INTERVALES: INTERactive Virtual and Augmented framework for industriaL Environment and Scenarios

Killian Richard, Vincent Havard, Jordan His, David Baudry

Version of Record: https://www.sciencedirect.com/science/article/pii/S1474034621001774
Manuscript_7335e6d7e0a9bd56243c5ea8532d05bf

INTERVALES: INTERactive Virtual and Augmented framework for industriaL Environment and Scenarios

Killian RICHARD [*,a,b,c], Vincent HAVARD [a], Jordan HIS [a], David BAUDRY [a]

[a] LINEACT, CESI, Saint-Étienne-du-Rouvray, 76800, France

[b] ENSAM, Université Art et Métiers ParisTech, Paris, 75013, France

[c] Oreka Ingénierie, Cherbourg, 50100, France

{krichard, vhavard, jhis, dbaudry}@cesi.fr

[*] Corresponding author.
E-mail addresses: krichard@cesi.fr, killian.richard@viacesi.fr, killian.richard@ensam.eu (K. Richard).
Permanent work address: LINEACT, CESI, 80 avenue Edmund Halley, 76800 Saint-Étienne-du-Rouvray, France (K. Richard).

Abstract

One of Industry 4.0's greatest challenges for companies is the digitization of their processes and the integration of new related technologies such as virtual reality (VR) and augmented reality (AR), which can be used for training purposes, design, or assistance during industrial operations. Moreover, recent results and industrial proofs of concept show that these technologies demonstrate critical advantages in the industry. Nevertheless, the authoring and editing process of virtual and augmented content remains time-consuming, especially in complex industrial scenarios. While the use of interactive virtual environments through virtual and augmented reality presents new possibilities for many domains, a wider adoption of VR/AR is possible only if the authoring process is simplified, allowing for more rapid development and configuration without the need for advanced IT skills. To meet this goal, this study presents a new framework: INTERVALES. First, framework architecture is proposed, along with its different modules; this study then shows that the framework can be updated by not only IT workers, but also other job experts. The UML data model is presented to format and simplify the authoring processes for both VR and AR. This model takes into account virtual and augmented environments, the possible interactions, and ease operations orchestration. Finally, this paper presents the implementation of an industrial use case composed of collaborative robotic (cobotic) and manual assembly workstations in VR and AR based on INTERVALES data.

Keywords: Virtual reality; Augmented reality; Authoring; UML modeling; Virtual Reality Training System (VRTS); Industry 4.0

1. Introduction

In the context of Industry 4.0 and anthropocentric cyber-physical production systems [1], integration of human-centric user interfaces [2] based on extended reality (XR) technologies—such as virtual reality (VR) and augmented reality (AR)—provides cognitive aid, digital assistance, and collaborative tools. These technologies can have various applications in design, planning, execution, and industrial maintenance processes [1,3,4]. VR applications, with their immersive capabilities, create the sensation of user presence in the virtual environment (VE) and the ability to interact with the digital twin [5]. Such technologies are used in various activities in industrial processes as a cognitive aid for: product and process design and planning [1,3], remote collaboration [6], safety and ergonomic assessment of cobotic workstations [5,7], knowledge sharing and training [8,9], and user awareness of safety procedures [10]. In complement, the ability of AR to add virtual content to the physical space is used for providing visual assistance to an operator in the case of maintenance range, quality control, assembly assistance, and logistics [11–14]. Indeed, these technologies make it possible to improve the quality and the sharing of knowledge between the trainer and the learner on one hand, and between the expert and the operator on the other hand. However, creating content for these tools requires extensive company information and knowledge, professional skills (e.g., from experts in the industrial processes and trainers) [15], and strong computer development skills. Thus, it becomes imperative to define models to facilitate content authoring, whether to define the level of interaction of the environment, or to orchestrate tasks in training and activity scenarios such as assembly, maintenance, and inspection.

Several model approaches exist based on UML language, such as MASCARET [16], #FIVE [17], and others [18–20]; approaches also exist based on ontologies [21], such as HUMANS [22] and INOOVAS [23]. UML modeling allows for a structured formalism representing the environment, but also requires the expert to comply with it. Conversely, ontologies can be more easily adapted to an expert's vocabularies [24], and will let job experts express the knowledge they want to share more easily. Of course, using models with specific vocabulary and logic can complicate the formalization of complex scenarios. On the other hand, industry and IT are often confronted with a UML type of representation (or something close to UML) which facilitates its integration. Therefore, it is more efficient to adapt this existing type of representation, which is more commonly used in companies

working in an industrial context. Moreover, this allows for the implementation of XR tools with simple concepts to generate augmented and virtual environments.

In this research, an analytical review of existing models to produce augmented and virtual environments and scenarios is made according to the industry's AR and VR application needs. This analysis identifies the requirements and limitations of such models. Some existing models, although advantageous in certain ways, are unscalable when attempting to add new features (e.g., collaboration, autonomous agents). Similarly, most scenario models reviewed are dedicated to procedural scenarios (i.e., step-by-step). In general, there remains the need for a content authoring model that support VR and AR. Therefore, the main contribution of this paper is to propose a novel INTERVALES framework (**INTER**active **V**irtual and **A**ugmented framework for industria**L** **E**nvironment and **S**cenarios), which: (i) allows scenario content authoring for VR/AR applications; (ii) facilitates the transformation of existing knowledge into AR/VR compatible scenarios; (iii) is interoperable with third-party tools. Finally, the paper illustrates this framework through the implementation of an industrial use case composed of cobotic and manual workstation assembly processes in VR and AR, starting from the same model representation. An experimentation has been conducted to evaluate the authoring process through the graphical user interface.

2. Related Works and Existing Models Review

With the arrival of new VR/AR technologies in the industry, new opportunities and needs have arisen. Whether operators use VR to train themselves to work in a hostile environment in complete safety, or use AR to be assisted and facilitate machine maintenance, many use cases are already observable in both the industry and the literature. These different use cases can be grouped as follows:

- **Design, planning, or assistance tools:**
  - At design [1,3,5,6].
  - At implementation [5,15].
  - For operation and maintenance [1,3,4,11–14,23].
- **Training tools / Serious Game:** These concern applications with a training goal, using knowledge learning. Their goal is to immerse learners in an artificial environment designed to teach them the operation of a complex process [7–10].
- **Demonstration tools / Communication support:** These concern applications with a pure visualization interest (e.g., animation of a 3D model), a demonstration use, or serving as communication support [10].

Both virtual reality and augmented reality are critical in this context, and specific needs can appear among these different types. For example, the possibility of working with several agents, whether in a collaborative application with several operators for the layout of a workshop [6,25], or as part of the ergonomic design of a workstation or an assembly training with an autonomous agent, like a robotic arm [5]. This can also include the possibility of providing more freedom and involvement to users in the scenario by allowing them to make mistakes [8] or attaching consequences to their actions (i.e., the causality principle) [22]. To answer the needs of these advanced environments, certain models and frameworks can also include other characteristics, such as role models [26], learner tracking models [22], or multi-device compatibility. In the context of our proposal, the next section mainly focuses on environment and scenario models.

2.1. Models and Framework for Virtual and Augmented Reality

Prior research has already proved and validated the effectiveness of virtual reality in terms of learning, training, and positive impact on performance [27–30]. Moreover, augmented reality has demonstrated efficiency in terms of support and guidance performance [12,13,31]. AR primarily depends on the notion of presence, the interactivity of the virtual and augmented world and the way it was designed to

share knowledge. That is why particular attention has been focused on the authoring of the virtual and augmented environment, as well as the actions feasible within it. As explained by Nebeling et al. [32], there is “a lack of tools to quickly and easily prototype and test new AR/VR user experiences.” It is also interesting to note in the context of development that, despite advances in research, developers still frequently observe an offset between proposed solutions in literature and reality within development companies. This is mainly due to authoring processes for virtual and augmented environments that differ in terms of developers' experiences, companies’ policies, and the time available for each project. In the context of production, developers usually do not have the time to create features generic enough for collection within a shared framework.

The study proposed in this paper about existing model propositions is split into the following: environment model, scenario model, and framework, being a grouping of several models (Figure 1). Environment models (Figure 1: Environment model), are divided in three categories: “behavior-oriented”, “synoptic objects” oriented, and “objects-relations” oriented. There are reviewed in 2.1.1. Scenario models (Figure 1: Scenario model) can allow three levels of freedom in terms of scenarization: procedural (step-by-step), opened scenario with objectives (with guidelines), and completely opened scenario (simulation). Scenario models belonging to different frameworks are reviewed in 2.1.2. In addition, several characteristics identified from the industrial needs discussed in the previous section will be considered for each model, including the targeted technology (AR or VR), whether they mention collaboration or autonomous agent compatibility, and the causality principle.

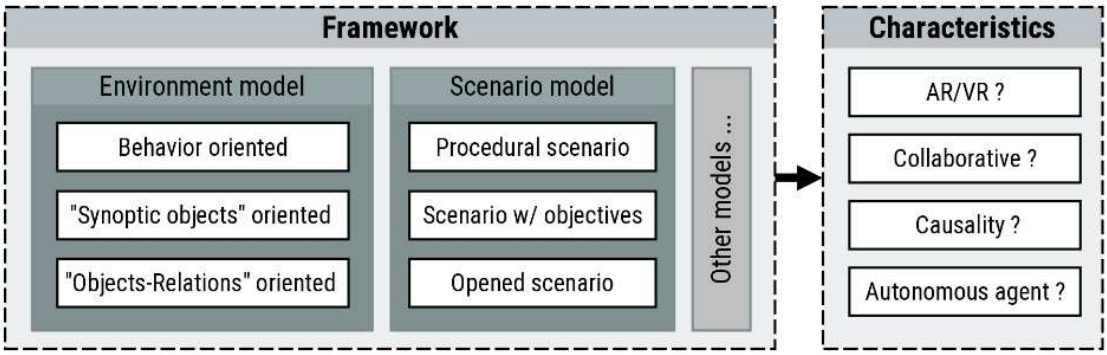

Figure 1: Models organization used for their comparisons

2.1.1. Environments Models

The first goals of an environment model are to format the way virtual and augmented environments and scenarios are described, provide developers and final users with a usable framework, and improve authoring time speeds. Once the model is created and establishes reusable concepts, it becomes easier to produce scenarios for developers; the process also becomes easier for an expert without developmental knowledge via a simplified interface. For instance, experts must be able to set up interactions easily and in a way that allows them to efficiently produce scenarios to share knowledge. Frameworks such as UMI3D [33] or D3PART [34] are good examples of the trend to ease the authoring process from the developer side.

In terms of AR, an important part of industrial digitalization [35], several works have already been developed for authoring content. Friedrich et al. [36] proposed a web-based architecture and AR model for authoring AR content step-by-step. Lee et al. [37] then added a context-awareness that adapts the AR application to the user’s specific experiment and tools. It enables the system to display augmentation to repair a car depending on the device used and its capacities (e.g., sensors available). In the same way, Su [38] proposed an architecture associated with a UML model to describe Operations and Maintenance (O&M). However, this model is not generic enough because it does not split the action from the object on which it acts. Zhu [39] thus improved Lee’s approach [37] by proposing an authoring model accessible not only to developers, but also to domain experts and

operators. Then, Martinez [19] modularized the concepts to propose a more flexible approach. This approach allowed for the implementation of different modules. There are nine modules total, including, for example, Marker recognition, Computer Vision, Optical Character Recognition (OCR), and rendering modules. Moreover, they can be updated according to technology changes. Martinez [19] also defined three crucial concepts: that maintenance and job represent a group of tasks; that a job can be reused in several maintenances; and that the task is the smallest gesture done in a maintenance. This prevents AR authors from duplicating the described procedure in each maintenance. Moreover, the Job only needs to be modified once to update at each Maintenance. The main drawback to these definitions is that the Task concept does not split an action (like unscrew) from the object on which it acts. Therefore, there is no template task that the expert can reuse in every operation. Havard [23] enhanced the proposition made by Martinez [19] by splitting the action description from the object acted upon. Moreover, Havard's proposal makes it possible to describe an augmented reality procedure in a virtual reality procedure with only one edition. Although modeling for AR authoring can be complex, it is even more complex for VR authoring, as the following paragraphs will discuss.

For VR, a virtual reality learning environment (VRLE) or Virtual Reality Training System (VRTS) model defines how actions within it function and how learners, virtual objects, and interactions are managed. Thus, environment models are organized in several ways to represent interactive virtual learning environments, and into several types of models with their own logic. In the literature, they are regrouped into three main ways of representing VE and interactions.

Behavior-oriented models:

Firstly, the interaction within a virtual environment can be represented without a specific entity, and instead as a behavior or function. Among these behavior models are HCSM [40] and HPTS++ [41]. Both use hierarchical state machine formalisms (see Figure *2*). Their advantage is in their ability to observe the environment state and react to any action. However, because those frameworks use specialized entities for objects and autonomous or real agents' interactions, it is necessary to adapt the behavior for each new virtual object or agent, and further additional development is required to make them work with the existing environment.

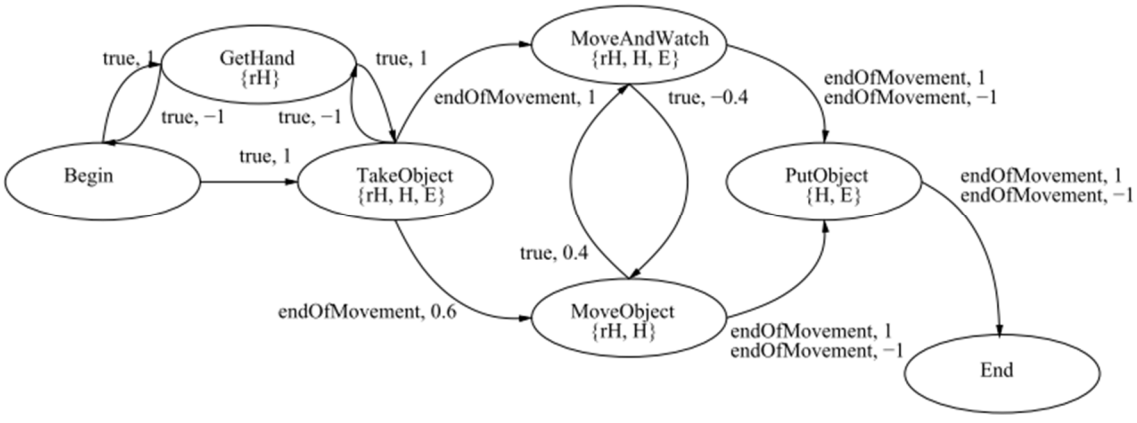

Figure 2: Example of a state machine for moving object behavior (HPTS++ [41])

"Synoptic objects" oriented models:

Secondly, possible interactions can also be contained in the objects themselves. These "synoptic objects" use the virtual object as its own action manager. When an actor wants to interact with an object, it offers a list of what interactions it can afford. In Starfish [42], for example, object interactivity is based on two simple components: actions and interactive surfaces. Actions are either predefined "primitive atomic actions" (for example, moving, giving, taking, or speaking), or a composition of these predefined actions. Interactive surfaces are commonly a volume per area, which

defines the parts of the object that correspond to the execution of an action. This object-oriented approach is very effective when it comes to creating single-object actions, like open/close a door or activate buttons. However, the limits of this become obvious when multiple-object actions, like assembly operations, are involved. In those cases, each new actions should be defined on each existing objects according to each new possible combination. Moreover, the process becomes even more difficult in collaborative applications for multiple instance management.

"Objects-Relations" oriented models:

Thirdly, interactions can be represented based on a combinatorial logic called "Objects-Relations". By adding another entity called "Relations" in addition to objects, the model manages to represent multiple object actions. A relation consists of behaviors involving different objects, with preconditions and conditions evolving during execution. If a condition is reached, behaviors change the object's state [43]. A change in condition can be induced from the definition of the object's "Type" involved in the relation, by the object's state (for instance, a property modification), or by its placement within the virtual environment. This allows for a more flexible and generic way to describe interactions, further simplifying multiple instance management. Indeed, some "Objects-Relations" models can even manage several actors (real and virtual) interacting in a common environment—STORM [44], for instance, has this ability. In scholarship, frameworks like MASCARET [45] or HUMANS [22] integrate their "Objects-Relations" oriented models (VEHA and Domain-DL, respectively).

VEHA is a UML-based model that enables representing and describing available entities and interactions inside the VRTS (e.g., types, structures, behaviors, their logical relationships with each other, and topologies). It also allows the representation of common types of relations, such as physical relation or object composition relation, like a book inside a desk. The composition relation can be used to represent an object as well as a composition of different objects, which is an advantage for assembly operations.

Domain-DL is based on a set of related entities: objects, actions, behaviors, and events. They describe objects as a set of properties. The proposed formalism describes the virtual world with an ontology (based on the OWL-DL language [46]). Objects can be concrete entities (e.g., 3D models or agents) or abstract ones (e.g., objectives or tasks) that can be described or manipulated in the VE. They can also be agents able to autonomously interact with their environment. Actions allow these agents (real or simulated) to realize relations between objects only if preconditions according to the state of the virtual environment are validated. When an action is achievable, corresponding behaviors are created. Behavior can also be created by an event. Once triggered, a behavior changes the state of one or more objects.

In [17], Bouville et al. presented #FIVE, a model that focuses on two concepts: relations and interactions. The basic principle of the model rests in the definition of types of objects, which, once grouped, form patterns of objects. Then, for each pattern having one or more types, it is possible to define allowed relations between them and the associated feasible user interactions. This model also extends this idea of defining types of objects by defining roles to actors, allowing for role management. This approach thus simplifies the development of the virtual environment and makes it possible to optimize interactions as well as content authoring by categorizing the elements into a set of object types, in the same natural way that the world can be interpreted.

2.1.2. Existing Scenario Models and Frameworks

In virtual reality, as well as in augmented reality, the need to orchestrate actions is present, whether for a step-by-step procedure (e.g., maintenance or assembly guides) or even to define objectives for industrial training. Scenario complexity can be defined as either procedural (step-by-step), opened scenario (simulation), or opened scenario with guideline objectives. A scenario model must therefore offer concepts allowing the content creator, IT developers, or job experts to scenarize the environment.

Existing solutions have already proposed simple scripting concepts (pure scenarization) or more advanced concepts (with, for example, educational concepts and learner monitoring, or operator role definition).

In [26], #SEVEN provided, in addition to an environment and role models, a scenario model based on Petri Net [47]. The use of Petri Net allows for a clear representation of the scenario and how it will unfold. However, in a large-scale scenario, such as a complex industrial assembly, it can quickly make such representation convoluted. Another limit becomes clear if, in the case of an unsolicited action like a disassembly, the scenario cannot go back unless it is defined first. Similarly, in the case of a fully-opened environment with the possibility of making unforeseen errors in the base scenario, this model will require that each eventuality be provided during the scenario authoring. Furthermore, since the context of this framework is training, it is focused only on virtual reality, and does not take into account the possibility of having a common scenario for virtual reality and augmented reality.

In [22], the authors presented HUMANS as a framework using ontologies, intended for training in a virtual environment. It provides a scenario engine (mainly activity model) called SHELDON, an environment engine (world and causality), and a learner monitoring engine. It also offers advanced possibilities for autonomous avatars. The framework is extensive and modular. SHELDON allows the user to make mistakes, since it uses Zone of Proximal Development (ZPD) to ensure effective learning. The underlying concept is that the learning goals adapt and are always neither too simple nor too hard. However, although this model is complete, it is very heterogeneous in terms of representations. These different representations of the data for each engine make it necessary to go through different models to author the environment, and therefore require knowledge of each one. In addition, the representations are intended to be interpretable and business/trainer-oriented, but, from a raw point of view (i.e., without interface work), these concepts can be more complicated to understand for a non-initiated and non-technical person.

In MASCARET [45], an environment engine (VEHA) and an activity engine (HAVE) based on UML are presented. Like #SEVEN, it uses a unified that is understandable by developers in particular and easy to translate into a user interface. Like HUMANS, thanks to the proposed model, the framework provides freedom to the user by allowing the participant to make mistakes.

Regarding models intended to be used in VR as well as in AR, Havard [23] proposed INOOVAS (INdustrial Ontology for Operation in Virtual and Augmented Scenes), an ontology that aims to allow a job expert to create and edit both an augmented reality guide and maintenance training in virtual reality. This model focuses on creating content for these two goals in the context of industrial maintenance operations. The advantage of this ontology is its common expert content creation process. It allows the expert to enter the operations that the operator will have to perform only once. However, translating this ontology into a VR/AR application can be complicated by its specific data representation that is triplet oriented and not Object Oriented.

2.2. Synthesis

The following table (see Table 1) summarizes the properties of the different models studied in this section, as well as other models seen in the literature. By combining the industry needs summarized above with the existing models, key characteristics have been defined. Thus, existing models will be compared following these characteristics: if the focus technologies are supposed to be used for **AR** or/and **VR**; if they propose different levels of scenario (procedural scenario, opened scenario, or opened scenario with objectives); if they are collaborative; if they incorporate notions of autonomous agents; if they have causality principle (i.e., if the actions done have consequences); and if they propose one or more industrial use cases following the context of the presented use cases.

Thus, some trends concerning models in the literature can be observed. Models are often specific to either VR or AR, limiting their uses. There is a current lack of model pooling content authoring which could be used in both VR and AR. Analysis made in this paper can also point out that, as already known, the use cases are mainly learning and maintenance applications for AR. However, at least simple scenarios with task orchestration can be found in both AR and VR.

Table 1: Synthesis of existing models

Model	Technology? (AR/VR)	Framework						Collaborative?	Autonomous agent?	Causality?	Industrial use cases?	Use cases
		Environment model			Scenario model							
		Behavior	Synoptic objects	Objects-Relations	Procedural scenario	Scenario w/ objectives	Opened scenario					
#FIVE [17]	Virtual Reality			X				X				Surgery preparation training, interaction demonstrator
#SEVEN [26]					X	X		X	X	X	X	Learning scenarios for surgery, training for the maintenance of an injection molding machine
HCSM [40]		X							X	X		Traffic simulation
HPTS++ [41]		X							X	X		Interaction and behavior simulation
HUMANS [22]				X	X	X	X	X	X	X	X	Learning application (ARAKI: conception of virtual environment for training in maintenance activities, NIKITA, SIMADVF)
Marigold [48]			X									Generate virtual environment and interaction prototype
MASCARET [45]				X	X	X	X	X	X	X	X	SÉCURÉVI: Scenario for the training of firefighters for operational management and commandment, and GASPAR: Aviation Management On Aircraft Carriers by Virtual Reality, conduct sizing, and ergonomics studies
Smart-Objects [49]			X						X			Interactive smart desk simulation
Starfish [42]			X						X			Simple interaction demonstrator on Synoptic Object (pulling a door)
STORM [44]				X							X	Industrial Training
[19]	Augmented Reality				X						X	Generic step-by-step system maintenance in AR
[50]					X			X	X		X	Maintenance of industrial system step-by-step using Hololens
ARAUM [51]					X						X	Repair engine step-by-step with AR
ARML [52]												Describe an augmented reality scene, display AR assets in the world
COIVA [53]					X							Find objects in AR in a building for specific needs according to the user's context
Context-aware AR [39]					X			X			X	Maintenance of industrial system step-by-step and editable by operators
EASE-R³ [54]					X			X			X	Maintenance of system in guided mode or assisted by an expert
MAR [55]					X						X	Describe Target, Tracking and AR Asset to add. Used in Maintenance, Medicine, AR localization, Games. All logic is hardcoded.
OARP [38]					X						X	Car step-by-step maintenance in AR
U-Car [37]											X	Car step-by-step maintenance in AR
[56]	Both				X	X		X	X		X	Aircraft Final Assembly Line Simulator, High-Voltage Cell Simulator, Machine Tool Simulator
D3PART [34]					X			X				Collaborative arrangement of a room
INOOVAS [23]					X	X			X		X	Scenario for training on a step-by-step maintenance in VR and guiding operator in AR
UMI3D [33]									X			Collaborative construction game

Among these models, limitations can be observed. Concerning environment models, behavior-oriented models are limited by the need to readapt each object when new objects or autonomous agents are implemented, since they need to know how to interact with each other. For "synoptic object" oriented models, the main limitation is in the use of multiple object actions like assembly operations or collaborative actions. Although not without advantages, they are hardly scalable.

All things considered, "objects-relations" models offer the same possibilities as other models in terms of behavior or representation of interactive virtual worlds, but also permit additional ones (co-manipulation, simple actions that could be easily edited). Moreover, this logic is effective to represent, create, and edit interactive environments because it comes closer to the way the world can naturally be interpreted. The physical world, as well as a virtual environment, can be summarized in terms of interactions, as it is populated with objects and actors. In a VE, actors can be controlled by users or be autonomous. The virtual environment can be modified during the simulation by objects and actors via actions. Since, among all presented models, "objects-relations" models are the most complete, this direction was chosen for the model proposed in this paper.

Concerning scenario models, most of those reviewed stop at procedural scenarios (step-by-step). For those that integrate more freedom, it becomes clear that they are not mistake-friendly, containing no open scenarios and focusing only on VR. For example, the possibility of going back and undoing an action is not integrated unless it is planned in the scenario.

Finally, from this research study and by grouping the positive points as well as the limits observed, several criteria are defined for the model to be implemented. The model must:

- Allow authoring a virtual and/or augmented environment (feasible interactions, roles, consequences of interactions in the environment, a procedure to follow);
- Allow defining learning objectives and scenarios with an orchestration of objectives;
- Disregard the display device used or the number of agents, if they are controlled by a real person or an autonomous agent (such as a collaborative robotic arm), or the display device used must guarantee its usability and scalability.

The contribution of this paper is to propose a framework based on UML model that answers those criteria and will facilitate environment establishment in VR as well as in AR, from simple step-by-step procedures to complex industrial scenarios involving more global objectives. The framework and model proposed will especially focus on opened scenarios with objectives.

3. INTERVALES Framework: Architecture and Model Proposed

Based on the "Entity-Feature" methodology presented below, the approach proposed in this paper focuses on responding to Industry 4.0 needs for complex assembly or maintenance operations, while also answering the criteria identified in 2.2 and remaining AR/VR compatible. Thus, we propose **INTERVALES**: **INTER**active **V**irtual and **A**ugmented framework for industria**L** **E**nvironment and **S**cenarios.

The proposed environment framework is the result of theoretical analysis of the different existing models in the field of VRTS and augmented reality presented above, combined with previous research focusing on industrial use cases of different pedagogical and training situations for maintenance and assembly operations in virtual and augmented reality [5,13,15,20,25]. As a result of this analysis, the framework has been decomposed into several modules in order to satisfy the identified requirements:

- Describe the features of each entity in the scene;
- Define doable actions and parameters associated with each one when interacting with entities;
- Manage and check scenarios during the training and export scenario and scene state in order to extend the INTERVALES framework with external tools.

So as to remain generic, INTERVALES meta-model modules are formalized in UML. In this section, the paper first presents the INTERVALES architecture that allows, on one hand, scenarios or procedure authoring and, on the other hand, use of them in AR or VR. Then, the various modules of the INTERVALES framework are presented and the UML data model associated is described. Finally, in the third part, the paper presents the implementation of use cases of an industrial shop floor use case about an assembly on manual workstations in VR and in AR.

3.1. INTERVALES Architecture and Workflow

The INTERVALES architecture has been built (i) to satisfy content and scenario authoring by IT and job experts or trainers, and (ii) to deliver training in VR or guidance in AR to end-users such as operators, trainees, or apprentices.

3.1.1. INTERVALES for Content Authoring

From the perspective of authoring (see Figure *3*), INTERVALES can be used by two different roles: IT developers or domain specific experts/trainers. The following sections first explain the use of the framework by IT specialists, then illustrate how those functions can be extended by a domain specific expert or trainer.

All AR or VR applications rely on assets, mainly 3D models, in which features are added to enable users to interact with them and to make the objects interactable. To reach this goal, INTERVALES relies on several modules for authoring scenarios. The first of these is the **Entity-Feature module**. This module allows for defining each asset's characteristics as a result of the Entity and Feature concepts. The Entity represents the 3D object, and is composed of Feature in order to define how it should behave. Section 3.2 will explain these concepts in more detail. Grounded in those two concepts, the **INTERVALES interaction module** is responsible for managing all interactions made by the user on Entity and Feature. This module is able to take into account the devices used (e.g., VR controller or mouse and keyboard) in order to adapt interactions made by the content and scenario authors. When interacting with the Entity, and based on the Features it contains, the INTERVALES framework dispatches events about an Action done on one Entity/Feature or Relation created between two Entity/Feature. The **INTERVALES Scene state module** (see Figure *3*) is responsible for gathering all Action and Relation done between Feature placed on each Entity.

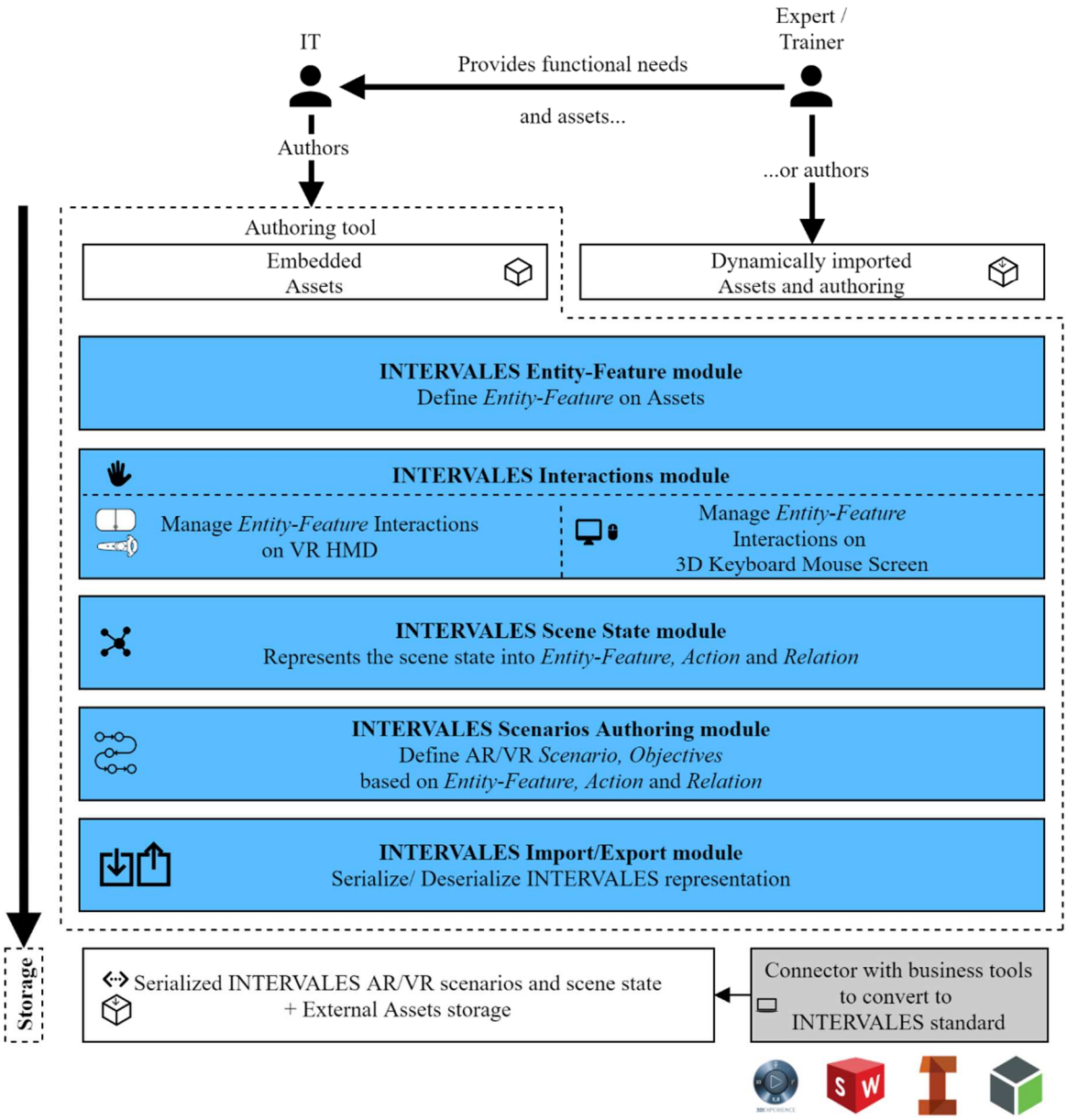

Figure 3: Authoring functions of INTERVALES architecture

Based on the status of the scene, the **INTERVALES Scenarios Authoring module** allows IT developers or experts/trainers to define scenarios thanks to objectives based on the actions the end-user must perform. Once defined, the **INTERVALES Import/Export module** is responsible for serializing/deserializing scenarios and scene states into a standard file based on the model described in section 3.2. Moreover, as the scenarios and scene states are based on a formalized UML model, external tools, such as Computer-Aided Design (CAD) and Product Lifecycle Management (PLM) tools, based on a specific plugin development, could provide scenarios authoring in an INTERVALES standard (see Figure *3*: Connector with business tools).

Once scenarios are defined, they are used to train people or assist workers. The next section describes how exported scenarios are consumed by the INTERVALES AR or VR player.

3.1.2. INTERVALES for Playing Scenarios in Augmented or Virtual Reality

The AR or VR player based on INTERVALES is used in the opposite order to the authoring tool, from storage to the final user. The player starts by taking the scenarios into the storage place. Then the **INTERVALES Import/Export module** (see Figure 4) manages the conversion from the serialized

scenario into a scene state and objectives for the user. From there, the user can interact with the VR scene. The **INTERVALES Interactions** and **Scene State modules** convert each interaction into an Action/Relation between Feature and Entity. Working from this representation, the **INTERVALES Scenarios Checker module** compares the Objectives defined by the authors to the real Scene State and Actions/Relations performed by the user. As each specific Objective represents a specific Action/Relation with tolerance value, the Scenarios Checker module ensures that each Action made by the user corresponds to an Objective planned in the scenarios. Thus, it validates that the scenarios defined are correctly performed by the end-user (see Figure 4). Example will be detailed in section 3.2.3. From the AR perspective, a scenario is a succession of steps that the end-user must perform on the physical system. Each Action/Relation is represented by an animated augmented reality object to assist or guide users in their task.

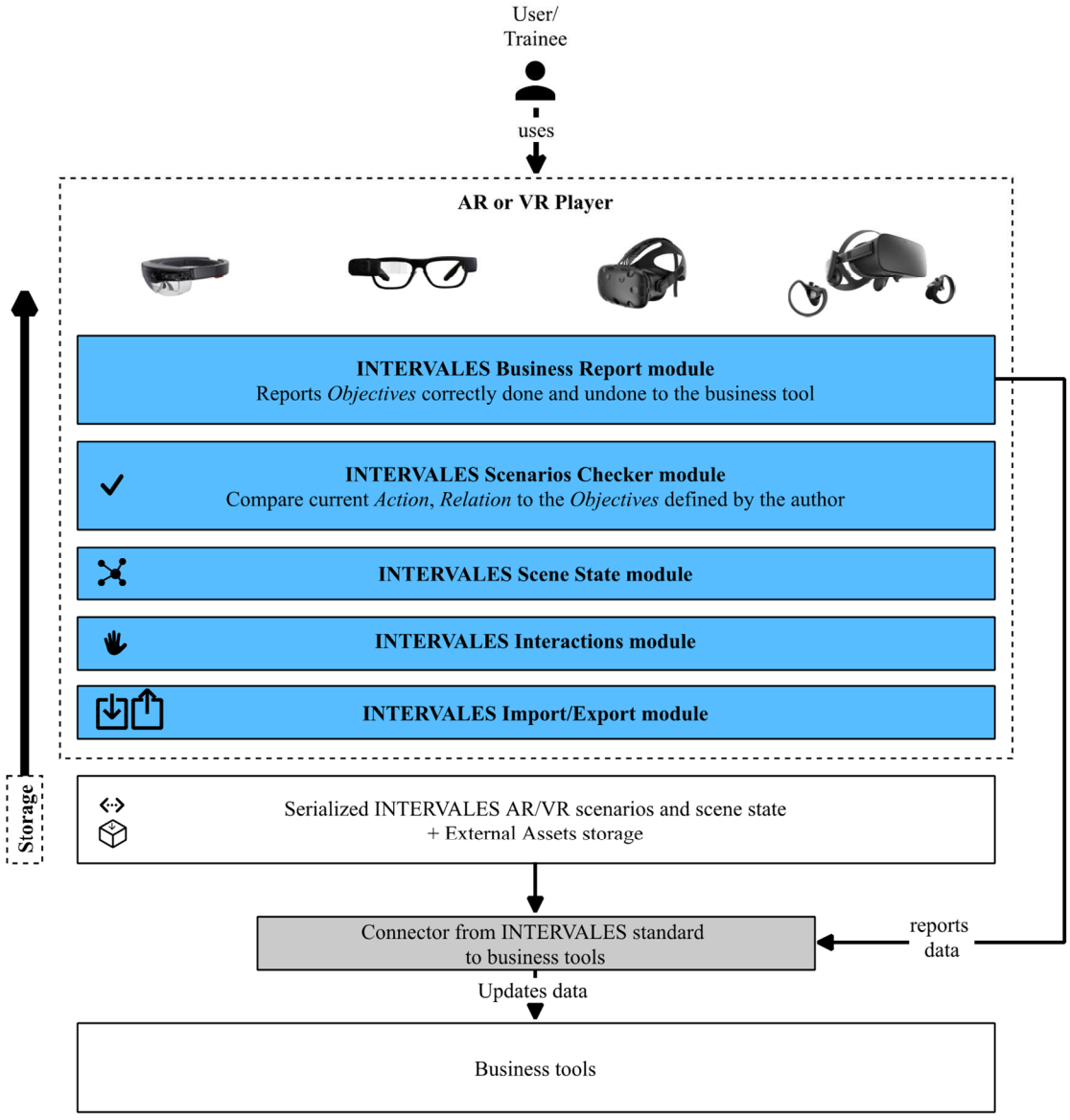

Figure 4: End-user functions of the INTERVALES architecture

Lastly, the **INTERVALES Business Report module** can report if each step is done correctly by the user to the business tool; this allows the business tool to keep track of what is occurring in the business resources, such as maintenance on a machine or training done by an operator for a specific machine.

The described architecture demonstrates that INTERVALES has the capacity to describe AR or VR scenarios in a formal language and be used in AR or VR applications. Moreover, the formal model enables other tools to describe or update scenarios. The next section further discusses the different concepts used in the modules, including Entity, Features, and Relation.

3.2. INTERVALES Model Based on Features-Relations

First, it is mandatory to describe a virtual scene with generic concepts that can be specialized for job-specific needs. That is why the INTERVALES model is UML-based with inheritance. The model defines the virtual world as a composition of two concepts: **Entities** and **Features** (see Figure 5). In the same way, each action done on Entities and Features must be tracked. It is thus possible to define **Actions** (performed on one Entity or Feature) or **Relations** (performed between two Features) and create interactions. Each Entity or Feature in the virtual or augmented world inherits from the **ObjectIdentityTransform** class to define the 3D object pose in the scene. This latter class inherits from the **ObjectIdentity** class, which allows it to quickly identify an object in the scene using a unique ID and path.

To illustrate these detailed concepts, an industrial assembly on manual workstations will be taken. More precisely, an assembly operation, including screws and insert nuts of different sizes; M6, M7, and M8 corresponding to the size will be used. As an **Entity** is an existing object in the virtual environment, it has a representation, referenced by ModelName. For example, it can be a screw, a nut, or a tool 3D model.

3.2.1. Entity and Feature

In order to make Entities interactive, a list of associated **Features** is defined (see Figure *5*: FeatureList). This allows for defining interactions available to the user and feasible relations between Features using compatibility rules. Each Feature is located on the entity in the 3D space with three axes (x, y, z). Moreover two property lists let define with which other Features a Feature can match. To do so, two property lists have been integrated: *OwnPropertyList*, which includes Features specifications, and *CompatibilityPropertyList*. The latter property list contains the required properties that another feature must have to be compatible and thus associable within a relation. Because Features own a 3D reference, no confusion of orientation is possible when they match. Indeed, two features are matched in an oriented way, with the y-axis in opposite direction.

As an example, if a feature *"Screw_M8"* has an *OwnPropertyList = ["M8_Male"]* and a *CompatibilityPropertyList = ["M8_Female"]*, this feature can only be associated with another Feature which has, for instance, *OwnPropertyList = ["M8_Female"]* and *CompatibilityPropertyList = ["M8_Male"]*.

It is important to note that when a Relation is made between two Features, the Entity containing each Feature will move to represent the Relation. Moreover, as an Entity can have several Features (e.g., an Entity "Screw" can have a Feature *"Screw_M8"* as well as a Feature *"Grabbable"*), it is possible to describe complex Entities in an assembly context which could have multiple of the same features among others. Additionally, **Features** can be extended as AR Features (see Figure *5*: FeatureARTracking), allowing an Entity to be linked, for example, to an AR tracking configuration to represent an assembly procedure.

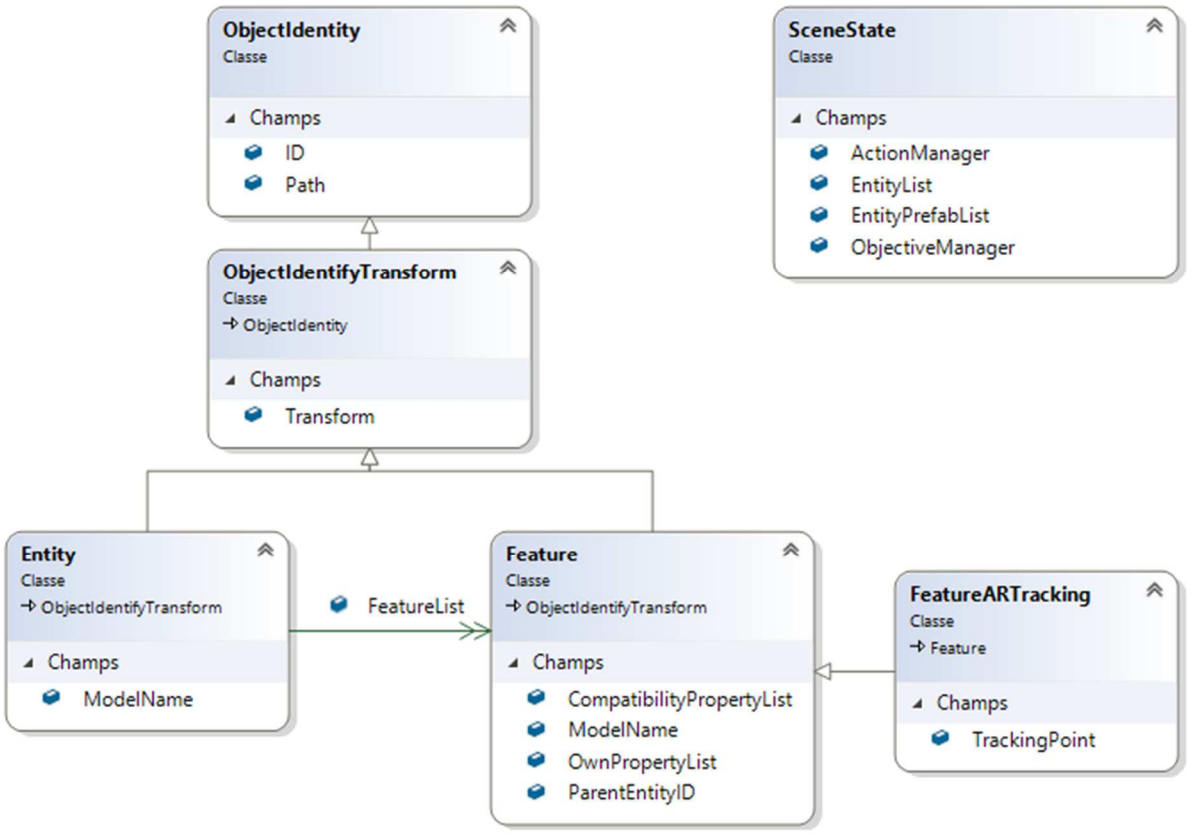

Figure 5: Class diagram of the Entity-Feature's engine

Moreover, since authors want to ensure the possibility of user mistakes during VR or AR training sessions, more complex compatibility cases appear. For instance, a scenario plans to associate M8 insert nut with M8 screw, but the authors want to enable the fact that it is physically possible to put in an M7 or M6 screw, even if it is a mistake. Therefore, the associations on each of them have to be done by defining compatibility rules independently.

As compatibility rules become more verbose the more there is to define, the exclusive compatibility "OR" in addition to "AND" has been included, allowing compatibility between the types XOR, NOT, NAND, NOR, and XNOR, taking the principle of the logical gates. Moreover, by allowing the use of Regular Expression (RegEx) as a compatibility rule, the model will facilitate a faster authoring process and optimization for the complex compatibility cases. For example, to enable the screw interaction between an M8 screw and an M8 insert nut while regarding a specific assembly, it is necessary to specify that only flat and round heads are concerned. Using the basic *CompatibilityPropertyList* of M8 insert nut, two compatibility lists have to be defined with the head type specified: *CompatibilityPropertyList = ["M8_SCREW_FLATHEAD", "M8_SCREW_ROUNDHEAD"]*. As with both flat and cross head screw compatibility list = *["M8_NUT"]*, two relations have to be defined (see Figure *6*: a).

Then, specific characters are supported: "&" = AND; " | " = OR ; " ^ " = XOR. Using a complex compatibility list, M8 screw and M8 nut properties can be merged, with the compatibility list *["M8&SCREW&(FLATHEAD|CROSSHEAD)"]* (see Figure *6*: b). Another way to enable mistakes would be not to specify some properties (see Figure *6*: c).

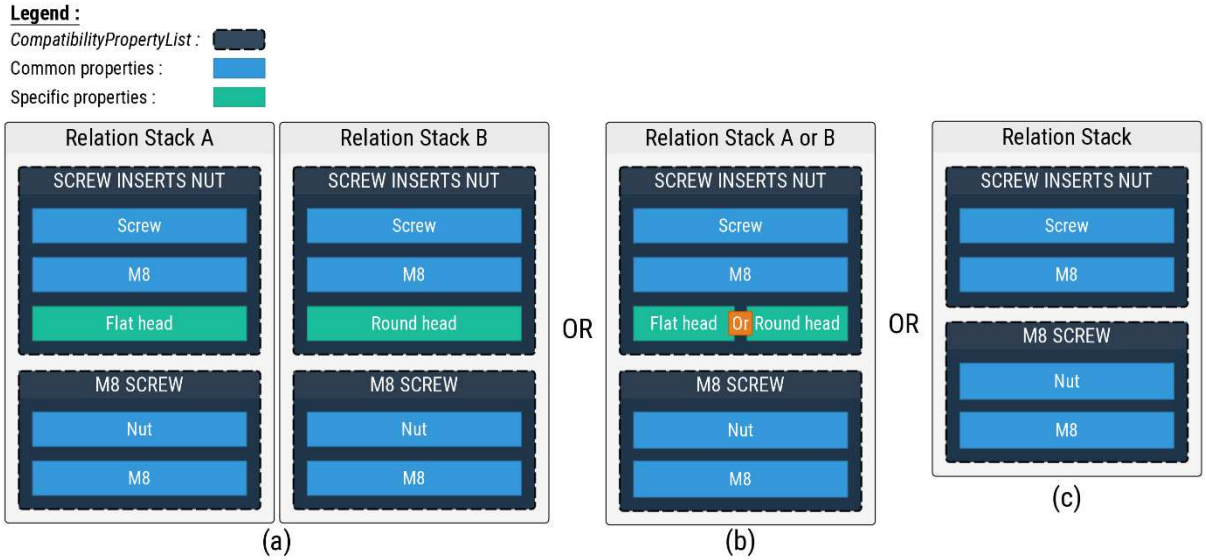

Figure 6: (a) Declaration of two relations without complex compatibility rules. (b) Declaration of only one relation using complex compatibility rules. (c) Declaration of only one relation but without specified properties.

In a use case where all the entities in the virtual world have a definite relation with a specific entity except one, it would be necessary to recreate each compatibility rule independently. It is therefore worth considering including an XOR "anything except…" compatibility as, for instance, the *CompatibilityPropertyList = ["^FLATHEAD"]*.

As a result, IT developers and experts/trainers can describe features in a more realistic way that allows interaction, even if there are theoretically wrong combinations (e.g., putting an M6 screw in a larger M8 nut), while also managing which errors are possible or not.

3.2.2. Action and Relation

Following the above descriptions of Entity-Feature, this section focuses on **Actions** on them and **Relations** between them. The aim of this part of the UML model is to track each action performed during the VR session. It produces a complete list of actions that is used during the scenario checking. Actions (see Figure *7*: b) are defined by an *ActionType* and an *EntityFeatureReference*. This action could be on a single Entity, such as a property modification or spawning an Entity, or it can result in a Relation (see Figure *7*: c) with another Feature. *EntityFeatureReferences* correspond to the required Entities and Features involved in an Action/Relation. Inheritance is used in order to specialize Relation and add custom information about it. For example, the **RelationSlider** (see Figure *7*: d) contains the distance from the start point of the slider. The **ActionManager** (see Figure *7*: a) enables the current state of the system with a list of currently active Actions and Relations.

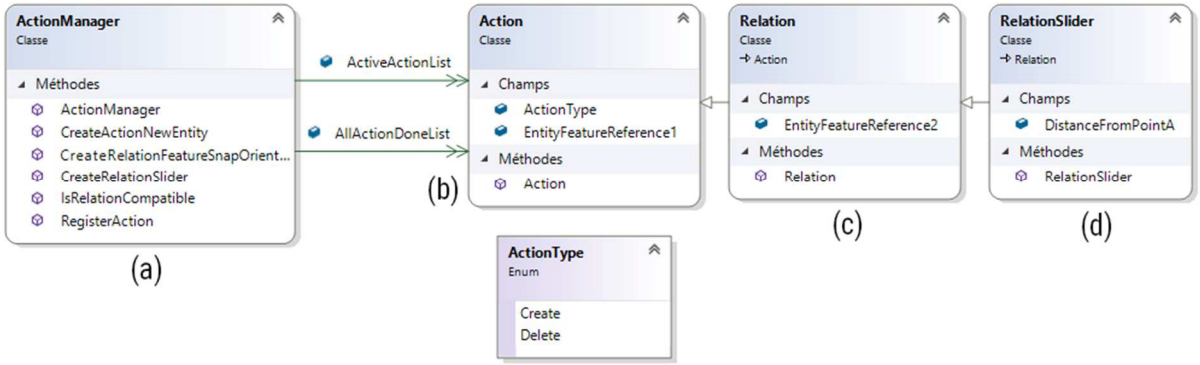

Figure 7: Class diagram of the relation's engine

Based on these four concepts (Entity-Feature-Action-Relation), the INTERVALES interaction module manages interactions made by the user. For instance, when a user grabs two Entities and brings two FeatureSlider closer one to each other, the user can see in the virtual environment a ghost piece showing where the first Entity would be placed if the second Entity was released and the RelationSlider made. Figure *13* The use case section (section 3.4) of this paper will further illustrate this interaction.

3.2.3. Scenario, Objective, and Task

With this knowledge of the environment and the actions and relations state between entities, **Scenarios** can be more easily authored by defining expected changes to the environment state. Indeed, a scenario is a list of Action/Relation with specific parameters that user must perform to reach the objective. In UML perspective, a scenario is divided into a series of **Objectives** to be reached (see Figure 8: Objective). An objective can be a unique **Task** (see Figure 8: UnitTask) to perform or a grouping of Tasks (see Figure 8: GroupTask) to be done in any order. A Task is in itself an Action that must be achieved by an *Agent* (real or virtual). Each Action (or Relation) has a corresponding Task. For example, the RelationSlider corresponds to the TaskSlider task. By specifying the tolerance threshold of the expected Action, it is thus possible to define how this task can be considered as correctly completed. For the example of the RelationSlider, by defining the *DistanceFromPointA = 25* mm and the *ValidationRules = [“DistanceFromPointA.Tolerance=±1”]*, the TaskSlider is considered correctly performed by the INTERVALES Scenarios Checker module if the parts are slid between [24mm, 26mm].

Finally, thanks to these three concepts (Scenario-Objective-Task), this layout of objectives and tasks can be used to manage procedural scenarios or more complex scenarios with tasks that could be done in different orders. For example, a procedural scenario for an AR maintenance application with actions to be done in a specific order can be set, or a VR assembly process containing several Tasks to execute without a specific order.

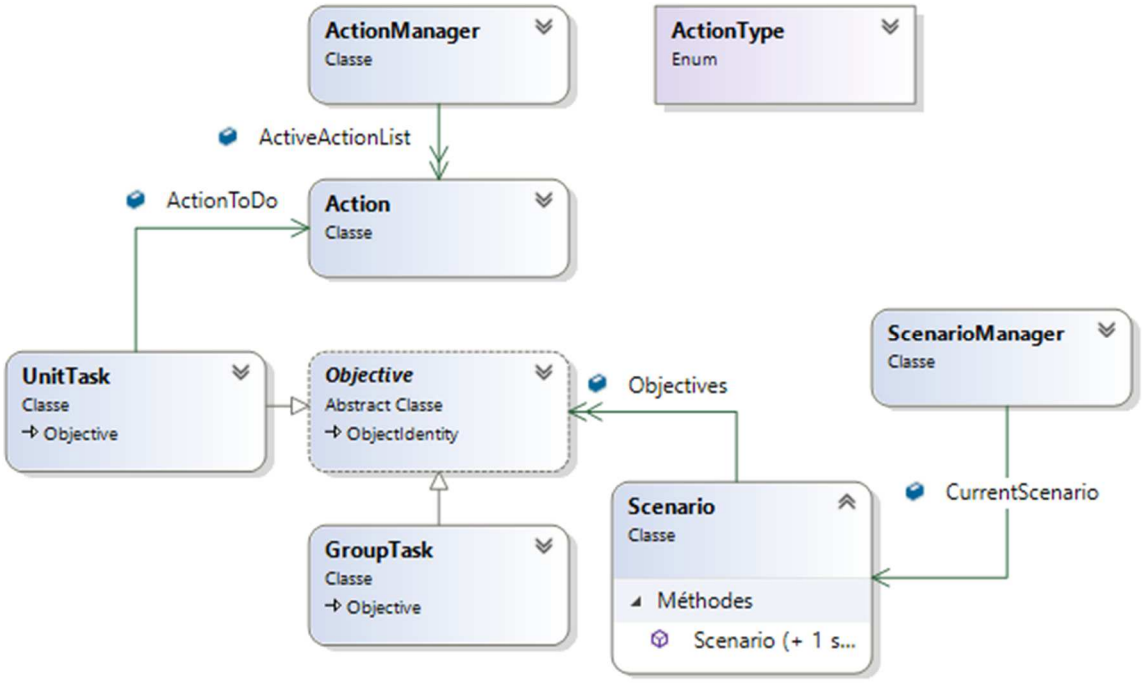

Figure 8: Task and Objective Orchestration Model

3.3. INTERVALES: The workflow Usage

The previous sections of this paper have defined the technical architecture and data model of the INTERVALES framework. To better understand the role of each stakeholder, this section will describe the usage workflow. As shown in Figure *9*, the process starts with the domain expert. This expert is the person mastering the business knowledge and process. From there, the expert has two options to work with regarding the INTERVALES framework. On one hand, the expert can choose to work with IT to produce the AR/VR applications and scenarios (see Figure *9*: in purple). In this case, the expert's role is to prepare application requirements while IT must understand the expert's job and develop the applications. On the other hand, the expert may choose to work autonomously (see Figure *9*: in blue). In both cases, experts or IT personnel must first prepare the 3D model with the Entity/Feature concepts, then define scenarios.

Once done, the application is validated and used in production (see Figure *9*: in green). As the application is based on Entity/Feature, Action/Relation, and Objectives, scenarios are defined in a UML-based language, they can be adapted to fit the industrial process evolution. For example, in Figure *9*, the AR/VR application user (or operator) suggests a modification; the modification is then taken into account by the expert, who is changing the defined scenario to take into account the operator's feedback.

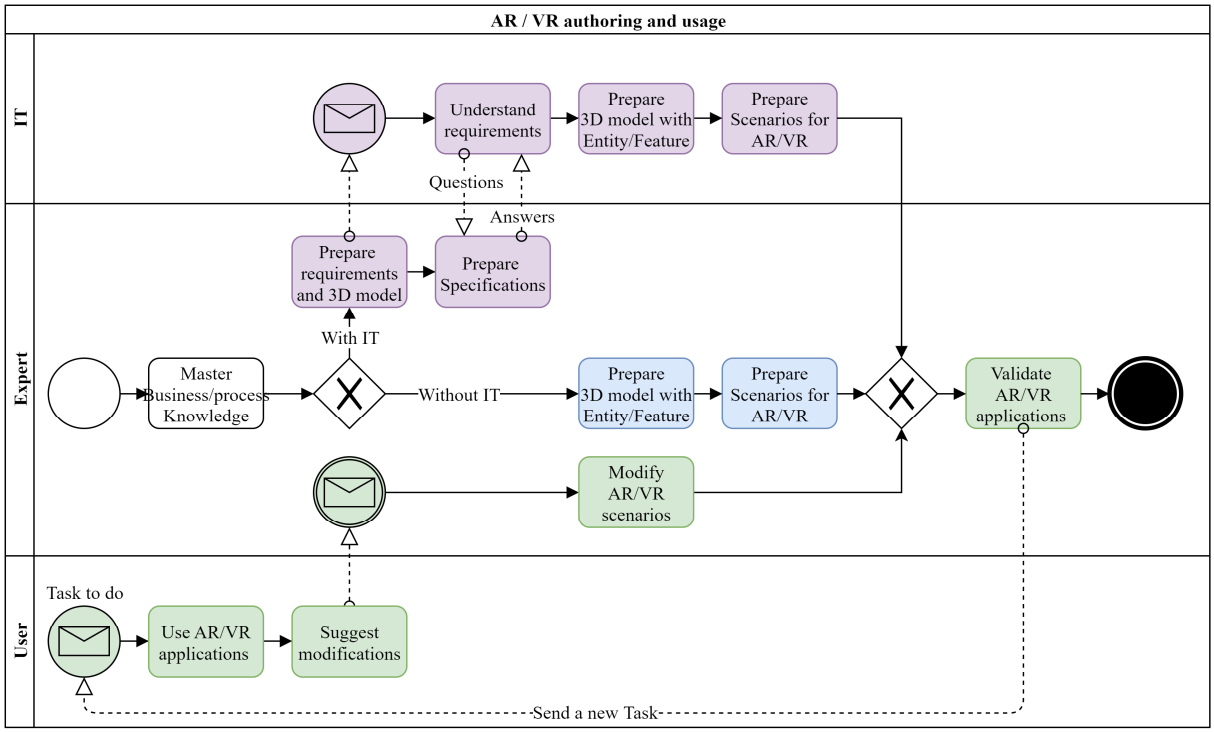

Figure 9: Usage workflow of the INTERVALES framework in BPMN

3.4. Industry 4.0 Case Study

3.4.1. Case study description and INTERVALES implementation

The use case is composed of manual and cobotic workstations for an assembly process (Figure *10*). The physical industrial workshop is composed of six workstations for the assembly and quality control of products (Figure *10*: a). Operations on one workstation are assisted by an UR10 cobotic arm for part manipulation and an augmented reality system for instruction assistance (Figure *10*: b). A digital twin integrating 3D CAD models, cobotic arm behavioral model, assembly instruction, and scenario has been developed. Based on this digital twin and on the INTERVALES framework, an interactive and collaborative virtual environment and an augmented reality scenario have been established. An

example of the VR environment of the industrial workshop is presented in Figure *10*: c, and a CAD view of the cobotic workstation in Figure *10*: d. During the design and digital prototype review phase, a VR environment is used for the design and ergonomic and safety assessments of the manual and cobotic workstations. Collaborative VR is also used for the workshop layout design. During the production phase, VR is used for training purposes regarding the assembly procedures on manual and cobotic workstations. The authored virtual environment is also used to train lean manufacturing principles [15]. An AR tool is used for guiding the operator through the assembly process. A digital twin associated with the data representing the process has been defined once according to the INTERVALES framework, and is used in both applications: the VR and the AR. As the data are the same for both environments, the next section will mainly describe how scenarios are built into the virtual one.

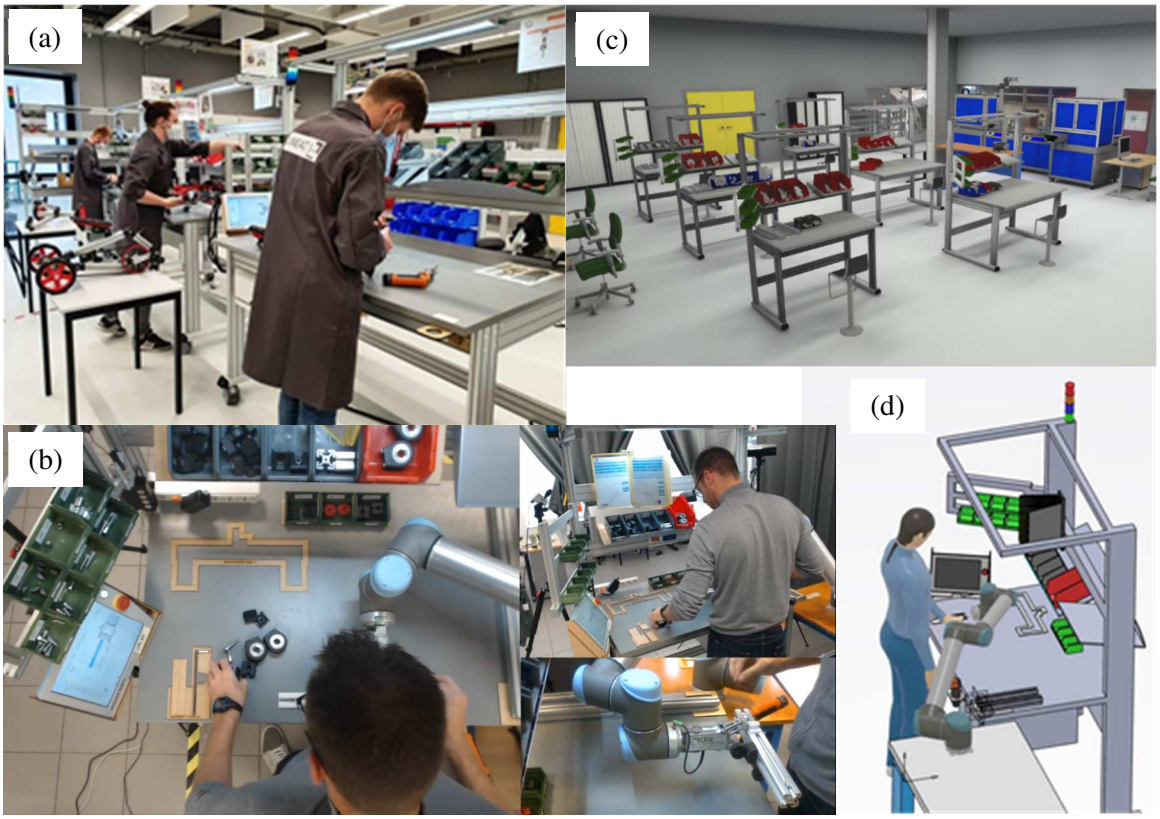

Figure 10: (a) Industrial workshop. (b) Assembly workstation assisted by a UR10 cobotic arm. (c) VR view of the industrial workshop. (d) CAD view of the cobotic workstation. All the showed pictures have been taken at LINEACT CESI's laboratory.

From the CAD model, a library of 150 entities—representing parts for the product assembly, parts of the workstation, tools, and cobot—has been created. With their features and relationships defined (Figure *11*), it is possible to freely assemble each of these parts. To understand how to format existing knowledge into an INTERVALES Scene, several steps have to be followed. In our case study, the assembly process is first analyzed; each required part and the different constraints requiring attention must be defined. We will focus on the 360 mm profile, labeled P360 in Figure *11*, as an example. For this part, other parts such as FIXA1 can be snapped on each side, and fixture keys can be slid in each of the four slits. Thus, two types of features, "FeatureSnap" and "FeatureSlider," with their properties and compatibility list, have been set (see Figure *11*: c). For instance, for the slider, particular attention must be paid to the distance from the starting point at which the fixture key has been slid. Thereby, this property is included in the feature slider. To support augmented reality, a FeatureARModelTracking is set so as to recognize the P360 with its 3D model. In the 3D engine,

Unity[2], the 3D models provided are imported, and the features previously defined in the model are positioned at the right place.

From this point, a scenario is defined to follow the instruction sheet visible in Figure 11: A. Three Objectives (each one containing a Task) are defined. The first one is TaskSlider between P360 and LARD with a *DistanceFromPointA="160mm"* and a *DistanceFromPointA.Tolerance=±2*. The second Task is a TaskSnap between FIXA1 and LARD with a *SnapAngle=0* and a *SnapAngle.Tolerance=0.5*. Finally, the third Task is a TaskScrew between B820 and LARD with no added parameter. From the authoring point of view, this process results in importing the part model (Figure 11: B), placing the features previously described on it (Figure 11: C) and then, configure them using the user interface (Figure 11: D). This will be converted into INTERVALES data that can be saved or reloaded later on (Figure 11: E).

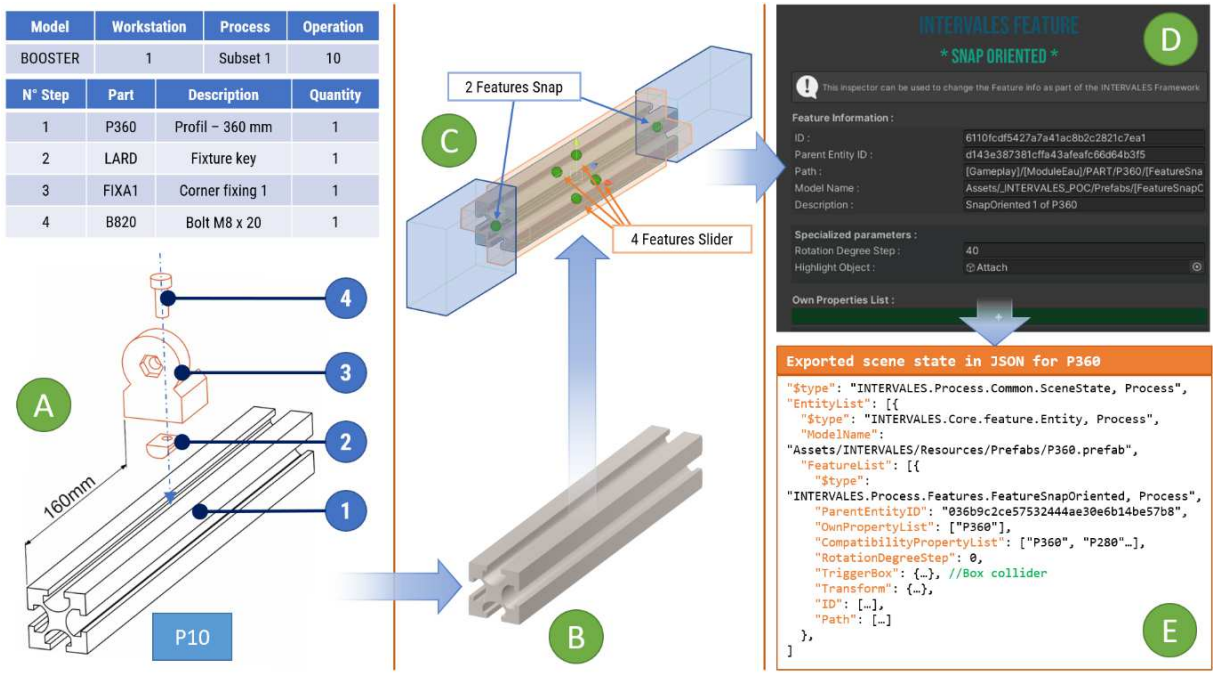

Figure 11: P360's Features placement from the instructions sheet to INTERVALES data. (a) Instructions sheet provided. (b) P360's 3D models. (c) Features placement in Unity3D. (d) SnapOriented Feature's configuration inspector in Unity3D. (e) INTERVALES Scene state in JSON.

Finally, when the VR application is launched, interactions are managed by the INTERVALES interaction module, which allows users to freely interact with the Entities. While interacting, the ScenarioManager checks that the Objectives are reached. For example, Figure 12 presents the expected first steps of the assembly. The user is expected to take a P360 part (Figure 12: 1) and to perform a SlideRelation between the P360 and the LARD (Figure 12: 2) at an expected 160mm of PointA with a tolerance of 5 mm (Figure 12: 3).

As far as the AR is concerned, tasks defined are used as a guidance system. For example, when the operator who is performing the assembly reaches the second step, the AR system displays a virtual LARD overlay on the real P360 to indicate where it must be placed.

[2] https://unity.com/

Figure 12: Scenario example as INTERVALES data in JSON

To author this presented scenario in a more understandable way, a user interface has been developed (Figure 13). It consists of a node graph user interface using nodes and group nodes, respectively, UnitTask and GroupTask. Users have access to all available actions such as, for this assembly use case, “Pick component”, “RelationSlider” or “RelationSnapOriented” and can configure them accordingly. For example, on a “Pick component” action, the user can select the entity type using a drop-down list and name the part. Then a list displays the currently used Entities in the scenario (Figure 13: A). All tasks in the GUI display a preview of the concerned Entity (Figure 13: B) or Feature’s location (Figure 13: C). To add a new task, the user can either right-click or click from the previous task “out” and then search for it using the search bar (Figure 13: D).

As a result, the IT or job expert can edit an INTERVALES scenario through the user interface by defining tasks and setting its parameters. As an example, the assembly instruction described in Figure 11: A are converted into a scenario, as shown in Figure 13, take a P360 part and a LARD part (Figure

13: 1), perform a SlideRelation between the P360 and the LARD (Figure 13: 2) at an expected 160mm of PointA with a tolerance of 5 mm (Figure 13: 3). Next assembly tasks are created in a same way.

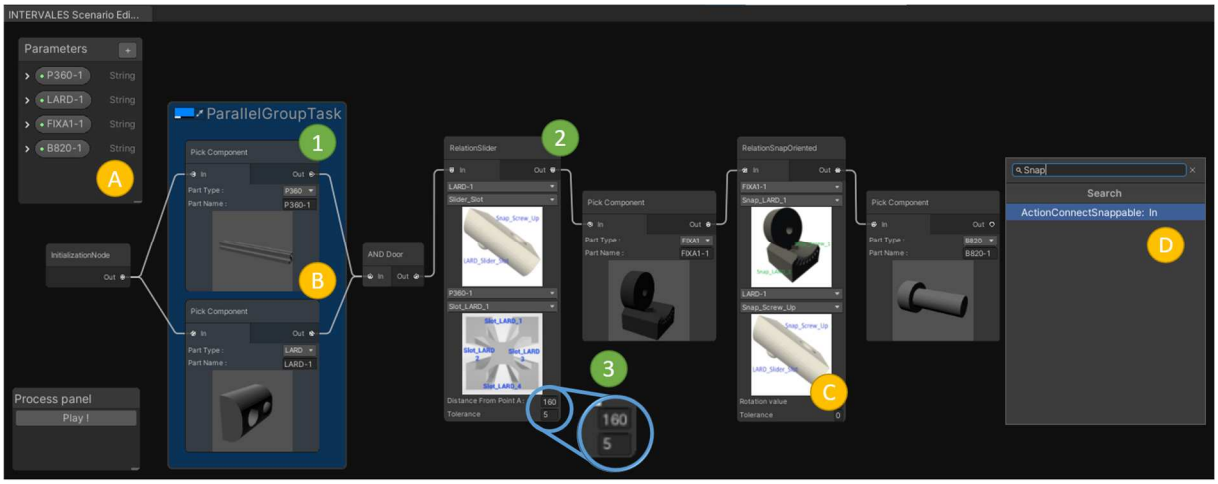

Figure 13: INTERVALES Scenario editor GUI

The scenario detailed above is illustrated through a case study about an assembly procedure assisted by a cobot (collaborative robot) in VR and AR (see Figure 14). In this use case, the operator processes assembly steps assisted by a UR10, a six degree of freedom (DoF) robotic arm. It holds a part while the operator acts on it. The scenario is composed of several phases. First, there is a preparation phase, realized with CAD/CAM tools on the workstation design with their configurations, including the robot selection as well as programming and the assembly instructions creation (see Figure 14: Design). Then, this knowledge prepared with CAD/CAM tools is exported into INTERVALES format, with Entity and Feature, and integrated into the framework. Next, and mainly as a result of the INTERVALES Interactions module, the authored virtual environment is used to validate the workstation design and ergonomic study by making an operator perform the assembly in VR scene (see Figure 14: Digital Prototype). Moreover, each operator is represented here by a yellow capsule to ensure that the operator and the cobot are not colliding during the process. Collaborative workshop layout design or assessment are also realized in the VR environment. Multiple users in VR can adjust and validate the layout of workstations, storage areas, and circulation areas. Once the digital prototype is validated, each assembly process is defined through the INTERVALES scenarios module. As a consequence, VR training scenarios are usable by beginner operators based on INTERVALES data exported from previous phases (see Figure 14: Production). Moreover, based on the same INTERVALES data and by using a camera on top of the manual workstations and a dedicated screen fixed on the workstation, AR instructions are provided to the operator in real-time during assembly (see Figure 14: Production in AR). AR instructions are visible from a remote display next to the operator, while the camera is filming the workstation from the top of the operator. AR applications are rendered with Unity associated to Vuforia AR framework. In this case, we choose to use AR with remote screen display to avoid equipping and encumber the operator with devices like Smart Glasses such as Vuzix M300 or Microsoft HoloLens. Tablet and mobile options were also discarded, since the operator must remain hands-free. However, as instructions are based on the INTERVALES model, those new devices could be integrated by developing the specific application able to read it. Thus, INTERVALES covers the workstation and ergonomic design, the assembly training with Human-Robot Collaboration (HRC), and the augmented reality assembly assistance. The next step for this application would be to offer real-time digital twin features—for instance, by transposing cobot movement between the virtual and physical systems.

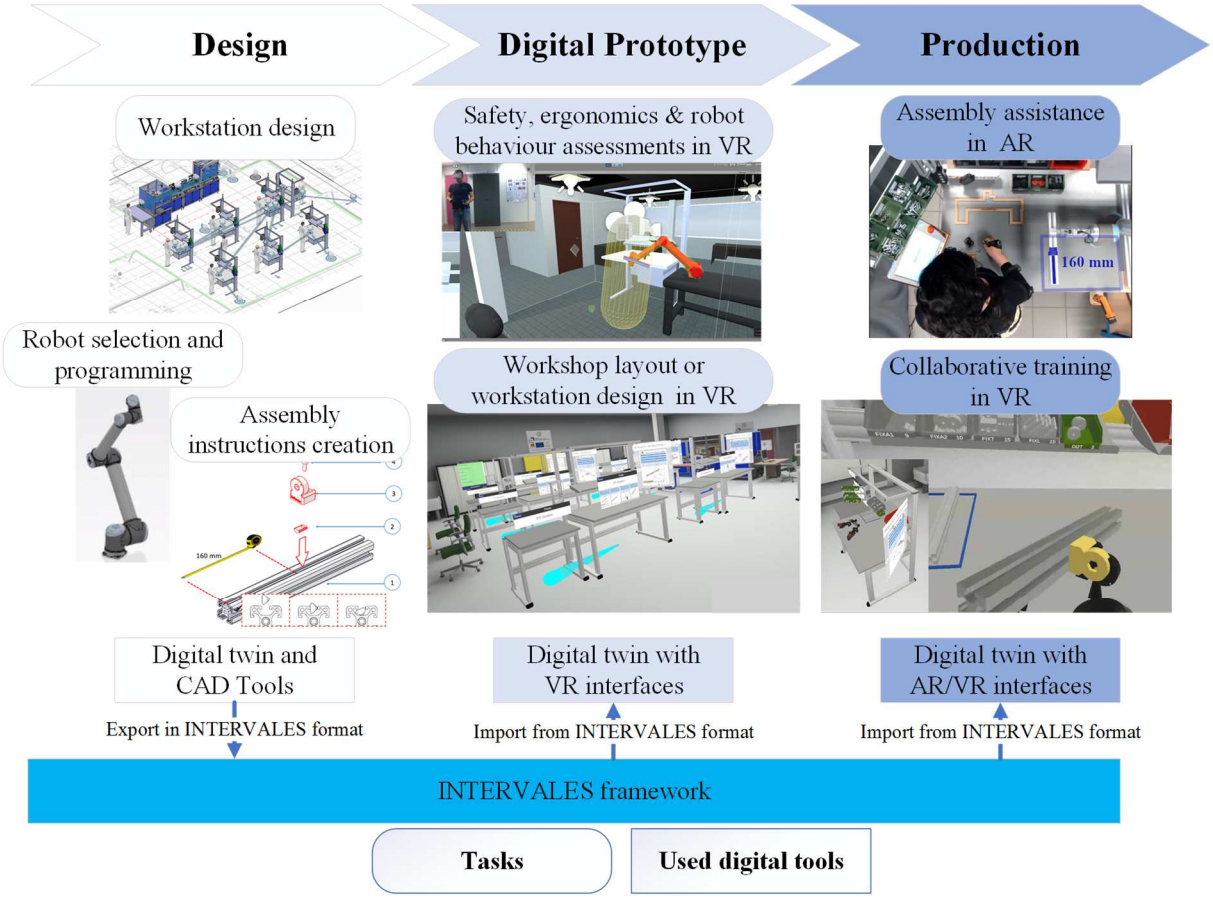

Figure 14: Assembly procedure assisted by a cobot in VR and AR with Workstation design, digital prototype, and production of a training to assembly based on INTERVALES and AR support based on INTERVALES

Thanks to the INTERVALES Framework presented in this section, we were able to author a virtual/augmented environment, the feasible interactions, and a scenario based on selected instruction sheets. The presented industrial use case is a collaborative tool in which actions can be performed by any users or autonomous agents, in this case the robotic arm. Furthermore, because the environment is based on INTERVALES data, it can be easily saved and restored during a VR session. Thus, it is possible for the trainer to interrupt the assembly procedure during a session and reload it where the trainee stopped.

To conclude, the following table (Table *2*) shows the characteristics seen in the state of the art and applies them to the different use cases in order to show the concepts taken into account by INTERVALES.

Table 2: INTERVALES characteristics applied to the presented use cases

Use cases	Technology (AR/VR)	Procedural scenario	Scenario w/ objectives	Opened scenario	Collaborative?	Autonomous agent?	Causality?
Workshop layout design	VR			X	X		
Workstation design and ergonomics assessment; Safety and robot behavior assessment				X	X	X	
Assembly training		X	X		X	X	X
Assembly assistance	AR	X				X	

3.4.2. Experimentation Results on INTERVALES editor evaluation

To collect user feedbacks and evaluate if the INTERVALES framework and its user interface for scenario authoring can be used by IT experts as well as job experts without development skills, we made an experiment with 11 respondents (5 developers and 6 job experts without development skills) from 25 to 53 years old. To do so, they had to author the scenario corresponding to a series of operations, shown in Figure 15, using the INTERVALES interface.

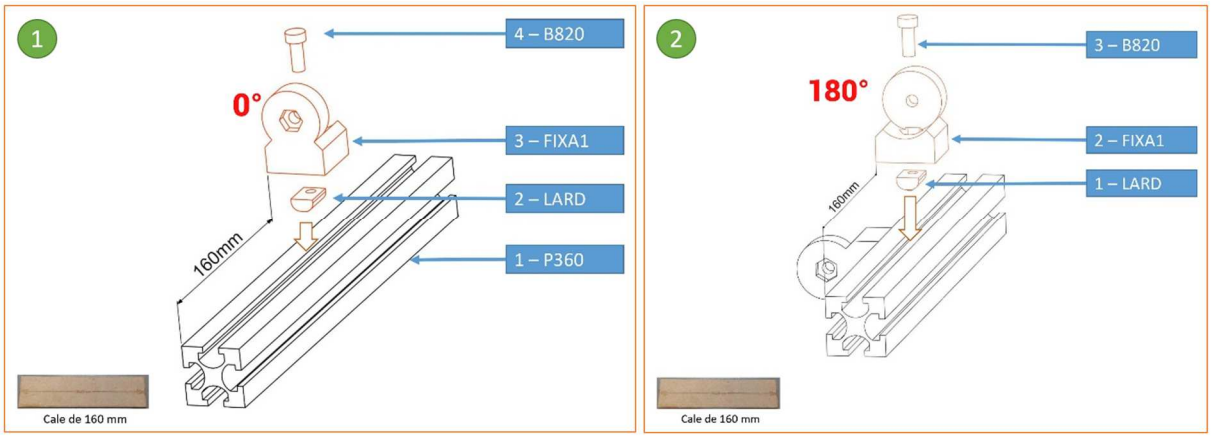

Figure 15: Series of operations used for the experience

The experiment takes place as follows: after an introduction video explaining the context and how to use the interface, each person had a free phase to practice with the INTERVALES editor, followed by a phase in autonomy to carry out the experience and their completion times were recorded. To finish, they had to answer a questionnaire composed of a SUS (System-Usability-Scale [57]) and a UEQ (User Experience Questionnaire[3]). We chose these questionnaires since the System-Usability-Scale provides a reliable tool to measure the usability with a score from 0 to 100 and the UEQ, to measure the user experience using 26 pairs of opposite adjectives (ex: complicated/easy).

Figure 16 presents the SUS score (on left) and the completion time (on right) according to the participant profile (developer DEV and job expert NDEV). To test if the results are statically different, we performed a T-Test. First, homogeneity of variances is tested with the Levene's test. The results return a p-value of $0.395 > 0.05$ for the SUS scores and a p-value of $0.493 > 0.05$ for the completion times. Then, data normality distribution is tested for both groups using the Shapiro tests. Concerning the satisfaction score, the results are a p-value of $0.317 > 0.05$ for developers and a p-value of $0.329 > 0.05$. For the time, it returns p-value of $0.124 > 0.005$ and p-value of $0.117 > 0.05$. Because both homogeneity and normality tests are validated, we could run the T-test and the results are a p-value of $0.079 > 0.05$ for the SUS score and a p-value of $0.893 > 0.05$ concerning the completion times. Since the T-test gives a p-value > 0.05, we can conclude that completion time and SUS score do not significantly depend on the participant's category: IT experts or job experts (non-developers).

That is why we can conclude that the INTERVALES editor allows IT experts or job experts to create a scenario in the same duration: **520 seconds** for the developers and **525 seconds** for the other group. Indeed, the mean duration to carry out the two operations series are similar (Figure 16). The same conclusion can be made concerning the SUS score. Even if there are statically the same, further comments can be made. The mean for IT experts and non-developers, are respectively, **70.5** and **83.8**. This can be explained by the fact that developers are more critical since they commonly used similar interfaces while job experts have shown good interest in the use of this interface and the possibility to

[3] https://www.ueq-online.org/

edit scenario with visual blocks. And indeed, the UEQ data analyse confirms that IT experts show less attractiveness than job experts with, respectively, a mean (from -3 to 3) of *0.7* and *1.89*.

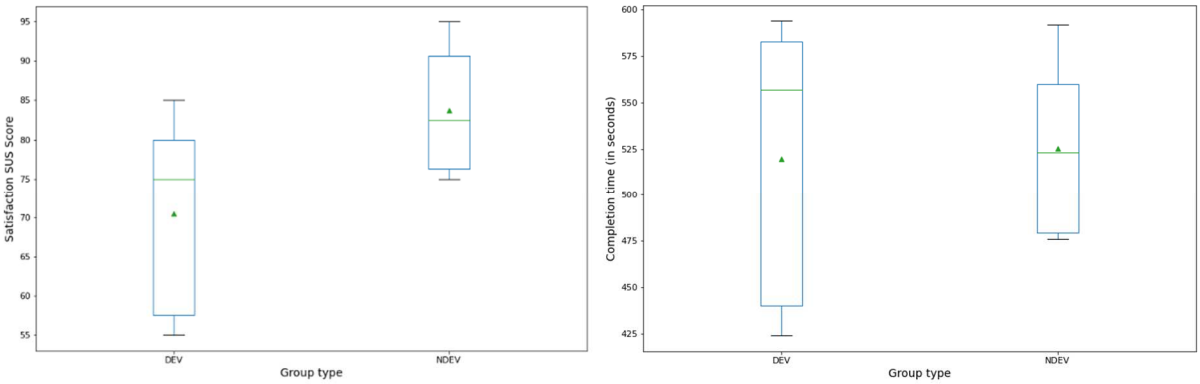

Figure *16*: *(On the left) Box plot of SUS score in percentage depending on the group's type. (*On *the right) Box plot of completion times depending on the group's type.*

4. Conclusion

This paper focused on modeling augmented and virtual reality interactions and scenarios to easily author training situations or procedural guidance in an industrial context. After reviewing existing models and approaches in the literature, this paper identified the required characteristics and proposed the novel INTERVALES framework and associated data model.

First, the different modules involved in the framework have been presented with the goal of allowing AR/VR scenario authoring either by IT developers or job experts. This proposed framework is composed of five different modules: the Entity-Feature module, to define each asset characteristic; the Interaction module, to manage all interactions made by the user; the Scene state module, to manage all Action/Relation done; the Scenario Authoring module, to edit scenarios; and the Import/Export module, to save and load scene state. Thus, our model makes it possible to author the environment and the interaction within it in order to define tasks and objectives. Moreover, this paper has described INTERVALES classes and how they architecture together to edit a complex assembly operations scenario that allows room for user mistakes. Finally, we have shown the usefulness of such a model by using it for the implementation of two industrial case studies in virtual and augmented reality with a common data content in the context of Industry 4.0. The experiment results show that there is a craze towards the user interface made as well as it is possible for an IT expert or job expert to author a scenario as fast as the other with a same time of practice.

Our future research will focus on procedure "authoring-by-doing," which will allow an expert to generate tasks by performing the assembly into the VR scene. This will allow users to easily create and update procedures by performing the assembly. Also, following the experiment carried out, we gathered feedbacks and ideas in order to improve the usability of the user interface and INTERVALES framework. For example, an idea was to display an animation preview of the action to improve understandability.

Acknowledgments

This work has been done as part of a PhD project in collaboration between the LINEACT laboratory (Laboratory of Digital Innovation for Enterprises and Apprenticeships in the Service of Competitiveness of the Territories) and the company OREKA Ingénierie.

This research did not receive any specific grant funding from agencies in the public, commercial, or not-for-profit sectors.

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.

References:

[1] E. Rauch, C. Linder, P. Dallasega, Anthropocentric perspective of production before and within Industry 4.0, Comput. Ind. Eng. 139 (2020) 105644. https://doi.org/10.1016/j.cie.2019.01.018.

[2] R. Raisamo, I. Rakkolainen, P. Majaranta, K. Salminen, J. Rantala, A. Farooq, Human augmentation: Past, present and future, Int. J. Hum. Comput. Stud. 131 (2019) 131–143. https://doi.org/10.1016/j.ijhcs.2019.05.008.

[3] L. Damiani, M. Demartini, G. Guizzi, R. Revetria, F. Tonelli, Augmented and virtual reality applications in industrial systems: A qualitative review towards the industry 4.0 era, IFAC-PapersOnLine. 51 (2018) 624–630. https://doi.org/10.1016/j.ifacol.2018.08.388.

[4] J. Geng, X. Song, Y. Pan, J. Tang, Y. Liu, D. Zhao, Y. Ma, A systematic design method of adaptive augmented reality work instruction for complex industrial operations, Comput. Ind. 119 (2020) 103229. https://doi.org/10.1016/j.compind.2020.103229.

[5] V. Havard, B. Jeanne, M. Lacomblez, D. Baudry, Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations, Prod. Manuf. Res. 7 (2019) 472–489. https://doi.org/10.1080/21693277.2019.1660283.

[6] P. Galambos, Á. Csapó, P. Zentay, I.M. Fülöp, T. Haidegger, P. Baranyi, I.J. Rudas, Design, programming and orchestration of heterogeneous manufacturing systems through VR-powered remote collaboration, Robot. Comput. Integr. Manuf. 33 (2015) 68–77. https://doi.org/10.1016/j.rcim.2014.08.012.

[7] L. Pérez, E. Diez, R. Usamentiaga, D.F. García, Industrial robot control and operator training using virtual reality interfaces, Comput. Ind. 109 (2019) 114–120. https://doi.org/10.1016/j.compind.2019.05.001.

[8] R. Stone, Virtual reality for interactive training: An industrial practitioner's viewpoint, Int. J. Hum. Comput. Stud. 55 (2001) 699–711. https://doi.org/10.1006/ijhc.2001.0497.

[9] L.P. Berg, J.M. Vance, Industry use of virtual reality in product design and manufacturing: a survey, Virtual Real. 21 (2017). https://doi.org/10.1007/s10055-016-0293-9.

[10] E. Matsas, G.C. Vosniakos, Design of a virtual reality training system for human–robot collaboration in manufacturing tasks, Int. J. Interact. Des. Manuf. 11 (2017) 139–153. https://doi.org/10.1007/s12008-015-0259-2.

[11] M. Funk, Augmented reality at the workplace: a context-aware assistive system using in-situ projection, PHD Thesis. (2016) 250. https://doi.org/10.18419/opus-8997.

[12] S. Henderson, S. Feiner, Exploring the benefits of augmented reality documentation for maintenance and repair, IEEE Trans. Vis. Comput. Graph. 17 (2011) 1355–1368. https://doi.org/10.1109/TVCG.2010.245.

[13] V. Havard, D. Baudry, X. Savatier, B. Jeanne, A. Louis, B. Mazari, Augmented industrial maintenance (AIM): A case study for evaluating and comparing with paper and video media supports, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9768 (2016) 302–320. https://doi.org/10.1007/978-3-319-40621-3_22.

[14] P. Vanneste, Y. Huang, J.Y. Park, F. Cornillie, B. Decloedt, W. Van den Noortgate, Cognitive support for assembly operations by means of augmented reality: an exploratory study, Int. J. Hum. Comput. Stud. (2020) 102480. https://doi.org/10.1016/j.ijhcs.2020.102480.

[15] A. Badets, V. Havard, K. Richard, D. Baudry, Using collaborative VR technology for Lean Manufacturing Training: a case study, in: Presented at the VRIC ConVRgence 2020: 22nd Virtual Reality International Conference, Laval Virtual, Laval, France, 2020.

[16] C. Buche, R. Querrec, P. de Loor, P. Chevaillier, Mascaret: Pedagogical multi-agents system for virtual environment for training., 2011. https://hal.archives-ouvertes.fr/hal-00610857 (accessed May 7, 2019).

[17] R. Bouville, V. Gouranton, T. Boggini, F. Nouviale, B. Arnaldi, #FIVE : High-level components for developing collaborative and interactive virtual environments, in: 2015 IEEE 8th Work. Softw. Eng. Archit. Realt. Interact. Syst. SEARIS 2015, 2017: pp. 33–40. https://doi.org/10.1109/SEARIS.2015.7854099.

[18] R. Querrec, P. Vallejo, C. Buche, MASCARET: creating virtual learning environments from system modelling, Eng. Real. Virtual Real. 2013. 8649 (2013) 864904. https://doi.org/10.1117/12.2008186.

[19] H. Martínez, S. Laukkanen, J. Mattila, A New Flexible Augmented Reality Platform for Development of Maintenance and Educational Applications, Int. J. Virtual Worlds Hum. Comput. Interact. (2014). https://doi.org/10.11159/vwhci.2014.003.

[20] V. Havard, D. Baudry, A. Louis, B. Mazari, Augmented reality maintenance demonstrator and associated modelling, 2015 IEEE Virtual Real. Conf. VR 2015 - Proc. (2015) 329–330. https://doi.org/10.1109/VR.2015.7223429.

[21] R. Hervás, J. Bravo, J. Fontecha, A context model based on ontological languages: A proposal for information visualization, J. Univers. Comput. Sci. 16 (2010) 1539–1555. https://doi.org/10.3217/jucs-016-12-1539.

[22] V. Lanquepin, D. Lourdeaux, C. Barot, K. Carpentier, M. Lhommet, K. Amokrane, HUMANS: A HUman models based artificial environments software platform, ACM Int. Conf. Proceeding Ser. (2013). https://doi.org/10.1145/2466816.2466826.

[23] V. Havard, B. Jeanne, X. Savatier, D. Baudry, Inoovas - Industrial Ontology for Operation in Virtual and Augmented Scene: the architecture, in: Int. Conf. Control Decis. Inf. Technol., 2017. https://doi.org/10.1109/CoDIT.2017.8102608.

[24] E. Sanfilippo, W. Terkaj, Editorial: Formal Ontologies meet Industry, Procedia Manuf. 28 (2019) 174–176. https://doi.org/10.1016/j.promfg.2018.12.028.

[25] V. Havard, A. Trigunayat, K. Richard, D. Baudry, Collaborative virtual reality decision tool for planning industrial shop floor layouts, in: Procedia CIRP, Elsevier B.V., 2019: pp. 1295–1300. https://doi.org/10.1016/j.procir.2019.04.016.

[26] G. Claude, V. Gouranton, R. Bouville Berthelot, B. Arnaldi, G. Claude, V. Gouranton, R. Bouville Berthelot, B. Arnaldi, a Sensor Effector Based Scenarios Model for Driving Collaborative Virtual Environment, 2014. https://hal.archives-ouvertes.fr/hal-01086237 (accessed July 2, 2020).

[27] S. Borsci, G. Lawson, S. Broome, Empirical evidence, evaluation criteria and challenges for the effectiveness of virtual and mixed reality tools for training operators of car service maintenance, Comput. Ind. 67 (2015) 17–26. https://doi.org/10.1016/j.compind.2014.12.002.

[28] Z. Merchant, E.T. Goetz, L. Cifuentes, W. Keeney-Kennicutt, T.J. Davis, Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis, (2014). https://doi.org/10.1016/j.compedu.2013.07.033.

[29] C.-H. Su, T.-W. Cheng, C.-H. Su, T.-W. Cheng, A Sustainability Innovation Experiential Learning Model for Virtual Reality Chemistry Laboratory: An Empirical Study with PLS-SEM and IPMA, Sustainability. 11 (2019) 1027. https://doi.org/10.3390/su11041027.

[30] Z. Feng, V.A. González, C. Mutch, R. Amor, A. Rahouti, A. Baghouz, N. Li, G. Cabrera-Guerrero, Towards a customizable immersive virtual reality serious game for earthquake emergency training, Adv. Eng. Informatics. 46 (2020) 101134. https://doi.org/10.1016/j.aei.2020.101134.

[31] Z. Wang, X. Bai, S. Zhang, M. Billinghurst, W. He, Y. Wang, D. Han, G. Chen, J. Li, The role of user-centered AR instruction in improving novice spatial cognition in a high-precision procedural task, Adv. Eng. Informatics. 47 (2021) 101250. https://doi.org/10.1016/j.aei.2021.101250.

[32] M. Nebeling, M. Speicher, The Trouble with Augmented Reality/Virtual Reality Authoring Tools, in: Adjun. Proc. - 2018 IEEE Int. Symp. Mix. Augment. Reality, ISMAR-Adjunct 2018, 2018: pp. 333–337. https://doi.org/10.1109/ISMAR-Adjunct.2018.00098.

[33] J. Casarin, N. Pacqueriaud, D. Bechmann, UMI3D : A Unity3D Toolbox to Support CSCW Systems Properties in Generic 3D User Interfaces, Proc. ACM Human-Computer Interact. 2 (2018) 1–20. https://doi.org/10.1145/3274298.

[34] J. Lacoche, T. Duval, B. Arnaldi, E. Maisel, J. Royan, Providing plasticity and redistribution for 3D user interfaces using the D3PART model, J. Multimodal User Interfaces. 11 (2017) 197–210. https://doi.org/10.1007/s12193-017-0239-x.

[35] T. Masood, J. Egger, Adopting augmented reality in the age of industrial digitalisation, Comput. Ind. 115 (2020) 103112. https://doi.org/10.1016/j.compind.2019.07.002.

[36] W. Friedrich, D. Jahn, L. Schmidt, ARVIKA-Augmented Reality for Development, Production and Service., in: ISMAR, 2002: pp. 3–4.

[37] J.Y. Lee, G. Rhee, Context-aware 3D visualization and collaboration services for ubiquitous cars using augmented reality, Int. J. Adv. Manuf. Technol. 37 (2008) 431–442.

[38] C.-J. Su, P.-T. Liu, Y.-C. Lin, others, Automatic generation of Augmented Reality enabled pedagogical system using Object-Oriented Analysis and Design in Process Modeling, Asia Pacific Ind. Eng. Manag. Syst. Conf. (2012).

[39] J. Zhu, S.K. Ong, A.Y.C. Nee, An authorable context-aware augmented reality system to assist the maintenance technicians, Int. J. Adv. Manuf. Technol. 66 (2013) 1699–1714.

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

[41] F. Lamarche, S. Donikian, Automatic Orchestration of Behaviours through the management of Resources and Priority Levels, 2002.

[42] M. Badawi, S. Donikian, Autonomous agents interacting with their virtual environment through synoptic objects, 2004. https://www.researchgate.net/publication/228919558 (accessed December 17, 2019).

[43] J.L. Lugrin, M. Cavazza, AI-based world behaviour for emergent narratives, in: Int. Conf. Adv. Comput. Entertain. Technol. 2006, 2006. https://doi.org/10.1145/1178823.1178853.

[44] N. Mollet, S. Gerbaud, B. Arnaldi, STORM: a Generic Interaction and Behavioral Model for 3D Objects and Humanoids in a Virtual Environment, IPT-EGVE 13th Eurographics Symp. Virtual Environ. (2007) 95–100. https://hal.inria.fr/inria-00475525.

[45] P. Chevaillier, T.H. Trinh, M. Barange, P. De Loor, F. Devillers, J. Soler, R. Querrec, Semantic modeling of virtual environments using MASCARET, in: 2012 5th Work. Softw. Eng. Archit. Realt. Interact. Syst. SEARIS 2012, 2012: pp. 1–8. https://doi.org/10.1109/SEARIS.2012.6231174.

[46] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein, OWL Web Ontology Language Reference, 2004. http://www.w3.org/TR/owl-ref/ (accessed December 19, 2019).

[47] C.A. Petri, Communication with automata, 1966.

[48] J.S. Willans, M.D. Harrison, Prototyping Pre-implementation Designs of Virtual Environment Behaviour, 2001.

[49] M. Kallmann, D. Thalmann, Direct 3D interaction with Smart Objects, in: ACM Symp. Virtual Real. Softw. Technol. Proceedings, VRST, ACM, 1999: pp. 124–130. https://doi.org/10.1145/323663.323683.

[50] B. Gajšek, S. Stradovnik, A. Hace, Sustainable Move towards Flexible, Robotic, Human-Involving Workplace, Sustainability. 12 (2020) 6590. https://doi.org/10.3390/su12166590.

[51] J.A. Erkoyuncu, I.F. del Amo, M. Dalle Mura, R. Roy, G. Dini, Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality, CIRP Ann. - Manuf. Technol. 66 (2017) 465–468. https://doi.org/10.1016/j.cirp.2017.04.006.

[52] OGC, ARML 2.0 SWG, (2015). http://www.opengeospatial.org/projects/groups/arml2.0swg.

[53] R. Hervás, J. Bravo, J. Fontecha, V. Villarreal, Achieving Adaptive Augmented Reality through Ontological Context-Awareness applied to AAL Scenarios., J. UCS. 19 (2013) 1334–1349. http://www.esi.uclm.es/www/Jesus.Fontecha/web/index.php/research-topics.

[54] F. Lamberti, F. Manuri, A. Sanna, G. Paravati, P. Pezzolla, P. Montuschi, Challenges, Opportunities and Future Trends of Emerging Techniques for Augmented Reality-based Maintenance, IEEE Trans. Emerg. Top. Comput. 2 (2014) 411–421. https://doi.org/10.1109/TETC.2014.2368833.

[55] ISO/IEC, Mixed and augmented reality (MAR) reference model, (2014) 69. https://goo.gl/ZNdqfJ.

[56] K. Ottogalli, D. Rosquete, A. Amundarain, I. Aguinaga, D. Borro, Flexible framework to model industry 4.0 processes for virtual simulators, Appl. Sci. 9 (2019) 4983. https://doi.org/10.3390/app9234983.

[57] J.R. Lewis, The System Usability Scale: Past, Present, and Future, Int. J. Hum. Comput. Interact. 34 (2018) 577–590. https://doi.org/10.1080/10447318.2018.1455307.