Towards Flexible Authoring and Personalization of Virtual Reality Applications for Training

KETOMA VIX KEMANJI, Heilbronn University, Germany and Université catholique de Louvain, Belgium
JEAN VANDERDONCKT, Université catholique de Louvain, Belgium
GERRIT MEIXNER, Heilbronn University, Germany

Fig. 1. Example of a Virtual Reality Application for Training.

Virtual reality applications offer the promise to immerse end users in a synthetic environment where several actions could be observed, simulated, and reproduced, before transferring them to reality, which makes them particularly appropriate for training. Yet, when the training requires complex handling of information, the tasks become cognitively intensive, and developing adequate applications becomes challenging. To address this challenge, we define a method for developing head-mounted-display-based virtual reality applications for modular training tasks, composed of a training model with parameters, a step-wise approach for supporting this development, and a software framework enacting the application of this approach. Authoring such applications is expected to become more flexible and provide personalization facilities. To evaluate the impact of this method, we define a case study concerning an application for training school teachers who deal with a variety of situations in a classroom for an experiment involving N=7 participants for a set of tasks. Pre-study and post-study acceptances reveal the impact of the software framework and a workload evaluation is conducted using the NASA TLX questionnaire.

CCS Concepts: • **Software and its engineering** → **Application specific development environments**; **Virtual worlds training simulations**; • **Applied computing** → *Interactive learning environments*; • **Human-centered computing** → *Virtual reality*; *User studies.*

189

Authors' addresses: Ketoma Vix Kemanji, ketoma-vix.kemanji@hs-heilbronn.de, Heilbronn University, UniTyLab, Max-Planck-Straße 39, 74081, Heilbronn, Germany, ketoma.vixkemanji@uclouvain.be and Université catholique de Louvain, Louvain Research Institute in Management and Organizations, Place des Doyens, 1, Louvain-la-Neuve, 1348, Belgium; Jean Vanderdonckt, jean.vanderdonckt@uclouvain.be, Université catholique de Louvain, Louvain Research Institute in Management and Organizations, Place des Doyens, 1, Louvain-la-Neuve, 1348, Belgium; Gerrit Meixner, gerrit.meixner@hs-heilbronn.de, Heilbronn University, UniTyLab, Max-Planck-Straße 39, 74081, Heilbronn, Germany.

2573-0142/2023/6-ART189 $15.00
https://doi.org/10.1145/3593241

Additional Key Words and Phrases: Software Development Method; Software Framework; Virtual Reality; Training; Training Scenario Authoring

ACM Reference Format:
Ketoma Vix Kemanji, Jean Vanderdonckt, and Gerrit Meixner. 2023. Towards Flexible Authoring and Personalization of Virtual Reality Applications for Training. *Proc. ACM Hum.-Comput. Interact.* 7, EICS, Article 189 (June 2023), 37 pages. https://doi.org/10.1145/3593241

1 INTRODUCTION

Humans are a multifaceted species, we have always sought ways to adapt to a changing environment, increase productivity and living standards, explore new environments and improve our understanding of the world around us. This led to the development of the locomotive engine, computer mainframes, personal computers, mobile phone, satellites, and space exploration technologies. These inventions helped improve our locomotive and information-processing abilities. Virtual and augmented reality technologies have received significant research and development efforts as a technique to enhance our perception abilities and enrich the interaction between humans and computer systems [62, 69]. While the first attempt at these immersive technologies dates back to the early 1960s, they only received prominent interest in research and development since 2012 [2]. Augmented reality technologies allow the extension of human perception while Virtual Reality (VR) technologies allow us to simulate and experience real-world events in a computer-generated virtual world. The need to enrich the interaction between humans and computers is a central theme of interactive computer system research and development. Smartphones have arguably been at the heart of interactive systems development despite being a limited human-computer interaction interface. VR systems, on the other hand, offer a more immersive, and multi-modal interaction interface. Research has shown that VR technologies can further enrich and extend the range of perception and interaction modalities between humans and computer systems, by offering immersive, multimodal perception and interaction in computer-generated virtual worlds [62, 69].

Many of the enabling technologies of VR stem from the entertainment industry; several of the VR technology developments in the last few years have been led by some research institutions, but also largely by companies in the games and movies industry [31, 45]. The use of VR in the professional context was sparsely researched in the past and only used in larger critical projects: *e.g.*, in flight simulations [31]. However, in the last few years, there has been increased interest in the use of VR technology in another professional context for skill and behavior training *e.g.*, healthcare, education, manufacturing, etc. Several pilot projects in the research community and industry have shown that the technology can be leveraged in process optimization leading to cost reduction, amongst other advantages [4, 27, 32, 38, 60].

However, Mahdi et al. [40] and other researchers acknowledge that, for a technology largely stemming from the entertainment industry, the complexities of the professional environment in which it is transferred pose challenges for the effective use of VR technology in the professional context [19, 58]. One such challenge is the need to incorporate knowledge from experts in a variety of domains in the design and implementation of effective training scenarios, Mishra *et al.* [46] discuss the interplay of three main components content, pedagogy, and technology in the design of such technical training systems. Secondly, due to the dynamic nature of their real-world counterparts, and the diversity of individual training needs; the virtual environments need to be quickly and easily adaptable to reflect different training needs, and a variety of situations encountered in the real-world [50]. Many entertainment applications are designed and shipped to the final user and the dynamics of the virtual environment are rarely modifiable from the original logic, which was already designed by the developers. In some cases where such applications are modifiable, the range of modifiable parameters is often very limited.

Furthermore, when VR technology is used in the professional context for skill and behavior training, it is desirable to not only design the dynamics of the objects in the virtual environment to match the real-world counterparts. The dynamics of the training scenario need to be adaptable during use to suit the training situation and individualized to the learning progress and needs of individual trainees. The training administrator also needs to be able to design the tasks of the users in the virtual environment, the types of learning feedback available to the user, record a wide range of performance metrics that enable monitoring of the trainee's progress, and optimize the training. These concerns assert that the development method of such applications needs to be adapted to meet the needs of the application context [48]. Therefore, the proposed methods for developing VR applications for training aim to address these issues.

Methods for developing VR applications for training seek to incorporate knowledge from application domain experts, pedagogic experts, VR engineers, human-computer interaction specialists, computer scientists, and other professions in the design of such applications [40, 50, 58]. These methods define approaches for incorporating knowledge from these different domains in the design and implementation of the VR training application. A common approach is to separate the technical implementation of the virtual environment objects from the behavior definition and semantics of the virtual objects. The behavior of the objects then can be defined with a high-level domain-specific language by the different non-technical experts. In this paper, we describe a method for developing VR training applications that incorporate knowledge and competencies from all stakeholders and a supporting software framework for enacting the method.

The main contribution of this work is a concept for the development of modular and configurable VR training applications, in an effort towards flexible intuitive VR training scenario authoring. This consists of a systematic development method providing method definitions for the specification, design, implementation, verification and operation of the VR training application. This method particularly incorporates domain expert knowledge into the development process and allows the definition of the dynamics of the virtual environment in a "simplified" user interface application, as opposed to a high-level domain-specific language. A software framework is presented to enact the method, targeted for use by developers at the implementation phase of the development. We present a study to uncover some insights on the implications of the proposed implementation approach of VR training applications for software developers. The results of this study reveal that while the approach is welcomed by developers and they have the necessary knowledge and tools to use the framework, the approach is not compatible with all aspects of the developers' current working style and they need to learn the new approach to become proficient at it.

The rest of this paper is structured as follows: Section 2 presents the context of the research and discusses related work on methods, techniques, and tools for developing VR training applications. Section 3 discusses concerns related to the domain and current practice, and based on these concerns, we formulate a set of requirements for the proposed development method and supporting framework. In Section 4, we present a proposed systematic method for developing VR training applications, based on previous research work, and current state-of-the-art and modern development approaches. Section 5 presents the development framework supporting the implementation of the proposed method. In this section, we present the architecture and implementation of the framework. In Section 6, we present a use case where the framework tool is used. We also provide a walk-through of the VRTrain framework and its use in the development process. In Section 7, we present an experimental evaluation of the development framework with experienced developers. We present the experimental procedure, outcome measures, and results. Section 8 discusses the results of the experimental evaluation. We also reflect on the proposed method and supporting framework, discussing the impact and situating the results in the current state-of-the-art and areas of further research and development. In Section 9, we conclude the paper and present the major findings and contributions.

2 RELATED WORK

2.1 Growing Interest in VR Usage in the Professional Context

There is a growing research and development interest on use of VR in the professional context for skill and behavior training. Many of the initial developments of VR technology were not geared towards application in the professional context, but for entertainment. However shortly after the technology was popularized, researchers started conducting research on the use of this technology in a non-entertainment context. Many of the developments following this era have been driven by advancements in the entertainment industry. However, there were several prototype applications in the professional context within the National Aeronautics and Space Administration of the United States (NASA) and other organizations [31], one of the more "popularized" applications of the technology in the professional context has been the use for training of soldiers [21, 51, 54].

In the last few decades advances in hardware technology have significantly improved, and the enabling devices – particularly display devices, have become much more accurate and affordable [31]. There has also been a significant advancement in software to create virtual environments [45]. Previously development and VR companies and research institutions had to create individual software and system-level libraries for authoring VR content [35, 64, 66], which was inefficient and characterized by much complexity. More recent advances in real-time software used for entertainment (in the gaming and movie industries), such as Unity and Unreal Game Engines, now provide support libraries for authoring VR content. These advancements in VR software and hardware technologies have contributed to the increased development of pilot VR projects leveraging the technology for use in the professional context [8, 16].

> *The term real-time is used to characterize systems that can respond to events within predictable and specific time constraints.*

2.2 Development Process for VR Training

The development process for early VR applications has been largely ad-hoc – not guided by any standard methodology [56, 58]. Where a pre-standard process has been used, they are often very similar to and borrowed largely from the development process of traditional immersive entertainment applications, such as games [29]. This can be attributed in part to the fact that the VR discipline is relatively new and not mature [56], but also to the case that most of the enabling technologies for VR development come from the entertainment industry [29]. Most traditional real-time entertainment applications place significant emphasis on user interaction, virtual environment fidelity, and engaging, and playful design.

Therefore, the development process of traditional entertainment applications does not directly translate to the development of useable VR applications which employ non-traditional 3D immersive and non-standard interaction interfaces. Consequently, the research community has made efforts to develop and propose standard processes and guidelines for developing VR content that borrows concepts from associated fields (*e.g.*, software engineering, game design, and development) while particularly accommodating usability and interaction aspects of VR [44, 47, 56]. Jerald [28] outlines a guideline for developing VR content with a particular focus on human factors.

However, these methodologies and guidelines focus on the development of generic VR applications without a specific focus on entertainment, professional, personal, or industrial use. There is concern that VR applications for training are particularly different from other categories of VR applications, and specific development methods should be defined for developing VR applications for training [14]. Proponents of a standard method for developing VR applications for training argue that there are significant differences when developing such immersive applications for skill and behavior training with the intent of transferring those skills to the real world [19]. VEDS describes a comprehensive development process for VR applications that includes specific steps for accommodating training [15].

To develop pedagogically oriented VR training scenarios, Mahdi et al. [41, 42] propose a stepwise process consisting of: "teachers expressing pedagogical needs", "identifying and adapting the 3D environment", "operationalization of scenarios", and "simulation and test". The authors also present a tool to assist teachers in developing virtual training scenarios; however, the tool is custom-built, limited in features compared to modern game engines, and thus has not been very much adopted. Saunier et al. [58] proposes a method by separating the role of different stakeholders that play a role in the creation of VR training environments. The authors identify four main roles: designer, job expert educational specialist, and teacher. Each of these roles actively participates and contributes to the creation of pedagogically driven training scenarios.

2.3 Techniques for Implementing End-user Programmable VR Training Applications

In the previous section, we discussed the process for developing generic VR applications and VR training applications. Some of the processes for developing VR applications are very comprehensive and include 'arguably' all the steps necessary to develop a VR application from requirements gathering to deployment [17, 56]. When developing VR training applications, it is agreed that knowledge from a wide range of experts should be considered in the process [58]. Researchers and practitioners also acknowledge that these systems often require several updates even after deployment to continually meet a wide range of training requirements [14].

However, some of these processes discussed do not provide detailed technical guidance or tools on how to implement these VR training applications to incorporate pedagogic requirements and enable non-technical administrators to configure the training system to meet specific training needs. We have discussed a number of methods in the previous section which have been proposed to enable non-technical experts to modify the training application during use to suit the required needs. The proposed methods for designing parametrized and runtime-modifiable VR training applications focus on separating expert and domain knowledge from the virtual environment and interactive objects. These methods allow the specification of the virtual environment by the expert using a domain-specific language and supporting tool to specify the semantics and ontology.

Solutions in this category include VR-Demo [66], where a designer/domain expert defines the ontology of domain knowledge that is mapped to objects in the virtual environment and also carries semantic data about the virtual environment. In the EAST project [59], and the collaborative virtual environment by Barange [5], the authors utilize a domain-specific language specification of the virtual environment and the MASCARET model [9, 12]. As part of the EAST project, Saunier et al. [59] propose a method for designing virtual environments for training. This ensures that the final environment appropriately reflects the professional real-world environment, and the implementation choices of the computer scientist do not significantly impact the environment. Some of these techniques and tools were built around custom in-house libraries, and thus were not widely adopted and due to lack of continued development have become obsolete.

Tools such as MASCARET [9, 12] were implemented for use with modern Game Engines - which have become the common preference for implementing real-time simulations and VR training applications. This approach, allowing non-technical experts to modify aspects of the VR training system to suit particular training needs, constitutes end-user programming – enabling end users to creating new digital assets [1, 6].

2.4 Commercial Software for Implementing Training-oriented and Adaptable VR Training Applications

We have discussed approaches and techniques for implementing adaptable VR training in the research. Some of the solutions discussed in the previous section were implemented a long time ago and do not meet the current needs of the industry; there is a need to develop these solutions on modern tools and a scalable technology stack. In this section, we discuss current commercial tools for achieving this purpose. There is currently a large number of startups and small and medium-sized companies that are conducting research and developing tools for prototyping and implementation of VR training. present4D is a German company which created and markets VR-Suite – a tool for creating and conducting VR training. This tool primarily uses 360° images and videos in VR to define the context in which training occurs, and trainers can include training information as overlay text, and objects. While the system can be administered with a cloud application, it offers relatively less interactivity. Additionally, 360° images and videos represent a static context that can not be adapted at run-time and is not appropriately scaled in relation to other training content. XVR Platform is developed by the company XVR Simulation BV, this tool also provides a platform for training administrators to create VR training content. XVR Platform offers seemingly a more interactive VR training experience relative to VR-Suite since it uses integrated 3D models to define the training environment context and this is relatively more modifiable to suit different contexts. Virtual Instructor Platform, Pacelab WEAVR are a similar tool created by OneBonsai, and PACE Aerospace & IT respectively. These solutions provide a cloud platform where an instructor can monitor the progress of one or more trainees and provide support for completing the training. While these systems are more generic and used for a wide range of training application scenarios; other systems such as Virtual Therapy Research Systems which is used for behavior training - are applied for more specific training use-cases. There is currently no common standard for implementing VR training applications, and limited support for an open Application Programming interface (API) allowing configuration and interoperability between these individual systems. Next, we discuss concerns and requirements for the proposed development method and supporting framework.

3 CONCERNS AND REQUIREMENTS

This section discusses some concerns from research and development and then state the requirements for the method and its supporting framework. The following concerns are identified based on related research work and reports on practical applications of VR for skill and behavior training. Based on prior related projects [34, 37, 38, 53], their research reports and documented experiences of other projects in online research databases (Google Scholar, ACM Digital Library, IEEE Xplore, Springer Link, and Elsevier), we discuss the identified concerns below.

3.1 Concerns

Concern #1: No globally recognized and adopted standard systematic method or process model for the development of parameterized VR training systems exist.

There is increasing development and research interest in the application of VR technology in a professional context for skill and behavior training [16]. With this increased interest has arisen the need for domain experts and training administrators, who often lack technical skills, to be able to personalize and modify attributes of the training system during use [59]. However, while we have discussed some contributions in the previous section to develop standards and guidelines, much of the development process of such training applications are still typically ad hoc [59] and there is no specific systematic method to support the process of developing such parameterized VR training systems.

Concern #2: Limited supporting software frameworks or packages for implementing a parameterized VR training simulation.

Tools for developing VR training applications mainly stem from the traditional game development industry. While these tools have recently been adapted to developing VR content, the development of professional training-oriented VR applications is still considered a complex endeavor [12]. Third-party frameworks have been developed which are integrated into traditional game development tools [67], and specific training tools, such as WorldViz, have been developed. However, there are limited tools to support the development of parameterized VR training systems with the traditional and more popular game development tools, such as www.unity.com and Unreal game engines. Additionally, early attempts to develop VR training applications have been done in government, security organizations, or competitive industry where publication of the methods and tools could not be made public. Therefore, attempts to make these approaches advance the public effort towards the adoption of a structured approach have been limited [15].

Concern #3: Lack of a holistic method to implement VR training applications that cover the entire range of training-relevant elements.

VR training systems consist of many different important elements: objects, behaviors, tasks, feedback, data recording, and analytics [55, 63]. Many of the methods that have been proposed define how to implement modeling of virtual objects, but do not describe how object behaviors can be programmed, how tasks and feedback are designed and implemented, and methods for tracking and collecting data. However, these different components of the VR training application are closely interrelated and a holistic approach is needed to ensure that these different elements are modular but tightly integrate with each other, to offer flexibility, performance, and a wide range of personalization options.

Concern #4: Limited understanding of the effect of different VR training elements and VR instructional design formats on training.

VR technology is relatively new [52], and the technology has proven useful in a number of application procedures [16]. However, there are limited studies and experimental results which inform our understanding of the effect of VR instructional design on a number of levels, such as cognitive processes during VR training [23, 61]. Despite the immersive nature and reported effective application of VR in training. The results have been mixed, some studies reported increased learning efficiency and memory retention [3] in VR while yet other studies [20, 43] have recorded reduced training performance outcomes owing to the cognitive load imposed by the VR interface (particularly for novice users). When designing VR applications for training, design decisions on the features of the virtual environment can affect learning [55]. Feedback provided during training is known to affect learners' behavior and progress towards a learning goal [26]. Instructional design formats for VR systems designed for training have been found to affect the outcome of training [11]. Therefore, methods and tools that enable nontechnical experts to quickly implement different VR training scenarios, tasks, and feedback, collect- analyze data and quickly iterate through this process [57] can advance the pace of research studies and improve our understanding of the effect of a range of VR training elements.

Concern #5: Technologies and obsolete methods that evolve rapidly.

The enabling technologies for the development of VR training systems are evolving rapidly [22, 31]. Many of the methods that have been proposed are a few years old, and the technologies that are commonly used in industry and research rapidly evolve, and these approaches do not immediately fit into the new technologies and methods of development.

Concern #6: Offering easy integration of methods into developer tools and workflow.
Some of the existing methods offer their own custom tools/engines for developing VR training applications, [7, 10, 18, 41], which do not fit the tools currently adopted and most commonly used in the development process. Game engines (such as Cryengine, Unity, and Unreal game engines) have become very common in industry and research institutions for implementing VR training. Since some of the proposed tools in previous research work are not compatible or interoperable with modern game engines, tools that directly integrate into these engines are needed. These tools need to deliver value with minimal disruption to proven and established workflows, leveraging existing developer knowledge and skills.

Concern #7: Ease of use for nontechnical experts and fast iteration.
Some of the proposed approaches to separate the pedagogic methods from the application logic [5, 9, 65], use high-level domain languages to implement training scenarios. Nontechnical experts can define training aspects of the VR environment using a high-level domain-specific language. However, these approaches require nontechnical experts need to learn these high-level domain-specific languages.

3.2 Requirements

Based on the concerns discussed above, in this section, we define the requirements for the method definition and supporting framework. The definition and framework of the proposed method should address these concerns directly or indirectly.

3.2.1 Method Definition.

(1) Should adhere to and be based on standard software engineering principles.
(2) The definition of the method should be useful with current (and as far as possible future) standard development processes for VR training applications.
(3) Account for incorporation of pedagogic requirements, domain knowledge, and expertise from all relevant stakeholders in the development process.
(4) Define the systematic method to implement mutable VR training scenarios.

3.2.2 Framework: Non-functional Requirements.

(1) Enable environment: Integrate into existing tools and leverage existing competencies of VR developers.
(2) Relatively easy to learn and require minimum effort to use.
(3) Reduce the effort required to perform tasks and minimize error.

Framework: Functional requirements.

(1) Enable design and implementation of configurable behaviors.
(2) Enable design and implementation of adjustable and personalized tasks, feedback, and performance tracking. Here, adjustable refers to the ability to modify the training scenario during use, after it builds and deployed. Personalization refers to the ability to adjust the training scenario to meet the specific learning needs of a particular trainee.
(3) The implemented training systems and scenarios should be interoperable with third-party training administration applications.
(4) Provide an intuitive user interface for creating personalized VR training scenarios, based on specific training needs. This intuitive user interface should be available in the core developer framework, providing a similar workflow to that for nontechnical training experts.

4 VRTRAIN DEVELOPMENT METHOD

There are established standard development processes that have been proven and used in the industry for several years. The VRTrain method does not define all steps for such processes or replace these processes. For small and medium flexible project teams or companies developing VR training applications, the VRTrain method may be sufficient and define all necessary steps in the development process. However, the method described here may only be complementary or guide larger established companies or organizations developing VR training applications.

The VRTrain method describes method definitions for implementing VR training applications with a focus on enabling easy incorporation of application domain knowledge and pedagogic techniques while also allowing easy authoring and personalization to meet specific training needs. The method described here consists of a set of method definitions for all steps of a typical development process. We describe this method using the Software and Systems Process Engineering Meta-Model Specification Version 2.0 (SPEM 2.0) [49] method definitions. These method definitions can be instantiated in process definitions as part of any development process or lifecycle. This method definition is based on state-of-the-art research discussed in Section 2, and current industry practice and guidelines. Fig. 2 shows a broad overview of the method, and we briefly describe each step in the following sub-sections.

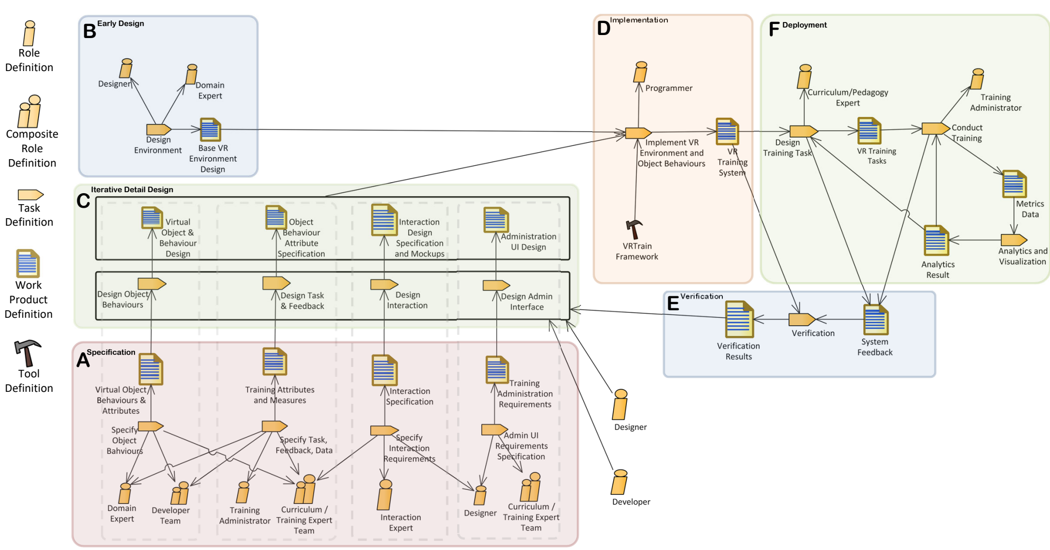


Fig. 2. Overview of VRTrain method. SPEM 2.0 Method definition.

4.1 A: Specification

When immersed in a virtual environment; users perform actions in the environment that change the properties/behaviors of objects or react to changes in the behavior/properties of objects. Jerald [30] differentiates between four types of object geometries in the virtual environment: the Background, Contextual geometry, Fundamental geometry and Interactive objects. The goal of the specification step is for stakeholders to define the training-related behaviors of the contextual, fundamental, and interactive objects of the diet (Fig. 3 shows a template for the specification of virtual objects). The requirements for the training administration application are also specified in this step.

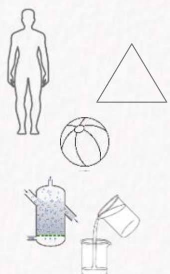
Object	Event Mutable behaviours	Behaviour Attributes	Comments
Name: *Description*: 	**Behaviour 1** *Name*: *Description*: Properties: Environment dependent/independent Time triggered. Event triggered. Interaction support, and interaction types.	**Attribute 1** Name: Description: Constraints: value constraints immediate/interpolated change **Attribute 2**: ...	
	Behaviour 2 *Name*: *Description*: Properties: Environment dependent/independent Time triggered. Event triggered. Interaction support, and interaction types.	**Attribute 1** Name: Description: Constraints: value constraints immediate/interpolated change **Attribute 2**: ...	

Fig. 3. VRTrain - behavior Specification Template.

4.2 B: Early Design

In this step, the background and contextual geometry object types [30] that are not directly related to the training simulation are designed. These objects provide contextual awareness in the environment, but will not be manipulated, actively changed, or personalized for different training scenarios. The work product of this step is a contextual environment in which training occurs. The contextual environment and objects designed in this step could represent a factory floor, hospital, laboratory, etc. In this case of the use case described in the evaluation study - Section 7, this is the classroom environment and surrounding objects.

4.3 C: Iterative Detail Design

The design of mutable object behaviors, training-related components, and administration interface is performed in this step, mainly by the designer, interaction designer, and developer (other stakeholders may also participate in a secondary role). This section makes use of the work products from the specification. The atomic behaviors of objects, tasks, and feedback designed in this step include modifiable attributes (as specified in the specification step) for creating a variety of training scenarios. This modifiable design allows the training administrator to change aspects of the VR training at setup time and at run-time for personalized training and to meet specific pedagogic needs. The implementation of the system to enable this modifiable behavior will be facilitated by the use of the VRTrain framework during the implementation phase.

4.4 D: Implementation

After the detailed design, the system is implemented. Here, the developer implements the application using work products from the detailed design. The VRTrain framework is provided as a tool at this stage to support the enforcement of the method. The implementation task can be divided into two subtasks: first implementation of the virtual environment and behaviors - where training takes place, and the second implementation of the administrative user interface - which allows the education expert, and training administrator to set up the training tasks, conduct training, and collect performance data. The VRTrain framework is used in the first step when implementing the training application (in a real-time game engine).

4.5 E: Verification

Due to the complex nature of such applications and the many interconnected hardware and software components, an iterative design/implementation approach is often recommended. Therefore, after each incremental implementation, the system is verified and it goes through another detailed design. The most important stakeholders involved in the verification step are: the Training Administrator, Pedagogy expert, Developer, Designer, and Interaction Expert. The Client and User also participate in a secondary role. Verification is also done with feedback from the education expert and training administrator when the complete system is developed and deployed in training. It should be noted that, though the model illustrated in Fig. 2 does not show a verification link to the early design, typically it is also an iterative process, just in this case it is designated for the design of immutable contextual cues, not directly affecting the training tasks.

4.6 F: Deployment

This step defines two critical tasks: Design of training tasks by the curriculum or pedagogic expert and conducting the training. The pedagogy expert designs the VR training tasks based on the curriculum guidelines. This is facilitated by the design method which allows the non-technical pedagogic expert to design tasks with a wide range of attributes based on the particular training needs. The training is then conducted by the training administrator. The training data is analyzed and also used to iteratively optimize the training. The two tasks here also produce system feedback that can be used to iteratively optimize previous steps of the method, *e.g.*, when this method is instantiated in an agile process.

5 VRTRAIN FRAMEWORK

In the previous section, we presented the VRTrain method. In this section, we will present the VRTrain framework, which is a tool to enable developers to enact this method at the implementation step of a VR training development process. While the VRTrain method provides definitions for conducting phases of the development process, the VRTrain framework primarily targets developers during the implementation phase of the development process (Fig. 4). The VRTrain method supports developers to implement the method in program implementation and building blocks for implementing the training system such that it conforms to the method definition.

5.1 VRTRain Software Architecture

This section presents an overview of the VRTrain software architecture (Fig. 6). The VRTrain framework is a software tool for enacting the development method presented above. The framework is aimed at developers for use in the implementation phase during the development process. The framework is used in the Unity game engine during the implementation of VR training applications with this game engine.

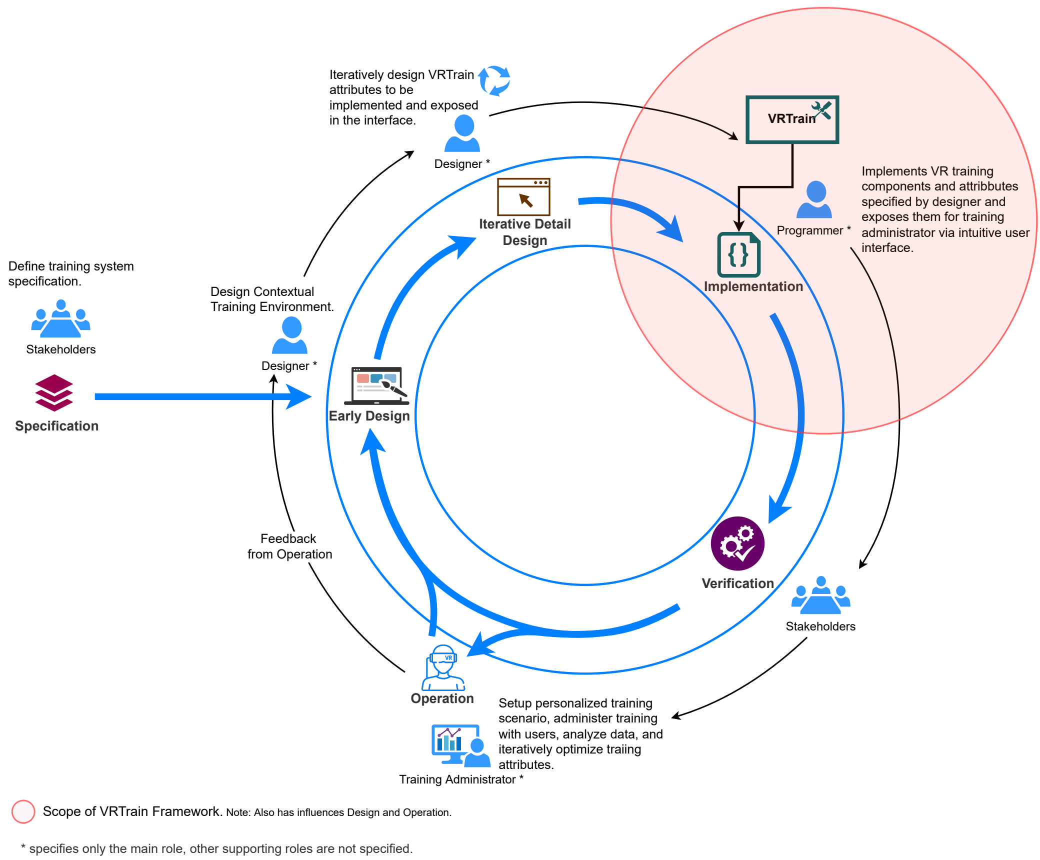


Fig. 4. VRTrain framework: target users and context of use.

The VRTrain framework consists of three main modules (Fig. 6 - orange bounding box): The behavior Module, the Task Module, and the Data Recording Module. The VRTrain framework is designed to also provide features to facilitate interoperability with third-party applications (Fig. 6 - blue bounding box) for administering VR training. Interoperability of VR training applications implemented with VRTrain framework allows communication and data exchange between the VR training application and other third-party apps. Third-party apps are mobile apps, web apps, or native apps, that are not an integral part of the VR training application, but are built purposefully as intuitive apps for authoring training scenarios by exchanging configuration data with VR training applications developed with the VRTrain framework. Non-technical users can author training applications using these third-party applications. The three core modules are now presented.

5.1.1 Behavior System. The Behavior System is used to implement and control the behavior of all training-related objects in the virtual environment. Every object in the VR application exhibits a range of behaviors or properties that change in the course of the training. VR training is often used to teach users how to change (or respond to changes in) the behavior/properties of virtual objects. The Behavior System is leveraged to implement these behaviors and properties of all objects of interest in the training. The Behavior System provides a modular implementation of behaviors and events which can be used to control the behavior or properties exhibited by any object. The developer programs the low-level implementation of behaviors, and third-party applications can be used to control the behavior of every training-related object in the environment. A training administrator can configure the behavior of every training-related object to match the specific needs of the training.

5.1.2 Task System. The task system is central to defining the training procedure. The task system provides APIs for implementing modular tasks that should be performed by the trainee as part of the training procedure. The training administrator can then configure training tasks to suit particular pedagogic requirements. The task system is designed to allow the coupling of single-atomic tasks into groups. Tasks can have relationships with other tasks that allow changes in the execution of one task to trigger changes in other tasks, *e.g.*, completing, failing to complete, or making some progress in the execution of one task can trigger another task to begin or end. The Task System is designed to be tightly coupled with the Behavior System; atomic tasks and behavior events can exchange data and trigger changes in each other. This design enables executing tasks to trigger changes in object behaviors and object changes in object behaviors can also trigger tasks.

Every task has attached data recorders (presented in section 5.1.3), and task feedback. Fig. 5 shows an overview of a task sequence. There is one main task in each scenario (The Session Goal), and in this main task, the trainer can add other tasks and the relationship between tasks. Tasks have different types of relationships:

(1) *Prerequisite*: A task can be configured to start or stop when changes occur in another task. Using this design, tasks can be set as pre-requisite to other tasks, and different types of trigger options can be set for triggering the task. Trigger options include progress on a task (a floating value from 0 to 1), the start of a task, or end of a task.
(2) *Sub-tasks*: In some cases, there is not only one way to complete a task. In such situations, a task can be defined as a group of sub-tasks. When more than one task is present in a group, the trainer can define the relationship between these tasks: all tasks in the group should be completed for the group task to be marked as complete, or completing any one of the tasks in the group will result in completing the main task.

5.1.3 Data Recording System. When performing a training task, it is necessary to monitor and collect data about events occurring in the virtual environment. The Data Recorder System is designed to collect data about events and actions occurring during the training session, and persist these data to a file or transmit the data. The VR-Train framework is designed to run object behavior, tasks, and feedback in an event-driven architecture. Each event that drives any of these training elements is automatically monitored by the core libraries in the VRTrain framework. Data about events, such as duration, start, stop, pause, and end time, are automatically recorded. The VRTrain framework is also designed to provide a base implementation for tracking and collecting data about object states such as position and movement. The base implementations can be extended by developers to create more complex reusable data tracking and collection components. As illustrated in Fig. 5, every task or task collection is designed to have data recorder components that record data about the task performance. Data about

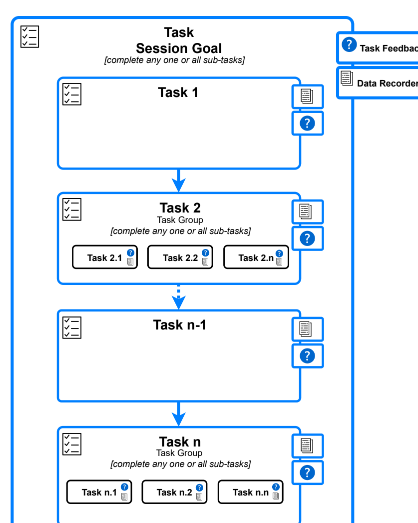


Fig. 5. VRTrain - Task System Overview.

task performance is not only used in real-time for analysis but can also be used to trigger other VRTrain events; for example, the VRTrain framework allows developers and training administrators to trigger feedback events in VR based on real-time tracked user performance data on tasks.

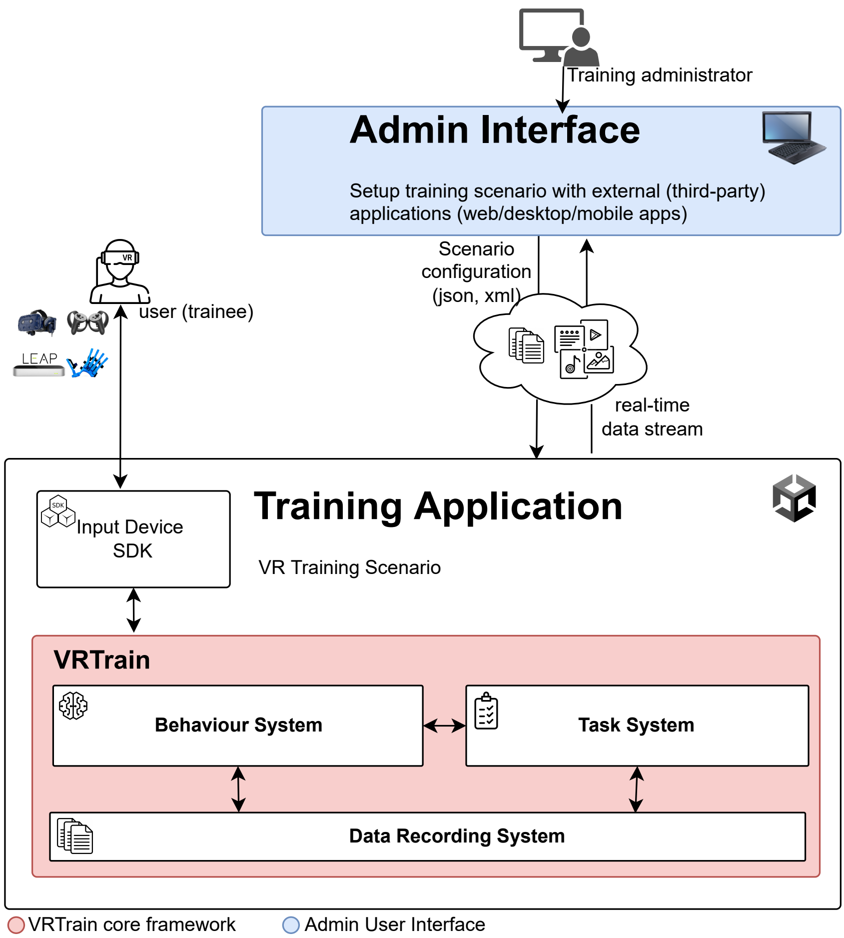


Fig. 6. VRTrain Software Architecture.

5.2 Implementation

The VRTrain framework is implemented in the Unity Game Engine. The Unity Game Engine is specifically chosen for the following reasons:

(1) To introduce minimal changes to established developer workflows and to increase the chances of mass adoption, it was important to implement the framework in a way that integrates directly into the developer workflow. Since Unity is a tool that is used by many real-time interactive system developers, the VRTrain framework was designed to integrate directly into this game engine and workflow.
(2) Many previous tools for developing parameterized and training-oriented VR applications have been based on custom real-time simulation libraries, which have become obsolete and not used by many modern VR projects. Therefore, the VRtrain framework was designed to

be used in modern game engines that are widely adopted in and used for the development of real-time VR application. For this reason, the implementation of the VRTrain framework presented in this paper was implemented in the Unity game engine.

(3) Unity is arguably the most widely adopted game engine used in many research and industry projects. Therefore, to increase adoption of the method and framework, the VRTrain framework was implemented for use with the Unity game engine. Implementation for use in other similarly widely adopted game engines such as Unreal Engine and CryEngine can be implemented in further works.

At its core, the VRTrain framework follows an event-driven publish-subscribe architecture but provides interfaces that can be implemented by developers to provide the specific logic for every element of the training system. The three core modules of the VRTrain framework ("Behavior System", "Task System" and "Data Recording System") are primarily driven by events as described in Section 5.2.1 below. The base class of *VTEvent* provides the fundamental behavior of VRTrain events. Each of the three subsystems extends this base class and provides additional behavior to control events within the subsystem. *VTEvent*s have a type property which determines whether the *VTEvent* is a Behavior type event, a Task System type event, or a Data Recorder System type event. Depending on the type of *VTEvent*, it can have different properties. *VTEvent*s are the primary driving elements of all training-related events that occur in the virtual environment.

*VTbehavior VTEvent*s are used to drive the behavior and properties of training-related behaviors of virtual objects. The behavior system provides C Sharp programming language (C#) base classes that developers can extend and implement their own custom object behaviors. All custom developer behaviors extending this base class will automatically be usable in all corresponding subsystems in the VRTrain framework; for configuring object behaviors, for triggering training tasks or task feedback, or data recorders, etc.

*VTTask VTEvent*s and *VTTaskFeedback VTEvent*s type *VTEvent*s are used to drive tasks and task feedback, in the Task System, respectively. The Task System provides C# base classes for VRTrain tasks - *VTTask*, and VTrain task feedback - *VTTaskFeedback*. All custom developer training task classes that extend *VTTask*s are automatically usable with *VTTask VTEvent*s to define training tasks, and as task prerequisites or subtasks in a task group as described in Section 5.1.2. A wide range of feedback can be provided during training, *VTTaskFeedback* provides a C# base class for developers to implement custom training feedback behaviors. All developer custom feedback classes extending *VTTaskFeedback* can be automatically used to set up feedback for training tasks, and as a trigger or response to events in the behavior system, other tasks, or data recorders.

VTDataRecorder Events are used to drive data recording events for training tasks. The Data Recorder System provides a base C# class for implementing data recorders that collect and persist data about a wide range of events occurring in the virtual environment during training. All custom developer data recorder classes that extend *VTDataRecorder* are automatically usable in the VRTrain training scenario setup process.

5.2.1 VRTrain Event Dispatcher.

Fig. 7 illustrates the class diagram of core VRTrain event system classes. We describe the order of its components as depicted in Fig. 7.

(1) Objects in the virtual training environment, have one or more behaviors. VRTrain provides base behavior classes that can be extended by developers to implement specific behaviors depending on the application context or use case.
(2) All Behavior types extend the base class VTConfigurableType. This configurable type class defines behavior of the class which allows it to be modifiable through exposed fields. Configurable types can have one or more fields that allow the behavior to be changed at run

time, depending on the Value property of it's VTFields. VTFields are exposed to external third-party applications which can change the value of these fields.

(3) EventlistTimelines define a sequence of events that occur during the training. Each training-related object in the scene has one event timeline that defines all the sequence of behaviors the object will exhibit in the course of the training session. Event list timelines are data-driven, and in this case are based on the Unity Scriptable Object class. This makes it such that each event timeline is self-contained and independent such that it can be assigned to any object or to more than one object. VTEvents contained in the event list timeline provide provides values for fields of the configurable type behaviors of the object. Event list timelines and VTEvents are both exposed to modification by external third-party applications.

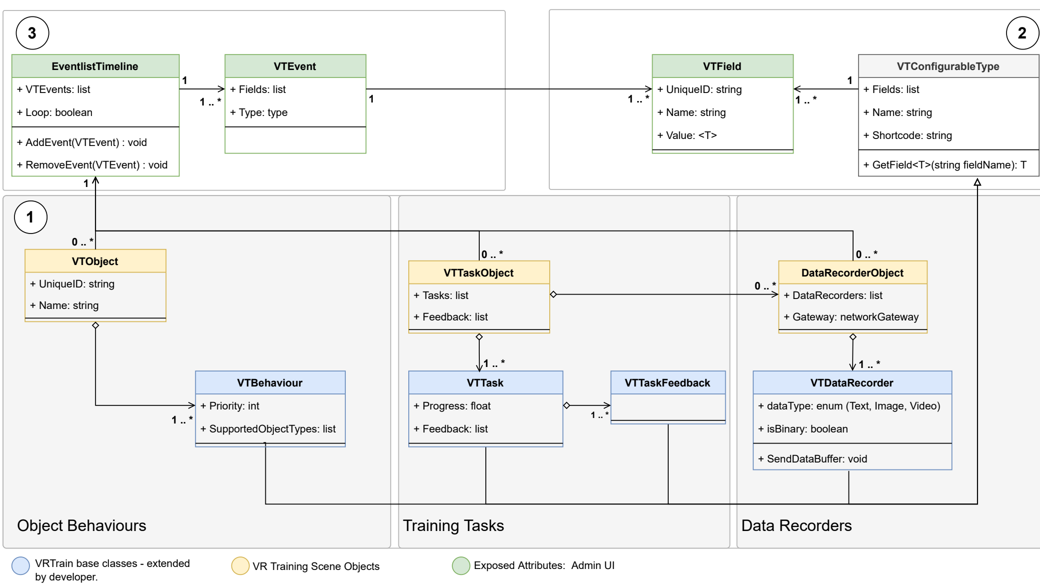


Fig. 7. VRTrain Event Dispatcher Class Diagram.

5.2.2 *Authoring VR Training Scenarios.*

Personalized and pedagogy-driven VR Training scenarios can be set up as shown in Fig. 8; for VR training applications implemented using VRTrain method and framework. We describe the different components in the order illustrated in Fig. 8 below:

(1) Developer implements training scenario (object behaviors, tasks, and task feedback, and data recorders) using VRTrain interface and base classes.

(2) During the training, the VRTrain framework communicates with the developer's custom behaviors, dispatching events that trigger behaviors, tasks, feedback, and collecting data. Prior to the start of the training, and during the training, the collection of events and triggers are serialized to JSON data format and exported via a network gateway to external third-party applications. The final setup of the training can be communicated back to the training application as a JSON message; where it is imported, deserialized, identified, and sent to the VRTrain event dispatcher which is used to trigger the sequence of events that occur during training.

(3) These third-party applications (web, mobile, or native PC applications) allow experts to modify the events that occur during training, based on the application domain knowledge

and training pedagogy requirements. These applications also receive analytic data which can be used to analyze training performance, and personalize the training for a specific trainee.

(4) Since the VRTrain framework is designed to be driven by external applications, a trained AI agent can also be used to drive the sequence of events that occur during training. This functionality can be implemented as an extension, not built into the VRTrain framework.

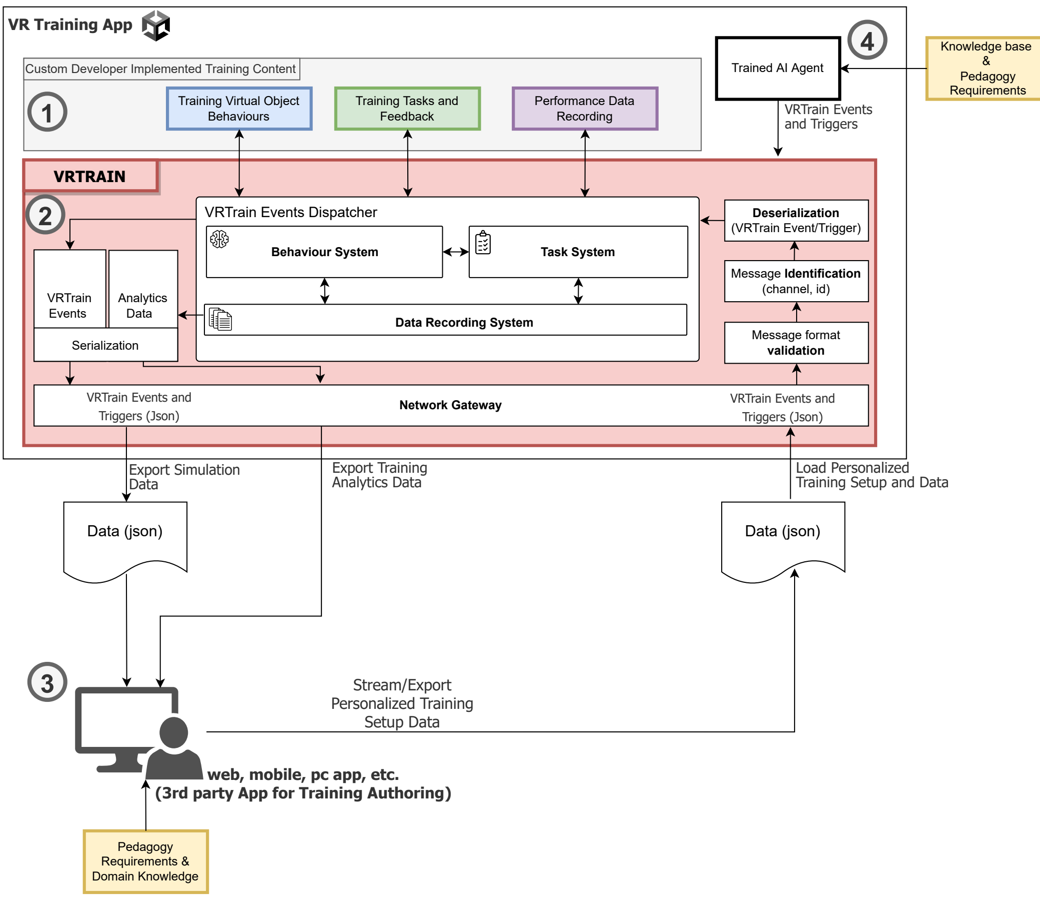


Fig. 8. VR Training Simulation: Development and Authoring with VRTrain.

6 USE CASE AND SYSTEM WALKTHROUGH

The VRTrain framework has been used in research and training projects for driving simulation, laboratory operation training, and teacher training. In this section, we discuss its use in the development of an application for training teachers and also provide a walk-through on using the framework in the development process (download and documentation are accessible online [1]).

For this use case, the goal is to train teachers on how to deal with a variety of situations in the classroom. The teacher should be able to identify specific behaviors of the school children in the classroom and respond to those situations that need the teacher's attention. The classroom is a virtual classroom, and all children in the class are virtual. Fig. 12 shows a view of the virtual

[1] https://bit.ly/vrtraindoc

classroom. All virtual characters in the room are static, the character behaviors, tasks, and feedback should be implemented with the VRTrain framework such that a wide variety of specific behaviors can be simulated based on the specific training needs.

Fig.10 illustrates the steps to use the VRTrain framework. To set up a Unity project for use with the VRTrain framework, the developer first downloads, and imports the framework into the project. The following steps are taken (in the order depicted in Fig. 10) after starting the project.

6.1 Step 1: Create Custom Behavior Scripts

The VRTrain framework provides classes with built-in methods for running behaviors, tasks, feedback, and data recorders. When a developer's custom class extends these built-in classes, it inherits these methods, and the custom class is automatically usable in VRTrain events for building personalized VR training.

For this use case, behaviors that enable the students to look at objects, speak, turn, and perform specific seated movements (sleeping/head resting on the desk, hand gestures, turning around, fidgeting, etc.) were implemented. A supervision task was implemented where the teacher should identify a situation and will be required to walk to the position in the classroom. If the teacher does not walk to that position, additional feedback in the form of an arrow pointing to the location and voice commands were issued. The time required to complete the task, and the teacher's movement path in the classroom were recorded.

6.2 Step 2: Create Custom Fields for the Behavior

To create personalized scenarios adapted to pedagogic needs, VRTrain uses fields. Each custom type that is created in VRTrain, can define which fields can be used to customize its behavior at runtime. Fields provide data for customizing behavior at runtime and thus have different types depending on the type of data they supported (number, text, texture, sound, animation, etc.). To ensure portability and re-usability custom types do not have to define their own field types, but simply declare which field types they want to use. VRTrain provides a behavior manager asset that automatically identifies and curates all VRTrain types (behaviors, tasks, feedback, and data recorders) created by the developer in the project and lists these types in one of four type managers: behavior Manager, Task Manager, Task Feedback manager, and Data Recorder Manager. Fig. 9 illustrates the behavior type manager and the available behavior setup options. Using these type managers, developers can define:

(1) Fields, field types, and field names that will be used by the custom type. The developer can define field names, and descriptions which will aid non-technical experts understand the purpose of those fields when setting up their training scenarios.
(2) Pre-configured object templates that will be instantiated by the VRTrain framework at runtime when an event corresponding to the specific task, feedback, or data recorder is running.
(3) When setting a training scenario, behaviors can run concurrently. While concurrency should be allowed for some behaviors, other behaviors should not run concurrently (*e.g.*, a person character cannot eat and talk concurrently). Thus the behavior manager provides developers the possibility to set:
    (a) behavior exclusion: behaviors that can not run concurrently.
    (b) behavior priority: When two behaviors are scheduled to run at the same time, the behavior with higher priority takes precedence. Note: this also depends on the VTEvent level priority settings as explained in Step 4.

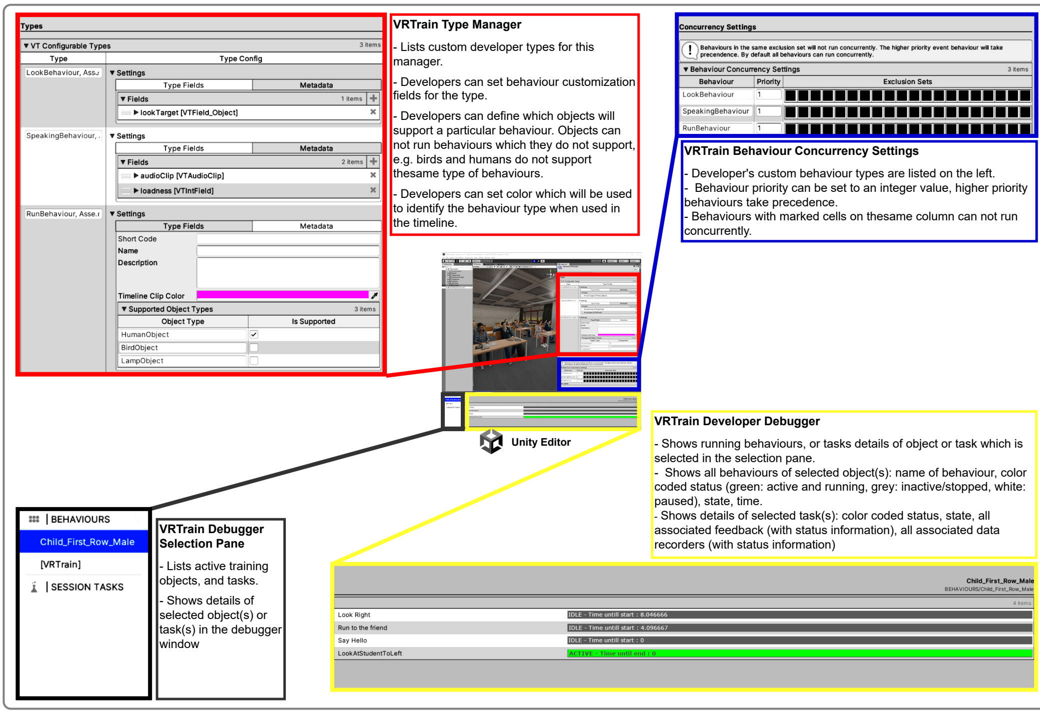


Fig. 9. Step 2: VRTrain Custom behavior Type Manager.

6.3 Step 3: Implement Behaviors using Fields

After defining VRTrain type settings and fields in the corresponding type manager, the developer proceeds with custom implementations of the behaviors, tasks, feedback or data recorders using the fields which have been defined in the manager. The VRTrain framework promotes re-usability and is designed to reuse fields, behaviors, tasks, feedback, data recorders, etc. Therefore, developers do not have to implement all atomic behaviors for a training scenario, but can import existing behaviors from other projects or other developers and use them seamlessly in their training system.

6.4 Step 4: Create VTEvents

Creating custom implementations of types in the previous step concludes the developer-centric tasks of the VRTrain development method. In this step, the developer creates training scenarios in the Unity game engine using a typical workflow which will also be used by non-technical experts for creating personalized training. The developer creates sample training scenarios, using the custom types created in the previous step. Fig. 11 illustrates a sample setup of a behavior timeline that can be assigned to any object to control it's behavior throughout the training session.

6.4.1 Behavior System. The VRTrain behavior System provides timelines for creating object behaviors. A timeline consists of one or many VRTrain behavior events. A timeline can be assigned to an object, and it determines the behaviors exhibited by that object trough out the training. A timeline can be assigned to any object that supports the behavior events in the timeline. A training expert can create and assign behavior events for a timeline before runtime, to suit specific training needs, or they can be created and assigned programmatically at runtime by an Artificial Intelligence agent.

6.4.2 Task System. Tasks to be executed by the training expert are created using VRTrain task events. As described in Section 5.1.2, a task event can be related to another task event via grouping (with AND or OR relationship), or as a prerequisite. Task events can also be associated with VRTrain feedback events that provide a variety of feedback to the trainee on task execution and progress.

6.4.3 Data Recorder. VRTrain data recorder events are trackers that record data about actions occurring in the virtual environment for analytics. Build-in data recorder automatically records data about all VRTrain behaviors. As described in Section 5.1.2 additional data recorders can also be associated with VRTrain task events and record information about trainees' performance on tasks.

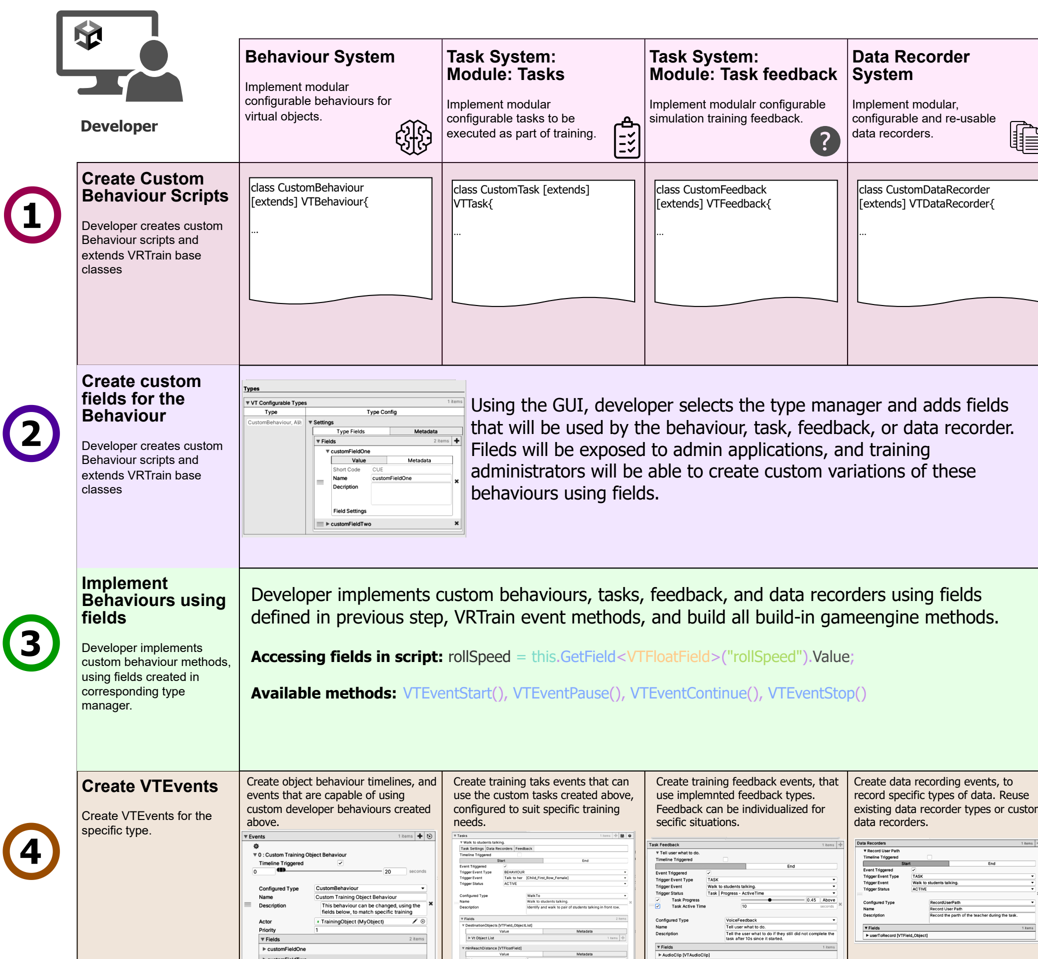


Fig. 10. Walkthrough, developer steps when implementing VR training with VTRain.

7 EVALUATION STUDY

As described in Section 4, development of VR training requires expertise of a diverse pool of experts. The VRTrain framework is a tool for enacting the method at the implementation phase of the development process, thus the evaluation was primarily aimed at developers. Since the evaluation is focused on development with the VRTrain framework, 3D content designers do not participate

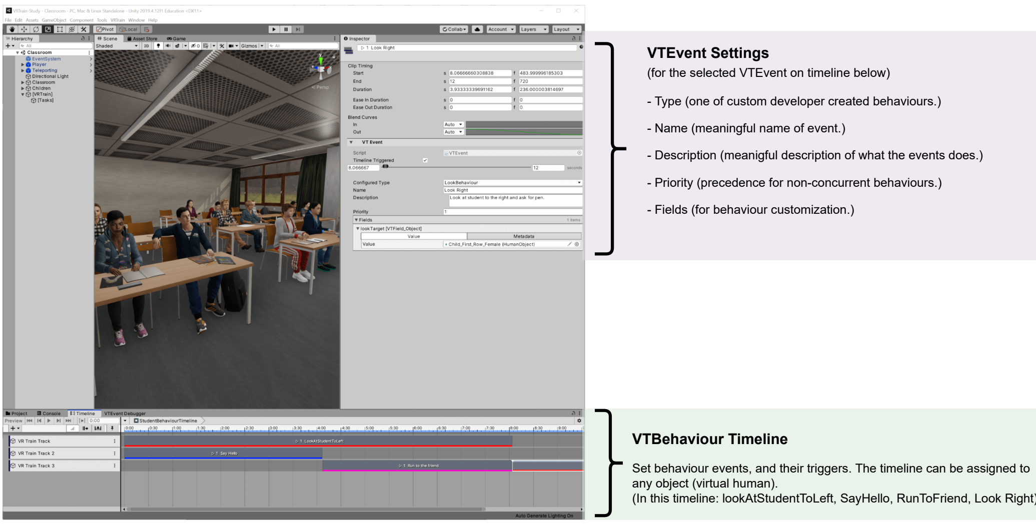


Fig. 11. Step 4: VRTrain behavior timeline and VRTrain Event.

Fig. 12. Virtual classroom for training teachers to deal with classroom situations.

in the study. To keep the entire length of the study reasonably short, we, therefore, provided participants with an already designed environment so they could focus on the development tasks. The goal of this study was to investigate the following research questions:

Research Questions:

RQ1 How difficult is the setup process and use of the VRTrain framework?
RQ2 What is the acceptance of the framework by developers?
RQ3 How does the use of VRTrain framework affect the developer workflow?

The VRTrain framework was evaluated in a study with seven external developers (out of the core team which developed the framework) who were using the framework for the first time. Developers were specifically selected who have experience developing real-time applications with the Unity game engine. Participants were provided with the virtual classroom with students, described in Section 6, but without any behaviors, tasks, feedback, or data recorders. Participants were provided with a pre-designed virtual environment to ensure that they focused on programming tasks with the VRTrain framework and not the virtual environment design. The goal of the study was for participants to implement a similar scenario for training teachers to identify situations in the classroom and respond to those situations. The starting project was provided to participants on GitHub. The study was designed to let developers work on a wide range of tasks with the VRTrain framework, and to mimic as closely as possible the natural developer workflow. Since developers worked with multiple components of the VRTrain framework and some of the tasks required some brainstorming; instead of packing all these in a few minutes of study, the developers were allowed the flexibility to split up and complete sub-tasks over multiple sessions. The entire study lasted two weeks, but the effective development time was five to nine hours.

7.1 Experiment

In this section, we report the experiment procedure according to the structure of Ko *et al.* for reporting experiments in the field of Human-Computer Interaction [36]. The study was conducted remotely, communication was done over the Microsoft Teams platform. The study protocol and all resources needed to complete the study were provided to participants online.

7.1.1 Recruitment. A total seven participants, six male, and one female, were recruited for this study. The average age of participants was 25.4 years (*M*=25.4, *SD*=4.03). All participants had experience developing real-time applications with the Unity game engine, as programmers or programmers and designers.

7.1.2 Informed Consent. Prior to starting the study, participants were informed of their rights, potential risks and discomforts, and confidentiality. Only participants who agreed to and signed the GDPR-compliant consent document were allowed to proceed with the study. All data collected in the study were kept anonymous and handled in compliance with the relevant local data protection regulations. After signing the consent form, participants filled out a questionnaire collecting anonymous demographic data about the participant.

7.1.3 Group Assignment. All participants performed the same set of tasks in the study, so there was no group assignment. After filling out the consent form and demographic questionnaire, participants proceeded to execute the study tasks.

7.1.4 Training. After completing the demographic questionnaire participants watched a 48 minutes video covering both VRTrain concepts. The first part of the video covered the main concepts: background to the subject, objectives, and goals of the VRTrain method. The second part of the video covered the technical fundamentals of how to use the VRTrain framework, and a technical demonstration (tutorial) on the process of implementing VR training applications with the VRTrain framework.

7.1.5 Study Tasks. The entire study consisted of seven tasks.

(1) *Watch the video and complete pre-study acceptability questionnaire.*
Participants watch the 48 min. introductory video and complete the pre-study acceptability questionnaire.

(2) *Download evaluation project and VRTrain framework. Set up VRTrain project and fill the setup questionnaire.*
Participants had to clone the project from GitHub on their local computers and download the VRTrain framework separately. After obtaining a copy of the VRTrain framework, participants had to import the VRTrain framework into the project and set up the project ready for use with the VRTrain framework. After setting up the project for use with the VRTrain framework, participants responded to the VRTrain setup questionnaire.
(3) *Development with VRTrain task 1: Implement looking behavior.*
Users were expected to use the VRTrain framework, to implement configurable control of the looking behavior of virtual human characters in the environment. The characters should be able to turn and look at objects in the environment. The look target should be configurable with the VRTrain framework.
(4) *Development with VRTrain task 2: Implement "speaking" behavior.*
In this task, participants were expected to implement configurable control of the speaking behavior of virtual human characters in the environment. The characters should be able to speak, and the speaking behavior (*e.g.*, what the character is supposed to speak) should be configurable with VRTrain fields.
(5) *Development with VRTrain task 3: Implement "walk to" task.*
In this task, participants were expected to use the VRTrain framework to make a pair of students start a conversation (re-using behaviors from the previous step and the collaborative event timeline feature of VRTrain framework). Secondly, participants were expected to implement a "supervise students" task: this task requires the first-person player (VR user or teacher) to supervise students by walking up to the pair of students talking [with keyboard navigation]. Participants were also expected to implement configurable task feedback, using the VRTrain base classes, to provide task help if the teacher (VR user) does not intervene after 10s.
(6) *Development with VRTrain task 4: Create script scenario.*
At this step participants had completed all programming tasks. In this task, participants were expected to use the work products from the previous tasks to create a typical training scenario using the VRTrain framework. A scripted sequence of events that should occur in the classroom was provided to the participants, and they would implement the training scenario without writing any code, but using the VRTrain framework user interface elements. The script consisted of what the students in the class should do (start a rowdy conversation), a task for the teacher to react to the students' actions (identify the pair(s) of rowdy students and walk up to them), tracking the teacher's performance and providing feedback based on the performance (if the teacher does not identify the situation or moves to the students after 10s, tell the teacher what to do).
(7) *Complete post-study acceptance questionnaire.*
After completing all programming tasks in the study, participants completed the post-study acceptance questionnaire. The post-study acceptance questionnaire contained the same questions as the pre-study acceptability questionnaire.

7.1.6 Outcome Measures. The following data was collected during the study.

Pre-Study Acceptability and Post-study Acceptance: A pre- and post-study acceptability questionnaire, both based on the technology acceptance questionnaire UTAUT [68], was completed by participants. The pre-study acceptability questionnaire measures participants perceived acceptability of the VRTrain method and framework before they get the opportunity to use the framework in a project. The pre-study acceptability questionnaire was completed by participants after watching the introductory video which presented the VRTrain method and framework (Section 7.1.4). The post-study acceptance questionnaire evaluates participants' acceptance of the VRTrain method and framework after they have had the opportunity to use the framework in a project. The post-study acceptance questionnaire was completed by participants after executing all the study tasks described in Section 7.1.5. These questionnaires were used to evaluate participants' acceptability and acceptance of the VRTrain method and framework along four scales:

(1) Effort Expectancy: How easy is it for users to learn to use the system, become skillful at using the system, and get the system to do what they want it to do?
(2) Performance Expectancy: Evaluated if using the system makes it easier for users to complete their job, attain higher quantity and quality results in their work, and reduce time spent on routine tasks.
(3) Attitude Towards Use: Evaluates if users consider use of the system to be a good idea, if it makes work pleasant or interesting, and look forward to using the system in their job.
(4) Facilitating Conditions: Evaluates if users have the necessary knowledge, competence, and compatible tools to work with the system and if it is compatible with aspects of the user's work and work style.

Work Load Evaluation: The workload of each of the study tasks was also evaluated using the NASA TLX questionnaire [25]. Participants completed the NASA TLX questionnaire to evaluate:

(1) The setup process of the VRTrain framework, *i.e.*, setting up a Unity real-time simulation project for use with the VRTrain framework. The questionnaire was completed by participants after downloading and setting up the study project for use with the VRTrain framework.
(2) Each of the development tasks in the study procedure. Participants completed the NASA TLX questionnaire after completing each of the programming study tasks where they used the VRTrain framework to accomplish the task.

7.2 Results

In this section, we present the results of the VRTrain evaluation study.

7.2.1 Pre-Study Acceptability and Post-Study Acceptance.

Effort Expectancy. Effort Expectancy measures the expected amount of effort participants will require to learn and effectively use the system (Fig. 13). Regarding the *pre-effort expectancy*, the internal consistency is estimated as reliable enough (Cronbach's α=0.55, which is interpreted as 'minimal', but Spearman's ρ=0.93 and Guttman's λ_4=0.87, which are interpreted as 'good'). The interrater reliability is estimated as moderate (Kendall's W=0.29, df=5, p=0.067, *n.s.*). Since the distribution of the answers does not follow a normal distribution (Kolmogorov-Smirnov normality tests did not pass with α=0.05), we computed for each question a Wilcoxon signed-rank test for a simple sample with ties and continuity correction (with 10,000 iterations) to determine whether answers are significantly departing from the median value MD=3, either above or below (for example, for question Q5). Indeed, a question could receive an average answer above the median value, but not significantly. It is usually admitted that the average should be above the median MD=3. Q1, Q2, Q3, Q4, and Q6 are positive statements and all received an average above the median value MD=3, but only Q4 (M=3.86, SD=0.99, p=0.046*) with a large effect size (r=0.61) and Q6 (M=3.86, SD=0.35, p=0.015*) with a very large effect size (r=0.88) were significantly above the median. Similarly, Q5 (M=2.14, SD=0.35, p=0.015*), the sole negative statement, is significantly below the median. The average pre-study effort is above the median (M=3.40, SD=0.95), which denotes a reasonable average value. We repeated the same calculations for all statements for all scales. Therefore, from now on, we only summarize the main salient results.

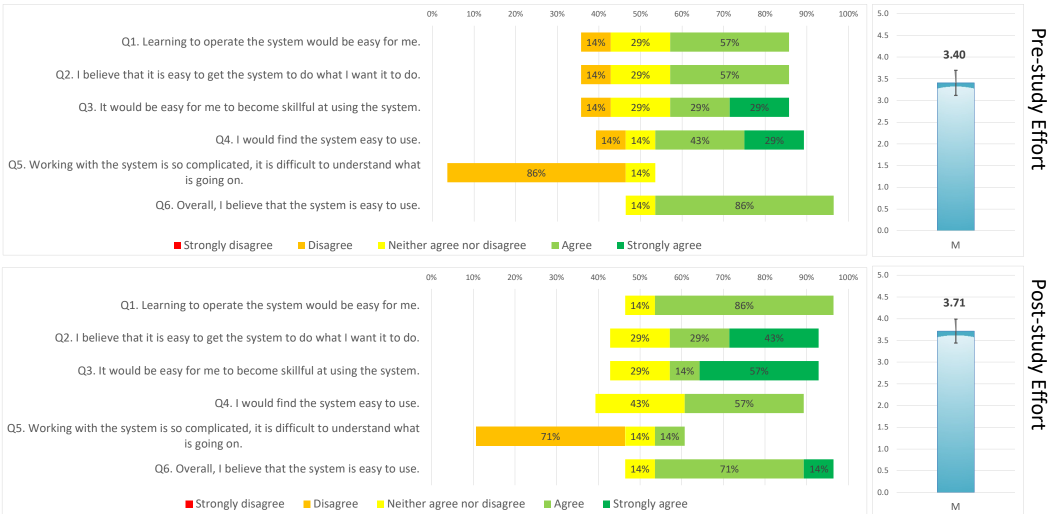


Fig. 13. Effort Expectancy: Pre-study (top) and Post-study (bottom).

All corresponding measures for the *post-effort expectancy* are improved with respect to the pre-effort: consistency (Cronbach's α=0.67) is better, interrater reliability is significantly better (Kendall's W=0.36, df=5, p=0.027*), averaged value is maintained above the median and slightly improved (M=3.71, SD=0.91). More important, this averaged value is highly significantly better than for the pre-study (a Wilcoxon signed-rank test for the two paired samples returned p=0.0038** for one tail and p=0.0076** for two tails) with a medium effect size (r=0.40), which suggests a really positive evolution from the pre-study to the post-study.

Performance Expectancy. Performance expectancy measures how much users expect to perform at their job when using the system (Fig. 14). Regarding the *pre-study performance*, the consistency is simply excellent (Cronbach's α=0.90, which is interpreted as 'very good', Spearman's ρ=0.90 and Guttman's λ_4=0.96, which are interpreted as 'very good'), thus suggesting that participants were highly consistent among them in answering the performance questions. The interrater reliability is estimated as moderate (Kendall's W=0.39, df=5, p=0.16, *n.s.*). Question Q1="Using the system would make it easier to do my job" (M=2.71, SD=1.03, p=0.31, *n.s.*) is the only question averaged below the median, but not significantly; hopefully, thus suggesting that the perceived performance of VRTrain was not convincing at first glance. All other questions are averaged above the median, none of them significantly.

Regarding the *post-study performance*, the consistency is again as excellent as it was for the re-study (Cronbach's α=0.91, which is interpreted as 'very good', Spearman's ρ=0.94 and Guttman's λ_4=0.96, which are interpreted as 'very good'), thus suggesting that participants were highly consistent among them in answering the performance questions, both during the pre- and the post-study. The interrater reliability is this time estimated as minimal, but not significantly (Kendall's W=0.12, df=5, p=0.48, *n.s.*). All questions are average above the median, and only Q5="Using the system improves the quality of the work I do" (M=3.71, SD=0.45, p=0.007** with a large effect size of r=0.89) and Q6="If I use the system I will spend less time on routine job tasks" (M=3.57, SD=0.73, p=0.008** with a large effect size of r=0.87) were significantly departing from the median, thereby suggesting that work improvement and time reduction were perceived as more pre-eminent. However, the averaged pre-study performance (M=3.65, SD=1.39) was not significantly different (a Wilcoxon signed-rank test for the two paired samples returned a z-score z=0.80, *n.s.*) from the post-study (M=3.45, SD=0.96).

Fig. 14. Performance Expectancy: Pre-study (top) and Post-study (bottom).

Attitude Towards Use. Attitude towards use evaluates users' perception of use of the system in their jobs (Fig. 15). Regarding the *pre-study attitude*, the internal consistency is estimated as moderately reliable (Cronbach's α=0.57, which is interpreted as 'minimal', but Spearman's ρ=0.35 and Guttman's λ_4=0.82, which are interpreted as 'medium', respectively 'good'). The interrater reliability is estimated as low (Kendall's W=0.09, df=4, p=0.62, *n.s.*), which suggests that there was little agreement among participants regarding aspects linked to what attitude to adopt with respect to VRTrain in the pre-study. The averaged attitude (M=4.00, SD=0.76) has reached the excellent threshold, which is one unit above the median value (3 in our case). Q1, Q2, and Q3 were significantly averaged above 4 while Q4 and Q5 were averaged 3.57 and 3.86, respectively, but not significantly above the median.

Regarding the *post-study attitude*, the internal consistency is reliable (Cronbach's α=0.74 is interpreted as 'good', Spearman's ρ=0.56 and Guttman's λ_4=0.96 are interpreted as 'medium' and 'very good', respectively). The interrater reliability is estimated as medium (Kendall's W=0.29, df=4, p=0.08, *n.s.*). Three questions out of five from this questionnaire were assessed as significantly above the median value: Q1="Using the system is a good idea" (M=4.29, SD=0.70, p=0.015*, with a large effect size of r=0.81), Q2= "I like the idea of using the system" (M=4.25, SD=1.64, p=0.00715**, with a large effect size of r=0.88), and Q4="The system makes work more interesting" (M=3.86, SD=0.99, p=0.046*, with a medium effect size of r=0.58). The other two questions are also averaged above the median but not significantly. Having the five statements of this questionnaire all perceived as very positive suggests that participants felt that using VRTrain was beneficial for their work, in particular after having used it.

However, the averaged pre-study attitude towards use (M=4.00, SD=0.76) slightly decreased for the post-study (M=3.94, SD=0.83), but not significantly (a Wilcoxon signed-rank test for the two paired samples returned a z-score z=0.38, p=0.36, *n.s.*). Participants were more aware of the investment required by the test after completing the study, but this difference remains marginal.

Facilitating Conditions. When deploying a new system or method it is desired to introduce very little unnecessary changes as possible, thereby raising the chances of quick and easy adoption. A requirement for the VRTrain framework was to deliver value with minimal disruption to proven and established workflows, leveraging existing developer knowledge and skills. Therefore, we

Fig. 15. Attitude Towards Use: Pre-study (top) and Post-study (bottom).

evaluate if users have the necessary tools, knowledge, and expertise to use the new system (Fig. 16). Regarding the *pre-study facilitating conditions*, the internal consistency is reliable (Cronbach's α=0.75 is interpreted as 'good' and Guttman's λ_4=0.91 are interpreted as 'very good'). The interrater reliability is estimated as medium (Kendall's W=0.37, df=5, p=0.022*).

Although all statements of this questionnaire were positively phrased, two of them were answered with an average below the median: Q4="Using the system is compatible with all aspects of my work" (M=2.86, SD=0.35) and Q6="Using the system fits into my work style" (M=2.57, SD=0.90), thus suggesting that the model-based approach with parameters adopted in VRTrain represents an unusual way of working for familiar developers. This does not mean that this way of work is not appropriate, as other questions are averaged above the median with the peak reached for Q3="Given the resources, opportunities and knowledge it takes to use the system, it would be easy for me to use the System" (M=4.67, SD=1.11) which is significantly different (p=0.015*) with a large effect size (r=0.81).

Regarding the *post-study facilitating conditions*, the internal consistency is reliable (Cronbach's α=0.68 is interpreted as 'good' and Guttman's λ_4=0.84 are interpreted as 'good'). The interrater reliability is estimated as medium (Kendall's W=0.38, df=5, p=0.0059**). After completing the study, this questionnaire received a significant improvement regarding three questions: Q1="I have the resources necessary to use the system" (M=4.54, SD=1.63, p=0.015*), Q2="I have the knowledge necessary to use the system" (M=4.00, SD=1.07, p=0.046*), and Q3="Given the resources, opportunities and knowledge it takes to use the system, it would be easy for me to use the System" (M=4.42, SD=1.32, p=0.0014*), which was already the case for the pre-study. Nevertheless, Q4="Using the system is compatible with all aspects of my work" (M=2.57, SD=1.18, p=0.21, *n.s.*) received the most negative assessment (29% of participants expressed reservations not about the use of the system, but about its suitability with current development practices). This observation is pretty well expressed by a participant's comment: "*Although this framework does not cover what I'm currently doing in my job, I think it may be useful there someday, especially for projects with complex, changing scenarios. I think it may be useful to have provided with the framework some more premade scripts that would potentially be commonly used, like for example voice or pointer feedback, that would speed up the initial work*".

The averaged facilitating conditions were perceived more positively after the post-study (M=3.60, SD=1.18) than in the pre-study (M=3.29, SD=1.57) and significantly (a Wilcoxon signed-rank test for the two paired samples returned a z-score z=2.04, p=0.02*) with a small effect size (r=0.22), thus suggesting that the developers' feeling that they have the resources and knowledge to use VRTrain, but that its approach is not traditional and not familiar with the practices they know.

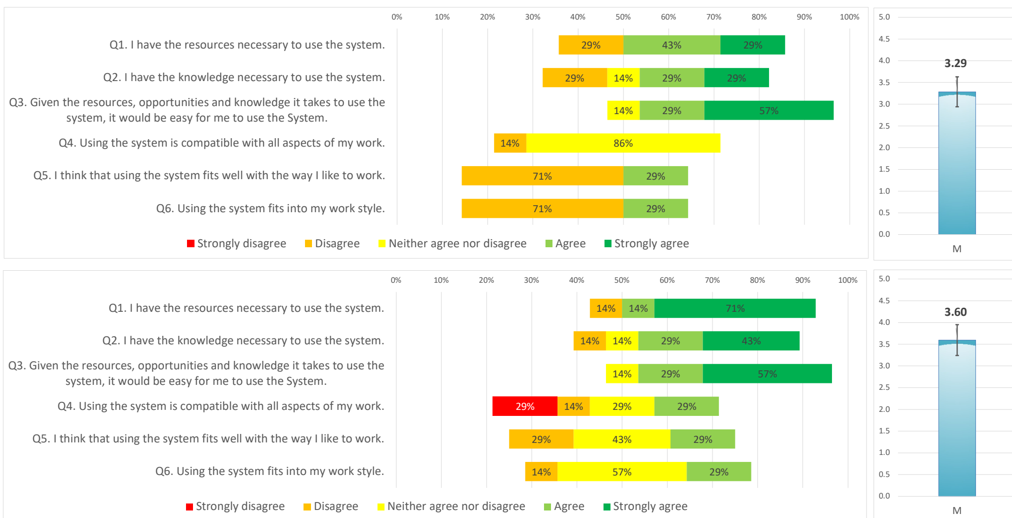


Fig. 16. Facilitating Conditions: Pre-study (top) and Post-study (bottom).

7.2.2 *Workload Assessment.*

Task Difficulty: Likert-scale evaluation. To capture how the various steps of the protocol were subjectively perceived by participants, they were invited to express their ease of completing each task on a 5-point Likert [39] scale. Overall, all steps were perceived as very easy (M=4.52, SD=0.66) to perform (Fig. 17): the setup process was very well accepted (M=4.86, SD=0.35, p=0.0078**), task 2 was the easiest (M=5.00, SD=0.0, p=0.0078**), followed by task 4 (M=4.83, SD=0.37, p=0.015*), task 3 (M=4.00, SD=0.58, p=0.031*) to end up with task 1 (M=3.86, SD=1.00, p=0.031*), which is normal since it was the first real task to perform. A participant's comment also reveals this: "*Hello, I find the framework to be a really cool idea. I think it is really hard, to get to understand how the framework has to be configured but once I could get it running it was just as easy as it gets. For the examples, I would have liked some screenshots for the settings but I got all the help from you I needed in the end.* ". A sign test with α=0.05 and MD=3 showed a significant difference among these answers ($p\leq$0.0001****). An item analysis with a discrimination cutoff of c=0.27 that the question related to task 2 was the most difficult to answer, while the one for task 1 was the easiest one (difficulty=3.85).

NASA TLX Evaluation. Participants were also asked to fill out the NASA-Task Load Index (NASA-TLX) questionnaire [24, 25], which assesses the estimated workload of the *configuration* and *development* tasks. It consists of six subscales: mental demand, physical demand, temporal demand, effort, performance, and frustration level. In our case, there is no physical counterpart, thereby leaving aside the corresponding subscale.

Fig. 18 compares the averaged scores obtained for NASA-TLX sub-scales for the four tasks of the protocol. Regarding the mental workload, the first task was more demanding since the first task

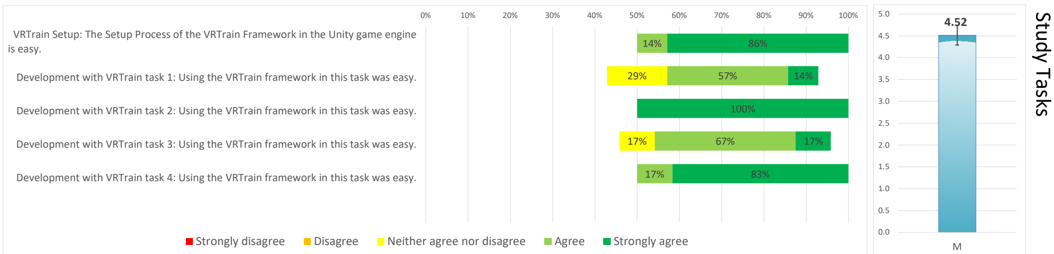


Fig. 17. Subjective ease of completing task with VRTrain framework.

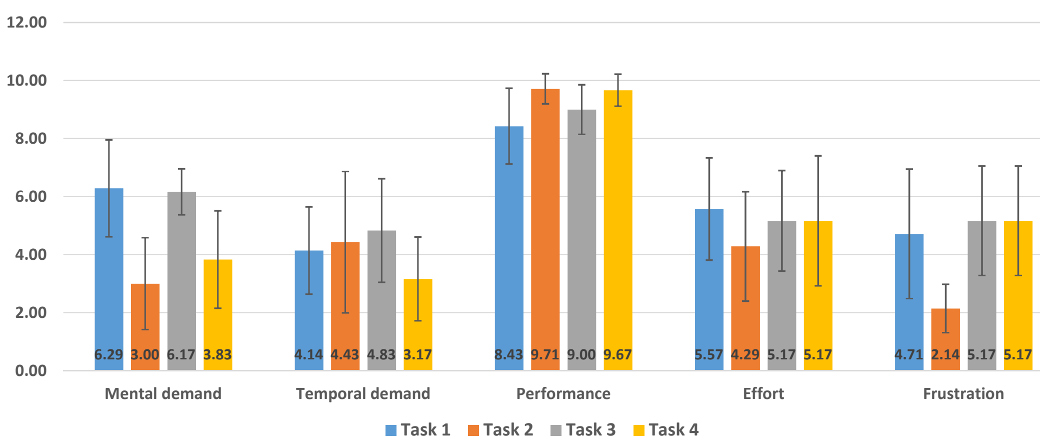


Fig. 18. Averaged scores for NASA-TLX subscales for all tasks. Error bars show a confidence interval of 95%.

always induces the entry cost, which decreases when repeated in the second task; the same pattern occurs with tasks 3 and 4. Regarding the temporal demand, the obtained scores are considered equivalent, thus suggesting that the time perceived by participants to complete the various steps are equivalent. Each participant managed to complete all steps of the experiment and some even reported being impressed by the results, considering that they would not have been able to do the same without the help of the framework. This could explain the high value of the "performance subfactor". Similarly to the mental demand, the perceived performance increases after its repetition, first from task 1 to task 2, then from task 3 to task 4. The perceived effort is again considered equivalent among tasks.

8 DISCUSSION

There is increasing interest and pilot implementations of VR technology in the professional context for skill and behavior training. However, for a technology that has largely developed in the entertainment and movie industry, there are still some challenges to its effective deployment and use in the professional context - which requires a broad range of expertise to design and successfully deliver appropriate training content and systems. In this paper, we proposed a development method for VR training applications, to enable the incorporation of diverse expertise in the development of VR training content through flexible authoring and personalization of the VR training simulations. An accompanying tool (the VRTrain framework) which is used by developers at the implementation step of the VR training development and helps enact this method was also presented. The VRTrain framework has been used in the development of three categories of VR research and training applications, and in this paper, we presented a use case in the development of a system for training teachers to deal with specific situations in a classroom. With a project in the context of the presented use case, and evaluation of the VRTrain framework was conducted with developers who were using the framework for the first time. We hereby discuss the results of the evaluation study in relation to the stated research questions:

RQ1: The results show that participants find it easy to setup and use the VRTrain framework (Section 7.2.2). However, users may encounter some difficulties with the first encounter using the framework in a programming task. One of the primary advantages of the VRTrain framework and method is to facilitate the creation of custom personalized training scenarios. Participants did not build any third party applications for authoring training scenarios in this study, but they were able to use a similar feature to simulate this task in Unity in study task 4. We find that this process was rated as second most easy task in the study, and users also report very high performance (success) in executing the task. While an evaluation of this step would be more meaningful with non-technical experts, this may hint that the process of creating personalized scenarios could be easy.

In study participants' comments about use of the framework, two themes about ease of use emerge: participants commented that it was easy to use the framework, and use of the framework enabled some implementation tasks to be completed faster. However, some participants also reported that the framework is only easy to use after one gets familiar with the basic principles of its use. The following user comments highlight this perception on ease of use of the VRTrain framework.

> *"It was a bit difficult to get used to it but once I understood everything (the documentation was very helpful and easy to understand) it was quite easy... It was way faster with the framework"*
>
> *"After completing the first task the use of the framework was way easier and it only took me a few minutes to finish the second task. It was way easier and faster with the framework."*

RQ2: Pre- and post-study acceptance evaluation results for attitude towards use, facilitating conditions, effort, and performance expectancy reveal that acceptance of developers is generally positive, with the acceptance increasing before first use of the framework and after using the framework. The results of our pre-study acceptability and post-study acceptance show that developers' effort expectancy when using the framework is significantly positive, which allows us to conclude that developers who have some experience developing real-time VR applications with Unity will be able to quickly and easily learn to use the system and become proficient at implementing applications with framework using the new development approach. The results show that effort expectancy significantly improved for the post-study after developers had used the framework in the study project, which confirms and re-enforces the positive pre-study effort expectancies of developers. The performance expectancy results showed that work improvement and time reduction were more critical for developers. Participants perceived more that "using the system would make it easier to do their job" after using the system, than before using the system. However, the average performance for pre-study and post-study did not differ significantly. Developers' attitude towards use slightly decreased from pre-study to post-study, but not significantly. A decrease does not mean a negative attitude towards use, because, in both pre-study and post-study, attitude towards use was highly positive (above median).

The following comments by participants highlight perceived acceptance of the framework:

> *"I think this framework would be a really good way to accomplish this task."*

> *"It was way easier and faster with the framework which was quite satisfying. ... after having completed the task i was very satisfied."*

These comments from participants highlight a more emotional acceptance of the framework, we discuss this in more context in the next section when we consider this in the context of how use of the framework affects and potentially changes the way participants accomplish their development tasks.

RQ3: User evaluation of enabling conditions in the pre-study and post-study was mixed, some questions scored below the median (Q4="Using the system is compatible with all aspects of my work" and Q6="Using the system fits into my work style"), while others scored above the median (Q1=" I have the resources necessary to use the system.", Q2=" I have the knowledge necessary to use the system.", and Q3="Given the resources, opportunities and knowledge it takes to use the system, it would be easy for me to use the System"). This suggests that the method of developing parametrized VR training applications is perceived as not fitting/compatible with user's style of work, but they have the necessary knowledge and tools to adopt the method. It is worth noting that participants in the study were all experienced at developing real-time applications with the game engine. Therefore, they already developed the habit of a certain work style which is different from the proposed parametrized method. This means that experienced developers have the required knowledge and skill but will have to learn the new method. This may be different for developers who are just starting to work with these tools since they will have to learn to use the new method at an early stage of their skill development. Considering the evaluation results of "effort expectancy" and "attitude towards use", experienced developers believe it is a good idea to use the system, but require a change of style for experienced developers.

Three main themes emerge from participants' comments on how use of the VRTrain framework affects the work of developers. First, participants perceive that less coding is required to complete subsequent VR training scenario setup tasks after the initial component scripts have been implemented. Secondly, use of the VRTrain framework enables greater code re-usability. Additionally, some architecture and design decisions are lifted enabling developers to focus on implementing the business logic of the simulation. Despite these, some participants still perceived some tasks as too monotonous and requiring exercise configuration and on-screen clicking on UI elements which is not typical of developers. The following comments emphasize these participant perceptions:

> *"Again, without the framework, I would just write a single script or extend a previously created one. I feel like this time less code is needed with the use of the framework than without it."*

> *"users dont have to build and design everything (system architectures, detail design, workflow, etc.) from the ground"*

> *"certainly the code is more reusable when following this approach, and easier to edit too."*

> *"It was really easy to accomplish this task and it requires no coding, but a lot of clicking in the editor. Without the framework, it would probably take longer and would be somewhat hard-coded."*

While the VRTrain method provides method definitions for the entire development cycle, the VRTrain framework is only used at the implementation phase of the VR training application in a real-time game engine, but not for the implementation of a training administration application. It is also worth noting that the evaluation presented in this paper does not directly evaluate the systematic development method, but only indirectly through the use of the framework.

In this paper, we proposed a method to design small atomic behaviors, tasks, and data recorder components that can be used by training administrators to build personalized training scenarios based on training needs. However, this raises concern about how small these atomic components should be; more atomic means greater flexibility but more effort and time to set up training scenarios, and larger behavior and task components would mean less flexibility and less effort and time to set up training scenarios. This is already observed in our evaluation study – in the execution of the study task *"Development with VRTrain task 4: Create script scenario."*; while 83% (Fig. 17) of participants strongly agree that completing the scripted scenario with the VRTrain framework was easy, a participant's comment on the task also suggested that the task involved too many mechanical operations and data input. To address this situation, small atomic components can be configured and grouped to form larger components that can be re-used between projects. Secondly, as presented in Section 6.4, the VRTrain framework is designed such that VRTrain events can be programmatically created at runtime by an artificial intelligence (AI) agent, this approach is also proposed by Coltey et al. [13]. However, use of an AI agent means that application domain and training experts are not directly involved in the process of creating training content. The effect of one or a combination of both approaches should be the subject of further research.

Another important, but also challenging and often ignored, aspect of VR training and VR systems, in general, is the ethical concerns [33]. Given that the chosen application scenario for this study trains teachers (the users) to develop new habits on how to interact with other persons (students), these skills could be transferred into actual interactions in the real world. While this study does not seek to investigate the effectiveness of this skill transfer, it is important to note that any amount of skill transfer that was learned in the virtual environment could lead to unintended consequences and raise ethical concerns. VRTrain allows educational experts to author training scenarios after the system is implemented. Therefore, it is important to ensure due diligence and adherence to ethical values at three stages: during the design and implementation of the configurable behaviors and tasks, during training scenario authoring by the educational expert, and during the operation of the VR training - to ensure that users can safely use the VR training system, act responsibly, and adhere to ethical values in the virtual environment.

Limitations (Threats to Validity).
In this section, we discuss some of the limitations of the evaluation study, and the VRTrain method and framework in its current state.

(1) The VRTrain Method does not provide tools for automating all phases of the development process (*e.g.*, Design, Iterative Design, Verification, etc.). Traditional software engineering tools can be used at these stages. However, it might be helpful to provide some specific automation tools, *e.g.*, for automatically transforming design specifications into software specifications for implementation with VRTrain. This is not currently implemented or provided in the current state of the work.
(2) The evaluation study results could be affected by individual developer experience with tools such as git, and the performance of individual developer end-devices used in the study. Additionally, participants were only allowed to begin with the study after watching the video and completing the pre-study evaluation. However, there were no additional parameters to decide when participants should start the study. It is possible that there could be variations in participants' understanding of the video which could introduce bias in the study.
(3) The evaluation study presented in this paper only evaluates the use of the VRTrain framework and the implementation phase of the method. All stakeholders were not included in the evaluation. A comprehensive evaluation of the method will require evaluation in the context of a complete project with all experts and stakeholders, particularly training experts.
(4) The introduction video for the pre-study evaluation (which eventually also served as training prior to starting the study) could be uncomfortably too long. It is likely that some participants

could become weary before the end of the video, which could affect their responses in the pre-study.

(5) Given the constraints at the time of the study and the wide range of tasks that were executed by participants, it was only possible to conduct the study remotely and allow participants to complete the tasks over a variable length of time without drastically disrupting their regular work schedule. This variability could also introduce inconsistencies in the evaluation.

9 CONCLUSION

In this paper, we discussed the use of virtual reality technologies in the professional context for skill and behavior training. We also discussed the state of the art and challenges to developing virtual reality training applications, primarily the need to define special development methodologies that incorporate the vast knowledge of creating training systems, and to personalize training to meet specific training needs. To address these issues, we proposed a systematic method for developing virtual reality training applications, and a framework tool to support developers of these systems enacting the method. We presented a use case of the method and an evaluation of the method with experienced developers. The results of the evaluation showed that developers are generally positive towards use of the method, they have the necessary knowledge and tools to use the method, and find it easy to learn and use in the development process. However, experienced developers who already have specific styles of work do not perceive the new method as fitting or compatible with their style of work. Therefore they will need to learn to develop applications with the new method.

The evaluation presented was conducted with experienced developers and did not directly evaluate the systematic development method since a complete team of diverse experts is required to evaluate the use of the method in a team. Developers also expressed some technical concerns when performing manual configuration tasks. Therefore, future research work should focus on two main areas: conducting larger studies with teams in the industry on the use of the method and framework, and the effect of developer experience should also be studied with experienced and non-experienced study participants. Secondly, future technical work should explore methods of automating some manual tasks with the use of artificial intelligence methods, and also evaluate the impact of these approaches. The main contributions of the research work presented in this paper can be classified into three categories: conceptual, methodological, and software tool.

(1) A concept for developing VR applications for training consisting of a systematic development method and a framework tool. The concept promotes using state-of-the-art tools, while adhering to standards, but also managing the complexities of the application domain.
(2) A systematic method with standard method definitions describing: roles, tasks, and work products, for implementing training-oriented virtual reality applications. The systematic method promotes the integration of pedagogic and domain expert requirements and knowledge in the development of VR training systems. The systematic development method includes method definitions for the design and implementation of the application with high interoperability. Interoperability subsequently enables the use of external third-party applications for intuitive authoring and personalization of VR training to meet training needs, without technical skills.
(3) A development framework/tool which enacts the proposed method, and can be used by developers at the implementation phase of VR training application development. We present our approach to developing the tool, a use case, and also discuss the results of an evaluation study, which opens avenues for further development, research, and discussion on the subject of VR training application development.

REFERENCES

[1] Saeed Aghaee, Alan F. Blackwell, David Stillwell, and Michal Kosinski. 2015. Personality and intrinsic motivational factors in end-user programming. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2015, Atlanta, GA, USA, October 18-22, 2015*, Zhen Li, Claudia Ermel, and Scott D. Fleming (Eds.). IEEE Computer Society, 29–36. https://doi.org/10.1109/VLHCC.2015.7357195

[2] Christoph Anthes, Rubén Jesús García-Hernández, Markus Wiedemann, and Dieter Kranzlmüller. 2016. State of the art of virtual reality technology. In *Aerospace Conference, 2016 IEEE*. IEEE, 1–19. https://doi.org/10.1109/AERO.2016.7500674

[3] Sooraj K. Babu, Sooraj Krishna, Unnikrishnan R, and Rao R. Bhavani. 2018. Virtual Reality Learning Environments for Vocational Education: A Comparison Study with Conventional Instructional Media on Knowledge Retention. In *18th IEEE International Conference on Advanced Learning Technologies, ICALT 2018, Mumbai, India, July 9-13, 2018*, Maiga Chang, Nian-Shing Chen, Ronghuai Huang, Kinshuk, Kannan M. Moudgalya, Sahana Murthy, and Demetrios G. Sampson (Eds.). IEEE Computer Society, 385–389. https://doi.org/10.1109/ICALT.2018.00094

[4] Shannon KT Bailey, Cheryl I Johnson, Bradford L Schroeder, and Matthew D Marraffino. 2017. Using virtual reality for training maintenance procedures. In *Proceedings of the Interservice/Industry Training, Simulation and Education Conference*.

[5] Mukesh Barange. 2015. Task-Oriented Communicative Capabilities of Agents in Collaborative Virtual Environments for Training. (Des agents avec des capacités communicatives orientées tâche dans les Environnements de réalité Virtuelle Collaboratifs pour l'Apprentissage). https://tel.archives-ouvertes.fr/tel-01232885

[6] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software* 149 (2019), 101–137. https://doi.org/10.1016/j.jss.2018.11.041

[7] Olavo Da Rosa Belloc, Rodrigo B. D. Ferraz, Marcio Calixto Cabral, Roseli De Deus Lopes, and Marcelo Knörich Zuffo. 2012. Virtual Reality Procedure Training Simulators in X3D. In *Proceedings of the 17th International Conference on 3D Web Technology* (Los Angeles, California) *(Web3D '12)*. Association for Computing Machinery, New York, NY, USA, 153–160. https://doi.org/10.1145/2338714.2338741

[8] Leif P. Berg and Judy M. Vance. 2017. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Real.* 21, 1 (2017), 1–17. https://doi.org/10.1007/s10055-016-0293-9

[9] Cédric Buche, Ronan Querrec, Pierre De Loor, and Pierre Chevaillier. 2004. MASCARET: A Pedagogical Multi-Agent System for Virtual Environments for Training. *Int. J. Distance Educ. Technol.* 2, 4 (2004), 41–61. https://doi.org/10.4018/jdet.2004100103

[10] Kevin Carpentier and Domitile Lourdeaux. 2013. Generation of Learning Situations According to the Learner's Profile Within a Virtual Environment. In *Agents and Artificial Intelligence - 5th International Conference, ICAART 2013, Barcelona, Spain, February 15-18, 2013. Revised Selected Papers (Communications in Computer and Information Science, Vol. 449)*, Joaquim Filipe and Ana L. N. Fred (Eds.). Springer, 245–260. https://doi.org/10.1007/978-3-662-44440-5_15

[11] Chwen Jen Chen, Seong Chong Toh, and Wan Mohd Fauzy Wan Ismail. 2005. Are Learning Styles Relevant To Virtual Reality? *Journal of Research on Technology in Education* 38, 2 (2005), 123–141. https://doi.org/10.1080/15391523.2005.10782453

[12] Pierre Chevaillier, Thanh-Hai Trinh, Mukesh Barange, Pierre De Loor, Frédéric Devillers, Julien Soler, and Ronan Querrec. 2012. Semantic modeling of Virtual Environments using MASCARET. In *5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems, SEARIS 2012, Costa Mesa, CA, USA, March 5, 2012*. IEEE Computer Society, 1–8. https://doi.org/10.1109/SEARIS.2012.6231174

[13] Erik Coltey, Yudong Tao, Tianyi Wang, Shahin Vassigh, Shu-Ching Chen, and Mei-Ling Shyu. 2021. Generalized Structure for Adaptable Immersive Learning Environments. In *22nd IEEE International Conference on Information Reuse and Integration for Data Science, IRI 2021, Las Vegas, NV, USA, August 10-12, 2021*. IEEE, 294–301. https://doi.org/10.1109/IRI51335.2021.00047

[14] D Dorsey, G Campbell, and S Russell. 2009. Adopting the instructional science paradigm to encompass training in virtual environments. *Theoretical Issues in Ergonomics Science* 10, 3 (2009), 197–215. https://doi.org/10.1080/14639220802151427

[15] Richard M. Eastgate, John R. Wilson, and Mirabelle D'Cruz. 2014. Structured Development of Virtual Environments. , 353–389 pages. https://doi.org/10.1201/b17360-20

[16] Christiane Eichenberg. 2012. *Virtual reality in psychological, medical and pedagogical applications.* BoD–Books on Demand.

[17] Clive Fencott. 2005. A methodology of design for virtual environments. In *Developing Future Interactive Systems*. IGI Global, 66–91. https://doi.org/10.4018/978-1-59140-411-8.ch003

[18] Pablo Figueroa, Mark Green, and H James Hoover. 2002. InTml: A Description Language for VR Applications. In *Proceedings of the Seventh International Conference on 3D Web Technology (Web3D '02)*. Association for Computing Machinery, New York, NY, USA, 53–58. https://doi.org/10.1145/504502.504511

[19] Jennifer E. Fowlkes, Kelly J. Neville, Jerry M. Owens, and Amanda J. Hafich. 2009. Challenges to the development of pedagogically driven engineering requirements for complex training systems. *Theoretical Issues in Ergonomics Science* 10, 3 (2009), 217–229. https://doi.org/10.1080/14639220802151575

[20] Joakim Grant Frederiksen, Stine Maya Dreier Sørensen, Lars Konge, Morten Bo Søndergaard Svendsen, Morten Nobel-Jørgensen, Flemming Bjerrum, and Steven Arild Wuyts Andersen. 2019. Cognitive load and performance in immersive virtual reality versus conventional virtual reality simulation training of laparoscopic surgery: a randomized trial. *Surgical endoscopy* (2019), 1–9. https://doi.org/10.1007/s00464-019-06887-8

[21] Małgorzata Gawlik-Kobylińska, Paweł Maciejewski, Jacek Lebiedź, and Aneta Wysokińska-Senkus. 2020. Factors Affecting the Effectiveness of Military Training in Virtual Reality Environment. In *Proceedings of the 2020 9th International Conference on Educational and Information Technology* (Oxford, United Kingdom) *(ICEIT 2020)*. Association for Computing Machinery, New York, NY, USA, 144–148. https://doi.org/10.1145/3383923.3383950

[22] Samuel Greengard. 2019. *Virtual reality*. Mit Press.

[23] David J. Harris, Mark R. Wilson, and Samuel J. Vine. 2020. Development and validation of a simulation workload measure: the simulation task load index (SIM-TLX). *Virtual Real.* 24, 4 (2020), 557–566. https://doi.org/10.1007/s10055-019-00422-9

[24] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (2006), 904–908. https://doi.org/10.1177/154193120605000909

[25] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[26] John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research* 77, 1 (2007), 81–112. https://doi.org/10.3102/003465430298487

[27] Quinate Chioma Ihemedu-Steinke, Rainer Erbach, Prashanth Halady, Gerrit Meixner, and Michael Weber. 2017. Virtual Reality Driving Simulator Based on Head-Mounted Displays. In *Automotive User Interfaces: Creating Interactive Experiences in the Car*, Gerrit Meixner and Christian Müller (Eds.). Springer International Publishing, Cham, 401–428. https://doi.org/10.1007/978-3-319-49448-7_15

[28] Jason Jerald. 2015. Content Creation: Design Guidelines, Chapter 24. In *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan & Claypool, 267–274. https://doi.org/10.1145/2792790.2792819

[29] Jason Jerald. 2015. Content Creation, Part IV. In *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan & Claypool, 223–274. https://doi.org/10.1145/2792790.2792814

[30] Jason Jerald. 2015. *Environmental Design*. Association for Computing Machinery and Morgan & Claypool. https://doi.org/10.1145/2792790.2792816

[31] Jason Jerald. 2015. *A History of VR, Chapter 2*. Association for Computing Machinery and Morgan & Claypool, 15–28. https://doi.org/10.1145/2792790.2792794

[32] Omur Kaya Kalkan, Sener Karabulut, and Gurhan Hoke. 2021. Effect of Virtual Reality-Based Training on Complex Industrial Assembly Task Performance. *Arabian Journal for Science and Engineering* 46, 12 (2021), 12697–12708. https://doi.org/10.1007/s13369-021-06138-w

[33] Ben Kenwright. 2018. Virtual Reality: Ethical Challenges and Dangers [Opinion]. *IEEE Technology and Society Magazine* 37, 4 (2018), 20–25. https://doi.org/10.1109/MTS.2018.2876104

[34] Vix Kemanji Ketoma, Rene Mpwadina, and Gerrit Meixner. 2022. Virtual Reality Assembly of Physical Parts: The Impact of Interaction Interface Techniques on Usability and Performance. In *Virtual, Augmented and Mixed Reality: Applications in Education, Aviation and Industry - 14th International Conference, VAMR 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26 - July 1, 2022, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 13318)*, Jessie Y. C. Chen and Gino Fragomeni (Eds.). Springer, 350–368. https://doi.org/10.1007/978-3-031-06015-1_24

[35] G Jounghyun Kim, Kyo Chul Kang, Hyejung Kim, and Jiyoung Lee. 1998. Software Engineering of Virtual Worlds. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '98)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/293701.293718

[36] Amy J. Ko, Thomas D. LaToza, and Margaret M. Burnett. 2015. A practical guide to controlled experiments of software engineering tools with human participants. *Empir. Softw. Eng.* 20, 1 (2015), 110–141. https://doi.org/10.1007/s10664-013-9279-3

[37] Marius Koller, Philip Schäfer, Magdalena Sich, Julia Diemer, Mathias Müller, and Gerrit Meixner. 2018. Next Generation Virtual Reality Exposure Therapy Systems - A Study exploring Design Implications. In *9th IEEE International Conference on Intelligent Systems, IS 2018, Funchal, Madeira, Portugal, September 25-27, 2018*, Ricardo Jardim-Gonçalves, João Pedro Mendonça, Vladimir Jotsov, Maria Marques, João Martins, and Robert E. Bierwolf (Eds.). IEEE, 528–535. https://doi.org/10.1109/IS.2018.8710504

[38] Marius Koller, Philip Schäfer, Magdalena Sich, Julia Diemer, Mathias Müller, and Gerrit Meixner. 2018. Next Generation Virtual Reality Exposure Therapy Systems - A Study exploring Design Implications. In *2018 International Conference on Intelligent Systems (IS)*. 528–535. https://doi.org/10.1109/IS.2018.8710504

[39] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 22, 140 (1932), 55–. http://psycnet.apa.org/record/1933-01885-001

[40] Oussema Mahdi, Lahcen Oubahssi, Claudine Piau-Toffolon, and Sébastien Iksal. 2018. Towards Design and Operationalization of Pedagogical Situations in the VRLEs. In *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*. 400–402. https://doi.org/10.1109/ICALT.2018.00095

[41] Oussema Mahdi, Lahcen Oubahssi, Claudine Piau-Toffolon, and Sébastien Iksal. 2019. Assistance to Scenarisation of VR-Oriented Pedagogical Activities: Models and Tools. In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, Vol. 2161-377X. 344–346. https://doi.org/10.1109/ICALT.2019.00107

[42] Oussema Mahdi, Lahcen Oubahssi, Claudine Piau-Toffolon, and Sébastien Iksal. 2019. A Model and Its Tool to Assist the Scenarization of VR-oriented Pedagogical Activities. In *Proceedings of the 11th International Conference on Computer Supported Education, CSEDU 2019, Heraklion, Crete, Greece, May 2-4, 2019, Volume 2*, H. Chad Lane, Susan Zvacek, and James Uhomoibhi (Eds.). SciTePress, 511–518. https://doi.org/10.5220/0007761905110518

[43] Guido Makransky, Thomas S. Terkildsen, and Richard E. Mayer. 2019. Adding immersive virtual reality to a science lab simulation causes more presence but less learning. *Learning and Instruction* 60 (2019), 225–236. https://doi.org/10.1016/j.learninstruc.2017.12.007

[44] F Mattioli, D Caetano, A Cardoso, and E Lamounier. 2015. On the agile development of virtual reality systems. In *Proceedings of the international conference on software engineering research and practice (SERP)*. The Steering Committee of The World Congress in Computer Science, Computer . . . , 10. https://www.semanticscholar.org/paper/On-the-Agile-Development-of-Virtual-Reality-Systems-Mattioli-Caetano/522afc363159287d4185af01e1d566321281ff35

[45] Tomasz Mazuryk and Michael Gervautz. 1996. Virtual reality-history, applications, technology and future. (1996). https://www.semanticscholar.org/paper/Virtual-Reality-History-%2C-Applications-%2C-Technology-Mazuryk/6cebef400df6c387852b6c76570f226a86965104

[46] Punya Mishra and Matthew J Koehler. 2006. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record* 108, 6 (2006), 1017–1054. https://doi.org/10.1111/j.1467-9620.2006.0068

[47] José P. Molina, Arturo S. García, Diego Martinez, Francisco J. Manjavacas, Victor Blasco, Victor López, and Pascual González. 2006. The Development of Glove-Based Interfaces with the TRES-D Methodology. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Limassol, Cyprus) *(VRST '06)*. Association for Computing Machinery, New York, NY, USA, 216–219. https://doi.org/10.1145/1180495.1180539

[48] Vivian Genaro Motti and Jean Vanderdonckt. 2013. A computational framework for context-aware adaptation of user interfaces. In *IEEE 7th International Conference on Research Challenges in Information Science, RCIS 2013, Paris, France, May 29-31, 2013*, Roel J. Wieringa, Selmin Nurcan, Colette Rolland, and Jean-Louis Cavarero (Eds.). IEEE, 1–12. https://doi.org/10.1109/RCIS.2013.6577709

[49] Object Management Group OMG. 2008. Software and Systems Process Engineering Metamodel Specification Version 2.0. https://www.omg.org/spec/SPEM/2.0/About-SPEM/

[50] Lahcen Oubahssi, Oussema Mahdi, Claudine Piau-Toffolon, and Sébastien Iksal. 2018. A process of design and production of Virtual Reality Learning Environments, In International Conference on Interactive Collaborative Learning. *The Challenges of the Digital Transformation in Education*, 353–364. https://doi.org/10.1007/978-3-030-11932-4_34

[51] Jarrell Pair, Brian Allen, Matthieu Dautricourt, Anton Treskunov, Matt Liewer, Ken Graap, and Greg Reger. 2006. A Virtual Reality Exposure Therapy Application for Iraq War Post Traumatic Stress Disorder. In *IEEE Virtual Reality Conference, VR 2006, Alexandria, Virginia, USA, March 25-29, 2006*. IEEE Computer Society, 67–72. https://doi.org/10.1109/VR.2006.23

[52] Jay Patel. 2015. Virtual Reality Device Developments. *XRDS* 22, 1 (nov 2015), 13. https://doi.org/10.1145/2837760

[53] Sebastian Felix Rauh, Marius Koller, Philip Schäfer, Gerrit Meixner, Cristian Bogdan, and Olga Viberg. 2021. MR On-SeT: A Mixed Reality Occupational Health and Safety Training for World-Wide Distribution. *Int. J. Emerg. Technol. Learn.* 16, 5 (2021). https://doi.org/10.3991/ijet.v16i05.19661

[54] Barbara Olasov Rothbaum, Matthew Price, Tanja Jovanovic, Seth D. Norrholm, Maryrose Gerardi, Boadie Dunlop, Michael Davis, Bekh Bradley, Erica J. Duncan, Albert Rizzo, and Kerry J. Ressler. 2014. A Randomized, Double-Blind Evaluation of d-Cycloserine or Alprazolam Combined With Virtual Reality Exposure Therapy for Posttraumatic Stress Disorder in Iraq and Afghanistan War Veterans. *American Journal of Psychiatry* 171, 6 (2014), 640–648. https://doi.org/10.1176/appi.ajp.2014.13121625 PMID: 24743802.

[55] Marilyn C. Salzman, Chris Dede, R. Bowen Loftin, and Jim Chen. 1999. A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning. *Presence: Teleoperators and Virtual Environments* 8, 3 (06 1999), 293–316. https://doi.org/10.1162/105474699566242

[56] Maria-Isabel Sánchez-Segura, Angélica de Antonio, and Antonio de Amescua. 2005. SENDA: A whole process to develop virtual environments. In *Developing future interactive systems*. IGI Global, 92–115. https://doi.org/10.4018/978-1-59140-411-8.ch004

[57] Ugo Braga Sangiorgi, François Beuvens, and Jean Vanderdonckt. 2012. User Interface Design by Collaborative Sketching. In *Proceedings of the Designing Interactive Systems Conference* (Newcastle Upon Tyne, United Kingdom) *(DIS '12)*. Association for Computing Machinery, New York, NY, USA, 378–387. https://doi.org/10.1145/2317956.2318013

[58] Julien Saunier, Mukesh Barange, Bernard Blandin, and Ronan Querrec. 2016. A methodology for the design of pedagogically adaptable learning environments. *Int. J. Virtual Real.* 16, 1 (2016), 15–21. https://doi.org/10.20870/IJVR.2016.16.1.2878

[59] Julien Saunier, Mukesh Barange, Bernard Blandin, Ronan Querrec, and Joanna Taoum. 2016. Designing Adaptable Virtual Reality Learning Environments. In *Proceedings of the 2016 Virtual Reality International Conference* (Laval, France) *(VRIC '16)*. Association for Computing Machinery, New York, NY, USA, Article 5, 4 pages. https://doi.org/10.1145/2927929.2927937

[60] Neal E Seymour, Anthony G Gallagher, Sanziana A Roman, Michael K O'brien, Vipin K Bansal, Dana K Andersen, and Richard M Satava. 2002. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery* 236, 4 (2002), 458–464. https://doi.org/10.1097/00000658-200210000-00008

[61] Traci Sitzmann. 2011. A meta-analytic examination of the instructional effectiveness of computer-based simulation games. *Personnel psychology* 64, 2 (2011), 489–528. https://doi.org/10.1111/j.1744-6570.2011.01190.x

[62] Mel Slater, Vasilis Linakis, Martin Usoh, Rob Kooper, and Gower Street. 1996. Immersion, presence and performance in virtual environments: an experiment with tri-dimensional chess. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST 1996, Hong Kong, July 01-04, 1996*, Mark Green, Kim M. Fairchild, and Michael Zyda (Eds.). ACM, 163–172. https://doi.org/10.1145/3304181.3304216

[63] José Luis Soler, Janaina Ferreira, Manuel Contero, and Mariano Alcañiz. 2017. The power of sight: using eye tracking to assess learning experience (LX) in virtual reality environments. *INTED2017 proceedings* (6-8 March, 2017 2017), 8684–8689. https://doi.org/10.21125/inted.2017.2060

[64] Mark A. Tarlton and P. Nong Tarlton. 1992. A Framework for Dynamic Visual Applications. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics, SI3D '92, Cambridge, MA, USA, March 29 - April 1, 1992*, Marc Levoy, Edwin E. Catmull, and David Zeltzer (Eds.). ACM, 161–164. https://doi.org/10.1145/147156.147189

[65] Tomas Trescak, Marc Esteva, and Inmaculada Rodríguez. 2010. A Virtual World Grammar for automatic generation of virtual worlds. *Vis. Comput.* 26, 6-8 (2010), 521–531. https://doi.org/10.1007/s00371-010-0473-7

[66] Lode Vanacken, Chris Raymaekers, and Karin Coninx. 2007. Introducing Semantic Information during Conceptual Modelling of Interaction for Virtual Environments. In *Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction* (Nagoya, Japan) *(WMISI '07)*. Association for Computing Machinery, New York, NY, USA, 17–24.

[67] Madis Vasser, Markus Kängsepp, Murad Magomedkerimov, Kälver Kilvits, Vladislav Stafinjak, Taavi Kivisik, Raul Vicente, and Jaan Aru. 2017. VREX: an open-source toolbox for creating 3D virtual reality experiments. *BMC psychology* 5, 1 (2017), 4. https://doi.org/10.1186/s40359-017-0173-4

[68] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. 2003. User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly* 27, 3 (feb 2003), 425–478. https://doi.org/10.2307/30036540

[69] John A Waterworth and E Waterworth. 2003. The core of presence: Presence as perceptual illusion. *Presence connect* 3, 3 (2003).

Received July 2022; revised October 2022; accepted December 2022