIST[3]: a virtual environment for assembly planning and evaluation [2]. The cases analysed were both in a small-scale as a seven parts teleprinter and large-scale assembly as a truck front-axle assembly. Overall, the system allowed real-time collision detection, simulation of dynamic behaviours of the parts held in the user's hand or constrained on the base part, dynamic interactions between the user and the parts. All the participants were mechanical engineers who had knowledge of the problem domain and of VR systems. The system was compatible with several hardware architectures, even if the experiments were primarily based on the use of an HMD[4] to obtain a full-immersion experience. This choice highlighted the difficulty related to a prolonged use of these devices: users were physically tired by the HMD. In fact, in that period, the HMDs with a satisfying level of resolution were cumbersome and heavy. On the other hand, the use of 'fish-tank' VR systems which were lighter could not offer an acceptable level of immersion. However, VADE had strong success and put the basis for many other similar projects. In fact, it was found that, after a necessary period to familiarize with the technology and with a reduced time for the use of the devices, users were fully satisfied with the virtual experience.

The following years, instead, research works became more focused on gesture recognition and haptic devices. In 2005 was released SHARP[5]: a dual-handed haptic interface for training [3, 4]. The system could operate on different VR systems configurations including low-cost desktop configurations, as Barco Baron Powerwall and four or six-sided CAVE[6] systems, allowing users to simultaneously manipulate and orient CAD

[1] VR: Virtual Reality.

[2] VADE: Virtual Assembly Design Environment.

[3] NIST: National Institute of Standards and Technology.

[4] HMD: Head Mounted Display.

[5] SHARP: System for Haptic Assembly and Realistic Prototyping.

[6] CAVE: Cave Automatic Virtual Environment. It is a virtual reality space where the walls, the floor and ceiling, act as projection surfaces to create a highly immersive virtual environment.

models to simulate dual-handed assembly operations. It was tested in several industrial applications related to maintenance and training issues, demonstrating promising results for simulating assembly of complex CAD models. SHARP demonstrated a new approach by simulating physical constraints, with collisions detecting accuracy of 0.0001 mm. Despite this, users could not manipulate parts during very low–clearance scenarios with the required precision because of the noise associate with the 3D input devices. For this reason, SHARP itself and many other projects have been focusing on crucial aspects as haptic and force feedback devices, or collision handling. An example is the experiment conducted by Shanghai Civil Aviation College in 2019 [5]. The aim of this work was to implement a Virtual training tool for aircraft maintenance operations that could track the workers' hand movements. The virtual platform selected was Unity; the hardware architecture, instead, consisted in HTC Vive Pro as visor with Leap Motion device mounted on it for hand tracking. Specifically, the system simulated two disassembly cycles of electronic components, as on-the-job training sessions were too difficult to arrange. It was found that this experience helped the users to be more self-conscious during the operations. In addition, it emphasized the fact that nowadays it is necessary to invest more on tracking technologies, not only haptic devices but also full body tracking techniques.

In conclusion, it has been found that a Virtual Training system offers the following advantages:

- User's interaction with the objects in the manual.
- Repetition of the training until satisfactory results are obtained.
- Training sessions execution regardless of work area or specific components' availability, without interfering with ongoing operations.
- User's greater awareness within the environment in terms of postures and devices for safety, without risking any accident [6].
- No permanent presence of qualified instructors, resulting in greater autonomy for users [7, 8].

3 Systems Engineering Approach

Systems Engineering is a multidisciplinary approach used for tackling complex and technologically demanding problems. Firstly, it provides for the choice of the model that will be the entire process' guideline. For the present systems, the model selected is the so-called "V-model", opposed to the "Waterfall model" [9]. Although both well structured, the peculiarity of the "V-model" is that the testing activities take place from the beginning of the project, not downstream of the implementation. It is widely used in the industrial environment, as it allows to recognize any errors much earlier, reducing resource expenditure necessary for the resolution of these. Even though it is considered by the Agile community as too simple and rigid method, it has obvious advantages: each stage is clearly defined and understandable, as well as transparent, optimizing communication between the parties involved and minimizing the risks [10, 11]. The V-model divides the process in three macro-phases: Design, Implementation and Testing. As showed in Fig. 1, the model's V-shape reflects the strict relationship between Design and Testing phases and its descendent/ ascendent trend.

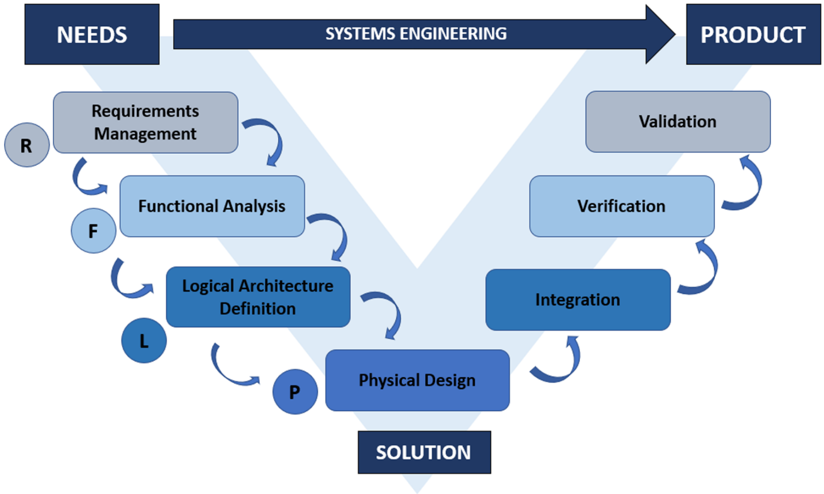


Fig. 1. V-model phases

Following the RFLP[7] path, the model starts with the analysis of the customer's requirements (R), which are secondly translated in functions that the system can offer, evaluating their feasibility (F). At this time, it is extremely useful to carry out a study in the literature to understand if solutions to identical or similar problems have already been implemented. The Design phase consists of the definition of the necessary hardware and software logical architecture (L) and ends with its physical design (P). Across the heart of the model, which is system's Implementation, the closing phase is the system's Validation and Verification. Several test procedures must be conducted to verify the correct functioning of what implemented and the compliance with the required functions: if passed, the virtual or physical prototype can be finally released.

4 Case Study

The case study selected consists in the simulation of some assembly steps of the MCU[8], courtesy of European missile and defence technology manufacturer MBDA S.p.A. This object is a military shelter, with a total weight of about 15 tons, normally mounted on trucks at a height of about 1.5 m. The MCU is designed to work in adverse weather conditions, particularly in hot and sandy areas, with heavy rain. Therefore, each sub-assembly must be electrically independent from the others and capable of working in "stand alone" mode.

Specifically, the simulation in immersive environment has involved the installation of the ACU[9], as it is a good representative of a complex assembly cycle characterized by several operations and different levels of difficulty. The operation manual was provided

[7] RFLP: Requirement, Functional, Logical, Physical.
[8] MCU: Mobile Control Unit.
[9] ACU: Air Conditioner Unit.

in digital format, with textual instructions accompanied by snapshots of 3D models and real photos in case of crucial operations. Although already detailed, the manual presents some critical comprehension issues, which are not exceeded by the only enrichment with images. Furthermore, a problem often occurred was the absence of correspondence between the textual instructions and the respective attached images.

4.1 System's Requirements, Functions and Logical Architecture

The first phase of "V-model", as previously described, is the system's Design. Originally, all the customer's requests have been collected and subsequently translated into functional requirements, excluding the ones which were not feasible in the present system. As showed in Table 1, the only remaining features have been organized into five categories described as the fundamental aspects of a Virtual Training system in the current literature [12, 13].

Table 1. Virtual training system's functions required

Category	Functions required
Realism	Perception of depth
Immersivity	Stereoscopic view
Interoperability	Active behaviour towards the objects in the scene
Scalability	Correct sequence of instructions with increasing level of detail
Mnemonic notions	Repetition of one or more operations

Specifically, for each category mentioned:

- Realism is associated with the experience lived by the user and is what allows him/her to feel completely immersed in an alternative reality. Perception of depth is considered the main feature that provides for a satisfying level of realism, in addition to realistic dimensions and colours to be applied to every element of the virtual environment.
- Immersivity is considered the accessibility of both the digital devices used and the virtual spaces. For the first ones, the digital devices must ensure that the user remains interconnected in the system. For the second ones, it is the access capacity that virtual spaces must ensure in terms of freedom in any user's movement. This aspect is fully satisfied thanks to selected hardware architecture, which is detailed in the next paragraphs.
- Interoperability is the ability to interact and exchange information and elements between distinct systems and different platforms in a continuous and as transparent way as possible. This category, for the specific case study, consists in ensuring the simulation of the following actions:

– Manipulation of objects: grasp, move, position and orient the objects.

– Coupling between two components: bring an object closer until it lodges in the seat provided in the assembly.
– Screwing: involves the coupling of a screw and a nut screw.
– Support activities: prepare equipment, wear Personal Protective Equipment, go up and down stairs, etc.

- Scalability is what allows efficient use of the system regardless of the number of users present at the same time. The gradual and successive administration of instructions, as well as the on-off visibility of objects based on their degree of involvement in the specific operation are the main features that define a certain level of scalability of the virtual system. This aspect will be examined deeply in the following description of the logical architecture.
- Mnemonic notions are the complex of various expedients devised to help the memory retain dates, technical terms, chronological lists and other notions that are difficult to associate with each other or to remember. The feature which responds to this category is the possibility to activate the repetition of specific operations to impress them in the user's memory.

The Design process ends with the definition of the software's logical architecture. The one selected is the modular decomposition: the system is divided in several modules, with the aim of creating reusable units [14]. To create a scalable system for other assembly cycles or many other applications, every module must be as more cohesive and less decoupled as possible. Specifically, cohesion measures the functional coherence, which is maximum if the module does one only thing correctly or more things highly interrelated. On the other hand, coupling measures the degree of dependence of a software's modules, which is minimum when there are only a few addictions between the modules, all functionally relevant and necessary. In fact, a software designed with a high level of coupling results difficult to understand, maintain and correct [15]. The development of a software with the highest cohesion and the lowest coupling levels is only ideal, as actually it is a trade-off between these two fundamental aspects. To address this issue, both the modular decomposition approaches have been applied using two different software. The graphics engine Unity has been selected to implement the system with the OOD, meanwhile the opponent FOD has been applied using the industrial platform IC.IDO. Starting from Unity, all the objects used must be reported explicitly, which is time-consuming but also highly reusable, easily allowing localization of changes. As showed in Fig. 2, the logical architecture designed is based on three levels that put the player at the centre. By opening the menu, he/she has two main options: interact with the objects or navigate within the virtual scene. The several Navigation modes designed can offer different experiences to the user, from the classic and more realistic walk to the flying or teleport modes, which are more rapid ways to move within the scene, but less realistic.

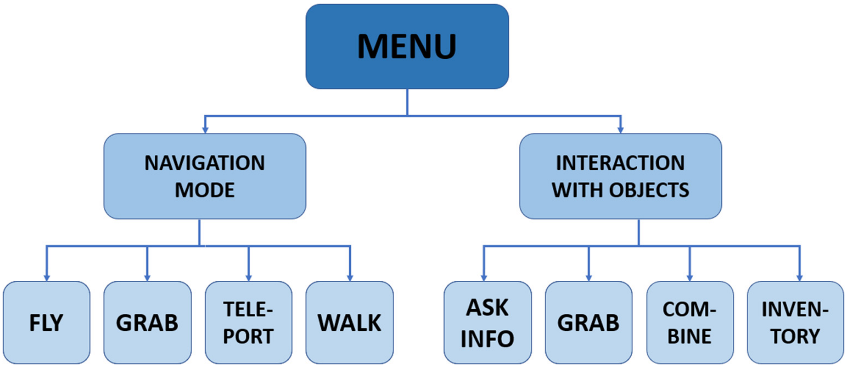


Fig. 2. OOD logical architecture of Virtual Training system

User's interaction consists in basic actions as "Ask info" about the component or "Grab" it, but also more specific as "Inventory" or "Combine". The latter has been designed for the recurrent need of combining two or more components as fastening elements, which is a preliminary action to the assembly itself. The management of an inventory, on the other hand, was designed to allow the user to save the item in a database and recall it at any time via the radial menu.

The commercial software IC.IDO, instead, has been selected to implement the FOD logical architecture, as it allows to add easily new functionalities and create independent modules, despite of the difficulty of establishing a format of interaction between the units developed. The logical architecture consists in several modules, which can be seen as big boxes that are slightly linked to each other but contain many common features. For example, Fig. 3 shows how the first three steps of a specific assembly cycle are structured, providing for the same features for different operations: the simulation of the operations as well as their repetition, the reading of rules and tips, the selection between two experience modes for the same step. Every module quite stands alone, even if the user can interact with all of them in the main[10] of the software, easily switching from a module to another. The selection between "Beginner" and "Advanced" has been introduced to emphasize the advantages of this architecture, which provides for a set of multi-tasking independent modules.

In the same module, the user can simulate every assembly operation in two different modes: Beginner is suitable for those who do not know the cycle, Advanced is for users that already know the main information about the assembly cycle. The Beginner mode has been designed to allow the user to understand the subsequence of the operations and the fundamental movements to be conducted. It is a semi-passive mode that contrasts with the Advanced mode, where the degree of user interaction with the environment increases significantly, as the operations are carried out in first person by the user.

In conclusion, the same case study has been addressed and implemented in two different platform applying two modular approaches. The Object-Oriented system developed in Unity is structured in a set of modules highly specific, but strictly dependent on each

[10] Main: Principal part of a software that generally controls the other.

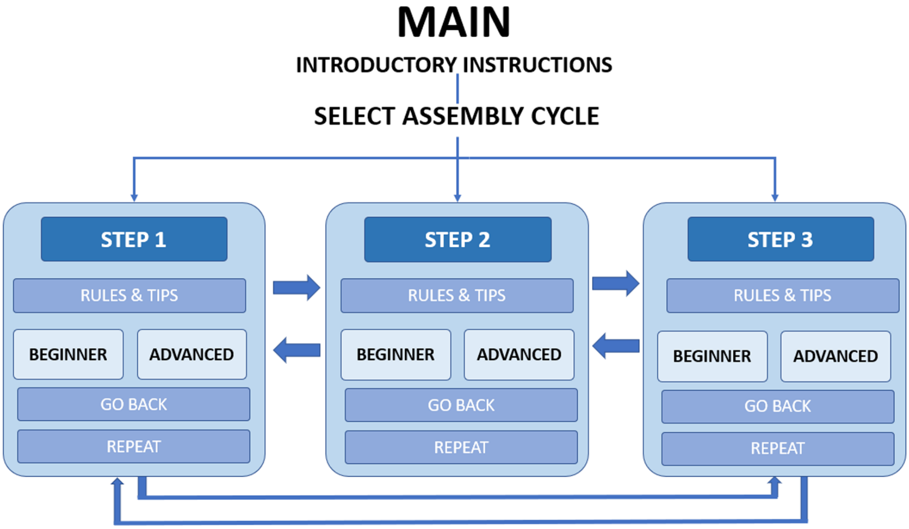


Fig. 3. FOD logical architecture of Virtual Training system

other. On the other hand, the Function-Oriented system implemented in IC.IDO provides for a set of quite independent modules, one of each often involves the same features.

4.2 Implementation Workflow for Physical Design

The heart of the V-Model consists of the system implementation phase which, although it took place in parallel in two different environments, conceptually follows the same workflow, which is presented below:

1. Analysis and modifications of CAD-models: The real work area was faithfully reproduced in the 3D design software SketchUp[11]. Subsequently, the CAD-models of the entire military shelter were viewed, checked and modified using Creo Parametric[12]. Only the components involved the simulation were selected and converted into a suitable format: ".fbx" for Unity and ".stp" for IC.IDO. Finally, the 3D models of equipment and support tools such as trolleys, small parts, tools, etc. were imported.
2. Import and organization of files in the two environments: The 3D models were imported into IC.IDO and Unity respectively and appropriately organized in a clear and recognizable hierarchical structure. Each component has been assigned an initial reference position.
3. Attribution of geometric and aesthetic characteristics: Each component has been attributed aesthetic characteristics as colours and textures to obtain a satisfying level of realism. In addition, preliminary actions as scaling, rotation and translation were carried out on the objects in the scene.

[11] SketchUp: 3D modelling software produced by Trimble.
[12] Creo Parametric: 3D modelling software produced by PTC.

4. Creation of Controller-Object interactions: During this step, all objects' offline and online animations have been created. The offline animations consist in the attribution of a discrete number of intermediate positions to be interpolated. For online animation, which are all the movements carried out by the user himself, the paths followed are partially different. In IC.IDO, the components become simulation nodes which simulate objects' mechanical behaviour. The simulation nodes are related to the others through constraints, generated after the snap[13]. On the other hand, in UNITY, characters, properties interface elements or any interactive content must be defined as Game Objects to be involved in simulation [13]. Specifically, they serve as containers for components which can be activated or deactivated by special methods and classes provided by the graphic engine, managed by C# scripts.
5. Creation of User-Controller interactions: In this phase, having defined the animations, it was necessary to manage the succession of these. In both software, the activation of the subsequent instruction was possible by generating the so-called triggers[14]: for each event, the triggering action and the resulting action must be defined. As showed in Fig. 4, the selection of a specific red button causes the activation of the respective instructions.
6. Insertion of dummies and Personal Protective Equipment: The presence of dummies allows the user not only to understand which postures and safety devices are specifically required, but also to recognize himself in them within the scene. The dummies are offered directly by IC.IDO itself, instead of Unity, where external 3D models are imported into the environment.
7. Laboratory tests: The final phase was conducted in immersive mode. It consisted in the selection and configuration of the input devices to allow the user the activation of the menu.

Fig. 4. Example of the assembly operations offered by the Virtual Training system

[13] Snap: creation of a constraint when the object reaches the correct position with respect to the target.
[14] Triggers: the singular or multiple actions which determinate a specific event.

4.3 Experimental Set-Up

The selected hardware architecture consists of a workstation and an HMD with tracking system for both IC.IDO and Unity [16]. Workstation was configured with Intel Xeon E5620 Processor, a Memory of 12 GB (6x2GB) 1.333 MHz DDR3 and a 1.5 GB Quadro NVIDIA FX4800 Graphics card. The HMD selected is HTC Vive Pro, a Virtual Reality device developed by HTC and Valve Corporation. Thanks to its Lighthouse technology, the device can support a 360-degree tracking mode in a volume of a few square meters, called "room-scale". Lighthouse, developed by Valve, makes use of base stations that generate laser beams and the photoreceptor sensors scattered on the device and its controllers detect their position in the space, allowing to determine the position and rotation of the tracked object. The device uses two high resolution OLED panels and a 110° field-of-view. HTC Vive runtime supports Microsoft Windows, macOS and GNU / Linux and uses Valve's SDK, Steam VR and OpenVR.

With the same architecture, it is important to point out that only IC.IDO permit a stereoscopic view to all the participants. Unity can currently offer the stereoscopic view only to the player, meanwhile the other participants are forced to witness the session with monoscopic view (Fig. 5).

Fig. 5. Test session of Virtual Training platform developed in Unity, MARTE Laboratory

4.4 V&V: Verification and Validation.

After having implemented the Virtual Training systems, the "V-model" provides for the ascendent and closing phase which precedes the virtual prototype's release. This phase, which consists in the systems' Verification and Validation, has been entirely carried out in CESMA's MARTE Virtual Reality Laboratory, kindly put at disposition from University of Naples Federico II.

The first tests conducted are the so-called UTPs[15]. As you can see in Fig. 6, they have been carried out in both IC.IDO and Unity by the systems' designers themselves. They consist in testing the individual modules developed to verify that the architecture thus conceived can function correctly.

Fig. 6. Example of UTPs conducted within Unity

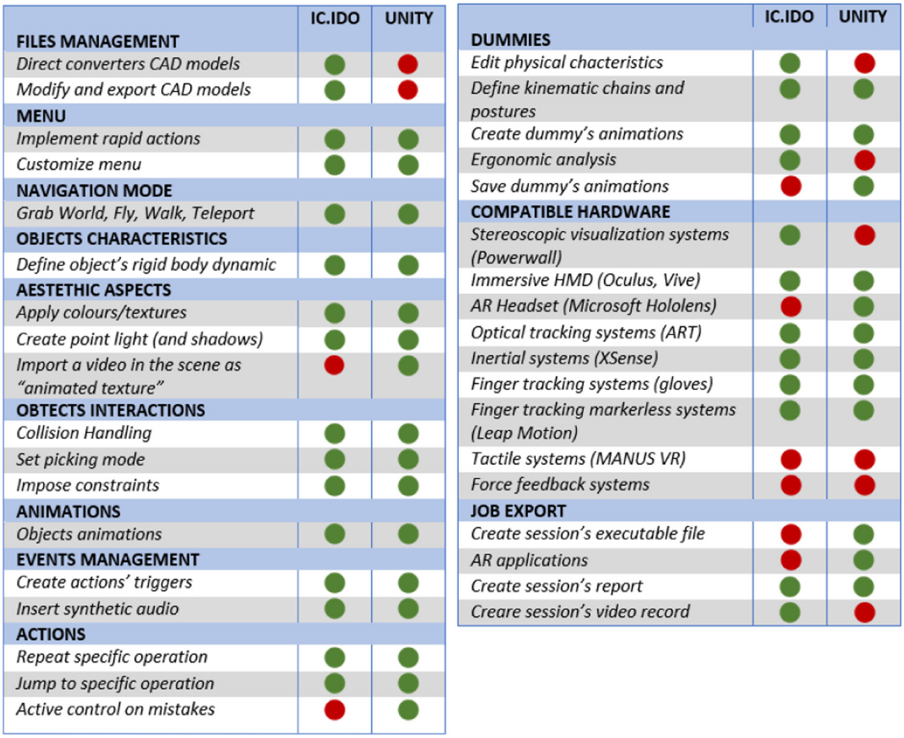
Once the UTPs have been passed, the Integration Tests have been carried out to verify how the independent units collaborate and communicate with each other, without showing anomalies or errors. Considering the V-shape of the model selected, this kind of tests has the aim of confirming that the functional requirements established have been implemented correctly. The Integration Tests have involved MBDA's experts, who have experienced both the virtual immersive experiences in first and third person. Finally, the Final Users' Test must be conducted to check if the overall requirements have been successfully satisfied, in terms of efficacy and efficiency, with a discrete satisfaction level.

5 Results: Benchmark Ic.Ido-Unity

The benchmark that follows is the result of several interviews to the participants and designers of the virtual immersive experience within IC.IDO and Unity. The Fig. 7 displays an overall comparison between them, highlighting in green the features offered and in red the ones currently not provided. The main differences in terms of features currently offered by IC.IDO and Unity will be explored in the following pages.

- File management: IC.IDO provides for any type of intervention on a CAD model imported into the scene. It allows to combine or separate different geometries and export them in different formats to exploit them even outside the software; this same possibility unfortunately was not found in Unity.

[15] UTPs: Unit Test Plans. Verification of a software's single units.



	IC.IDO	UNITY
FILES MANAGEMENT		
Direct converters CAD models	🟢	🔴
Modify and export CAD models	🟢	🔴
MENU		
Implement rapid actions	🟢	🟢
Customize menu	🟢	🟢
NAVIGATION MODE		
Grab World, Fly, Walk, Teleport	🟢	🟢
OBJECTS CHARACTERISTICS		
Define object's rigid body dynamic	🟢	🟢
AESTETHIC ASPECTS		
Apply colours/textures	🟢	🟢
Create point light (and shadows)	🟢	🟢
Import a video in the scene as "animated texture"	🔴	🟢
OBTECTS INTERACTIONS		
Collision Handling	🟢	🟢
Set picking mode	🟢	🟢
Impose constraints	🟢	🟢
ANIMATIONS		
Objects animations	🟢	🟢
EVENTS MANAGEMENT		
Create actions' triggers	🟢	🟢
Insert synthetic audio	🟢	🟢
ACTIONS		
Repeat specific operation	🟢	🟢
Jump to specific operation	🟢	🟢
Active control on mistakes	🔴	🟢

	IC.IDO	UNITY
DUMMIES		
Edit physical chacteristics	🟢	🔴
Define kinematic chains and postures	🟢	🟢
Create dummy's animations	🟢	🟢
Ergonomic analysis	🟢	🔴
Save dummy's animations	🔴	🟢
COMPATIBLE HARDWARE		
Stereoscopic visualization systems (Powerwall)	🟢	🔴
Immersive HMD (Oculus, Vive)	🟢	🟢
AR Headset (Microsoft Hololens)	🔴	🟢
Optical tracking systems (ART)	🟢	🟢
Inertial systems (XSense)	🟢	🟢
Finger tracking systems (gloves)	🟢	🟢
Finger tracking markerless systems (Leap Motion)	🟢	🟢
Tactile systems (MANUS VR)	🔴	🔴
Force feedback systems	🔴	🔴
JOB EXPORT		
Create session's executable file	🔴	🟢
AR applications	🔴	🟢
Create session's report	🟢	🟢
Creare session's video record	🟢	🔴

Fig. 7. Benchmark between IC.IDO and Unity's features

- Menu: It is one of the winning aspects of Unity, as, relying on a high-qualified programmer, it is possible to implement practically any action within the menu. In IC.IDO, this choice is more limited since it is possible to customize the menu in a predetermined list, limit that can be partially overcome if you have access to the software programming code, through the "Script" module.
- Interactions with objects: IC.IDO, thanks to the "Solid Mechanics" module, allows the definition of the dynamics of a rigid body [17]. It is possible to introduce gravity into the scene and impose predefined or customized constraints between two or more objects in the scene: this operation is also one of the necessary steps to make the snap.
- Operations: the repetition of a single operation is a time-consuming function to implement. In IC.IDO, however, it is possible to implement rapidly the repetition of an entire step thanks to the use of "States"[16]: the user can switch to another assembly step activating the respective "State". An active control on the user's errors is currently not implementable in IC.IDO. In Unity, instead, with good programming knowledge, is an achievable feature, showing a clear advantage between this software and IC.IDO.
- Dummies: thanks to the "Ergonomics RAMSIS" module, IC.IDO already provides for a library of mannequins with a well-defined and editable physical characteristics

[16] States: IC.IDO's feature that permits to "freeze" and save the entire session or only some aspects of it.

ad age, height, body sizes, nationality [18]. Furthermore, as showed in Fig. 8, it is possible to define kinematic chains that allow to impose several postures. Despite this, dummies' animations are still a critic aspect in IC.IDO, as they are achievable but do not remain in the session after saving.

Fig. 8. Example of dummy in IC.IDO

In Unity, dummies must be imported as a CAD model from other modelling software, with the need to carry out a series of very complex actions to impose a specific posture from building a skeleton to defining its kinematic chain [19].

- Compatible hardware: IC.IDO currently allows an immersive experience both on Powerwall and with helmet, in addition to sensor system for Body Tracking. It is not currently possible, but in the development phase, the use of haptic devices. Its main advantage against Unity is currently the possibility of the stereoscopic view also to the participants in third person. Despite this, Unity is the only one designed also for AR applications.
- Job export: Unity allows to export the session as an executable file, which certainly distinguishes it from IC.IDO. The latter, as commercial software, has a strong dependence on the availability of an adequate license. In contrast to Unity, IC.IDO currently offers to export a video of the session, as well as a report in ".pdf" format to always keep track of the operations performed.
- Time factor: It was found that the development of operations was certainly faster in IC.IDO as it was not necessary to design the basic architecture as well as the basic commands. The designer can concentrate almost directly on the improvement actions, putting the greatest effort on shaping the IC.IDO's pre-built features according to the specific needs. In Unity the preliminary design operations are more complex and took much more time, penalizing the number of cycle operations implemented in the same working hours.

6 Conclusions and Future Works

This paper aims to be a reference point for comparison between commercial and open-source software for VR applications.

The present Virtual Training Systems successfully passed the necessary tests conducted in MARTE Virtual Reality laboratory of CESMA. Both the systems ensure a significant level of realism and a satisfying User-Environment interaction level. All instructions are clearly and correctly administered, as no malfunctions or errors occurred during the test sessions, confirming the achievement of the first set objective. There are already several ideas for subsequent changes to improve the immersive virtual experience, as the insertion of audio and visual feedback, the assignment of a score based on time and errors to stimulate personal improvement, and the insertion of information about moments and forces to be applied.

With reference to the second objective, it was found that a universal choice between IC.IDO and Unity is not possible, as both have strengths and weaknesses. In conclusion, IC.IDO results the best solution in a wider industrial vision. It is a satisfying support tool for every company's department, offering several rapid features but not exploiting a particular flexibility to implement them. However, the significant expenditure for its licence is balanced by reduction of time spent to develop the system's architecture. On the other hand, Unity is preferrable for the implementation of a serious game[17] [20]. It offers the largest freedom and flexibility to implement every feature, with the great advantage of a low expenditure. Despite this, it requires high-qualified personnel about coding and much more working hours to implement the system.

Acknowledgments. The authors thank the University of Naples Federico II and CESMA for having putted at disposition its MARTE Virtual Laboratory to conduct all the necessary tests. A special thanks even to all those people who have actively collaborated to the success of this work, also during this critical pandemic situation. Specifically, thanks to MBDA S.p.A for having supplied, with innovation aims, the case study, ESI Group and Eng. Domenico Coccorese for their significant technical support respectively with IC.IDO and Unity.

References

1. Di Gironimo, G., Mozzillo, R., Tarallo, A.: From virtual reality to web-based multimedia maintenance manuals. Int. J. Interact. Des. Manuf. **7**(3), 183–190 (2013)
2. S. Jayaram, U. Jayaram, Y. Wang, H. Tirumali, K. Lyons P. Hart, “VADE: A Virtual Assembly Design Environment”, Washington State University and National Institute of Standards and Technology, 1999.
3. Seth A, Su HJ, Vance JM “SHARP: a system for haptic assembly & realistic prototyping”. ASME design engineering technical conferences and computers and information in engineering conference, Philadelphia, 2006.
4. Seth A, Vance JM, Oliver JH, “Combining geometric constraints with physics modeling for virtual assembly using SHARP”. ASME design engineering technical conferences and computers and information in engineering conference, Las Vegas, 2007.

[17] Serious game: games that do not have entertainment as their main purpose, designed primarily for educational purposes.

5. X. Shao, X. Wei, S. Liu, "Research On Aircraft Virtual Assembly Technology Based On Gesture Recognition", Departments of Aviation Maintenance and Aircraft Maintenance, Shanghai Civil Aviation College, 2019.
6. Tarallo, A., Di Gironimo, G., Gerbino, S., Vanacore, A., Lanzotti, A.: Robust interactive design for ergonomics and safety: R-IDEaS procedure and applications. Int. J. Interact. Des. Manuf. **13**(4), 1259–1268 (2019)
7. Di Gironimo, G., Patalano, S., Tarallo, A.: Innovative assembly process for modular train and feasibility analysis in virtual environment. Int J Interact Des Manuf **3**, 93–101 (2009). https://doi.org/10.1007/s12008-009-0066-8
8. Di Gironimo, G., Lanzotti, A.: Designing in VR. Int J Interact Des Manuf **3**, 51–53 (2009). https://doi.org/10.1007/s12008-009-0068-6
9. TutorialsPoint, "SDLC - Waterfall Model", 2018.
10. TutorialsPoint, "Modello V SDLC",2018.
11. Network Digital 360, "Agile transformation: how way of working changes to deal with digital transformation", 2021.
12. J. D. Dioniso, W. G. Burns III, R. Gilber, "3D Virtual worlds and the metaverse: Current status and future possibilities", 2013.
13. A. Comacchio, V. Dalla Libera, "La Realtà Virtuale come nuovo sistema a supporto dell'azienda competitiva, un'analisi delle opportunità." University Ca' Foscari, Venice, 2016.
14. A. Polini, "Architetture software", Software Engineering course, University of Camerino
15. P. Ciancarini, "La progettazione del software", Software Engineering course, University of Bologna
16. Coding with Unity Channel, https://www.youtube.com/channel/UCUPvKyKhh33gn0VatEjCrcQ
17. Esi Group, "ICIDO 13.2 IDO Solid Mechanics Reference Manual", 2020.
18. Esi Group, "ICIDO 13.2 IDO Ergonomics RAMSIS Reference Manual", 2020.
19. Tutorial Unity 3D https://www.unity3dtutorials.it/
20. Lanzotti, A., Vanacore, A., Tarallo, A., Nathan-Roberts, D., Coccorese, D., Minopoli, V., Carbone, F., d'Angelo, R., Grasso, C., Di Gironimo, G., Papa, S., "Interactive tools for safety 4.0: virtual ergonomics and serious games in real working contexts", (2020) Ergonomics, 63 (3), pp. 324–333.