Communication Papers of the 18th Conference on Computer Science and Intelligence Systems pp. 265–272
DOI: 10.15439/2023F1379
ISSN 2300-5963 ACSIS, Vol. 37

Knowledge-Based Creation of Industrial VR Training Scenarios

Paweł Sobociński, Jakub Flotyński, Michał Śliwicki, Mikołaj Maik, Krzysztof Walczak
Department of Information Technology, Poznań University of Economics and Business
Email: [pawel.sobocinski, jakub.flotynski, michal.sliwicki, mikolaj.maik, krzysztof.walczak]@ue.poznan.pl

***Abstract*—The application of virtual reality (VR) for building training systems has grown in popularity across diverse fields. This trend is particularly prevalent in Industry 4.0, where many real-world training scenarios can be expensive or pose potential dangers to trainees. The most important aspect of professional training is domain-specific knowledge, which can be expressed using the semantic web approach. This approach facilitates complex queries and reasoning against the representation of training scenarios, which can be useful for educational purposes. However, current methods and tools for creating VR training systems do not utilize semantic knowledge representation, making it difficult for domain experts without IT expertise to create, modify, and manage training scenarios. To address this issue, we propose an ontology-based representation and a method of modeling VR training scenarios. We demonstrate our approach by modeling VR training scenarios for Industry 4.0 in the field of the production of household equipment. The domain knowledge used represents training activities, potential errors, and equipment failures in a way comprehensible to domain experts.**

I. Introduction

COMPARED to traditional training methods, VR training systems offer significant advantages. First, training in VR is more engaging and attractive to users compared to paper, audio, or video materials. Second, virtual training eliminates the need for physical infrastructure or dangerous equipment, reducing the risk posed to users. Moreover, it liberates companies from acquiring expensive or unavailable devices, especially in Industry 4.0 environments where production devices cannot be suspended. Finally, VR training can be carried out to a certain degree without the need for instructors. This makes it simpler to organize, more cost-effective, more efficient, and more flexible compared to conventional training methods.

However, creating effective VR training environments with behavior-rich scenes and objects requires expertise in programming and 3D modeling, as well as domain knowledge to prepare practical and meaningful training scenarios in a specific domain. As a result, the development process often involves collaboration between IT specialists and domain experts, who typically have limited knowledge of IT. This collaboration can make the development of VR training environments complex, time-consuming, and costly. Therefore, the availability of user-friendly tools for domain experts to design VR training with domain knowledge is crucial in reducing the required time and effort and promoting the use of VR in training.

The semantic web is a leading method for representing domain knowledge, providing a range of standards for conveying content in a manner understandable to humans and processable by software. Ontologies are the primary form of content representation in the semantic web, formulated using the Resource Description Framework (RDF), the Resource Description Framework Schema (RDFS), and the Web Ontology Language (OWL). RDF establishes a data model, whereas RDFS and OWL expand RDF terminology allowing to build ontologies. The semantic web standards rely on description logic, which enables the representation of concepts, roles, and individuals. Such representations can be subject to reasoning, leading to the inference of implicit knowledge based on explicit knowledge and precise queries, including highly complex conditions. This is highly beneficial for content creation and management by users across various domains.

To date, the semantic web has primarily been used for the representation of 3D content, including its geometry, structure, and presentation, which is insufficient for managing complex VR training, with its users, tasks, and equipment, as well as possible problems and errors. User-friendly tools are needed for domain experts to design VR training with domain knowledge, making the development of training environments less complicated, less time-consuming, and more cost-effective.

In this paper, we present a new method for creating VR training scenarios that utilize the semantic web. Our approach includes two primary components: an ontology-based representation of domain knowledge in training scenarios and a user-friendly semantic scenario editor. The ontology-based representation covers various elements such as users, tasks, equipment, and potential problems or errors that may arise during training scenarios. Using the semantic scenario editor, domain experts can easily design scenarios through an intuitive visual interface. This method allows for domain-specific descriptions of training scenarios and scenes using well-known semantic web standards. Furthermore, the process of selecting and combining appropriate objects for training scenarios, as well as verifying modeling results, can be completed using well-recognized activities on description logics such as instance checking, query answering, and consistency checking against the used ontologies.

The project discussed in this paper focuses on developing a VR training system for the production of household appliances. Hence, all examples and discussions are centered on this application domain. However, the proposed approach can be adapted for other domains if the relevant objects and actions are identified and semantically described.

The remainder of the paper is structured as follows: Section II provides an overview of the current state of VR training environments and existing approaches to semantic modeling of VR content. Section III outlines the proposed approach, while Section IV explains the ontology-based representation of training scenarios. The semantic scenario editor, which utilizes this representation, is discussed in Section V. Section VI presents an example of VR training. Section VI-F provides a discussion of the results. Finally, Section VII concludes the paper and indicates possible future research.

II. Related work

Up until now, little attention has been given to the utilization of ontologies in virtual reality training and education by the research community. A study by [1] proposed ontologies for creating VR training at various levels of abstraction, including high, medium, and low. The high-level ontology defines entities that represent physical and non-physical objects that may occur, such as avatars, tools, vehicles, roles, animals, and events that could happen in VR environments. The medium-level ontology builds on the high-level ontology by providing a classification of avatars, tools, vehicles, roles, and animals with more concrete entities. The low-level ontology describes entities that are specific to a particular VR environment.

A medical diagnosis system has been described in [2]. It leverages an ontology-based approach to represent medical knowledge, where separate ontologies are utilized to illustrate patients' physical and mental states. An avatar, which communicates with patients through voice, employs these ontologies. To make diagnoses, the system employs probabilistic reasoning with a Bayesian network.

Numerous studies have focused on representing 3D content through ontology-based approaches, which involve a range of geometrical, structural, spatial, and presentational elements. An extensive evaluation of these methods has been provided in [3], and a summary of the existing techniques can be found in Table I. Among the methods, four aim to address low-level abstraction that is specific to graphics, while six approaches support high-level abstraction that is either general or specific to a domain. Furthermore, three of these methods can be employed with different domain ontologies.

TABLE I: Comparison of semantic 3D content modeling methods

Approach	Level of Abstraction	
	Low (3D graphics)	High (application domain)
De Troyer et al. [4]	✓	general
Gutiérrez et al. [5]	✓	humanoids
Kalogerakis et al. [6]	✓	-
Spagnuolo et al. [7]	-	humanoids
Floriani et al. [8]	✓	-
Kapahnke et al. [9]	-	general
Albrecht et al. [10]	-	interior design
Latoschik et al. [11]	-	general
Drap et al. [12]	-	archaeology
Trellet et al. [13]	-	molecules
Perez-Gallardo et al. [14]	✓	-

Another example of a knowledge-based 3D design method has been described in [15]. The paper presents a collaborative method for the interactive development of aircraft cabin systems in VR based on preliminary design data. The knowledge is stored in an ontology which is linked with design rules and external parameters, which can generate missing information needed for the design of cabin systems. The design rules are based on requirements, safety regulations as well as expert knowledge for design interpretation that has been collected and formalized. The data is stored in an XML file that can be used to generate a 3D virtual cabin mockup in which users have the possibility to interact with cabin modules and system components via controllers. This VR model enables interaction with complex product data sets by visualizing metadata and analysis results along with the cabin geometry, making it even better comprehensible and processable for humans. It allows the design to be evaluated and optimized at a low cost before the concepts are validated in a real prototype.

Another example of ontologies for VR is the OntoPhaco project presented in [16]. The goal of the OntoPhaco project was to develop a new approach to the evaluation and design of ontologies in ophthalmology, specifically for cataract surgery training. The authors propose a solution on how to design a proper domain model to support VR training in ophthalmology, which includes the OntoPhaco ontology, built using OntoUML based on UML. They also introduce systematic verification and validation processes that include theoretical and hypothetical evaluation of the system and the use of feedback from domain experts to verify and revise the ontology for VR training. The conducted evaluation shows, that OntoPhaco has the potential to improve the learning experience of students and facilitate the development of VR training in the future.

There is also an example of ontology-based, general-purpose and Industry 4.0-ready architecture to use with systems supporting factory workers that use mixed reality [17]. In the paper, authors describe a general ontology, that is capable of structuring knowledge to enable interoperability and standardization between such systems. The approach enables data findability and reusability. The proposed architecture was implemented and validated in two case studies in the manufacturing sector: scheduled maintenance and alarm management, and customer order management.

III. Overview of the Approach

The review presented in Section II shows that universal, cross-domain methods and tools for creating interactive VR training scenarios are still missing. The existing ontologies for VR are limited to either 3D-specific features that focus on the properties of static 3D content or domain-specific features that focus on a single application domain. There is a lack of domain-independent conceptualization of actions and interactions, which can be utilized by non-technical users to create VR environments with minimal assistance from programmers and graphics designers. Solutions that focus on 3D content behavior, like [18], are broad in scope and do not provide the specific concepts and roles required for training scenarios.

The main contribution of this paper is a solution to the problem mentioned above: an approach to semantic representation and modeling of VR training scenarios. The approach is illustrated in Fig. 1 and consists of two key elements: the *ontology-based representation of training scenarios* and the *semantic scenario editor*. The *ontology-based representation* comprises resources based on semantic web standards such as RDF, RDFS, and OWL, which cover training scenarios, scenes, and objects in terms of both their semantics and visualization. The central component of the representation is the *scenario ontology*, consisting of a TBox and an RBox, which includes concepts (classes) and roles (properties) associated with training scenarios, scenes, infrastructure objects, and equipment. Since the classes and properties are general, the ontology can be utilized in various application domains.

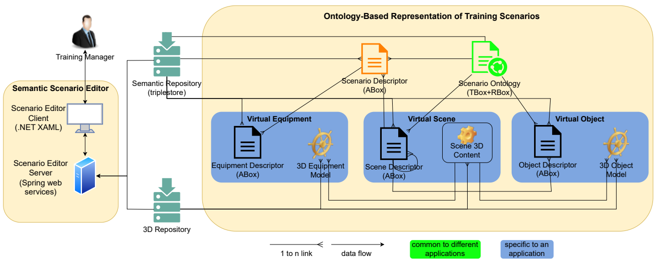

Fig. 1: Overview of knowledge-based representation and modeling of VR training scenarios.

Based on the scenario ontology, four kinds of *descriptors* are created: *scenario descriptors*, *scene descriptors*, *object descriptors*, and *equipment descriptors*. Each descriptor is an ABox that represents individuals linked to a specific scenario, scene, object, or piece of equipment, respectively. These individuals are characterized using classes and properties defined in the scenario ontology. Furthermore, every descriptor associated with a scene, an object, or a piece of equipment is connected to relevant synthetic content, which consists of hierarchical, interconnected and reusable 3D components, 2D graphics, as well as scripts that implement animations and interactions. The scenario ontology and descriptors are comprehensively described in Section IV. The creation of synthetic 3D content for scenes, objects, and equipment is achievable through the use of our scene editor and the Unity game engine, but it is beyond the scope of this paper [19].

The scenario ontology and descriptors are stored in a *Semantic Repository*, which is a triplestore, whereas the 3D content of scenes and objects is stored in the *Content Repository*, which is a relational database.

The *Semantic Scenario Editor* is a client-server application comprising a *client* and a *server*. The client is a desktop application designed utilizing .NET, with a GUI described in the XAML language, while the server is a Java-based application that provides RESTful web services developed using the Spring library. The client offers a user-friendly interface that enables a *Training Manager* to create and modify training scenarios by utilizing the scenario ontology, descriptors, and 3D content stored in the repositories. The editor is described in detail in Section V.

IV. Semantic Representation of VR Training Scenarios

The proposed semantic representation of VR training scenarios is based on an ontology and uses domain-specific classes and properties, which are comprehensible to domain experts. The representation comprises two primary components: the *scenario ontology* which is common to different domains and applications, and descriptors that are domain-specific and built on top of the ontology.

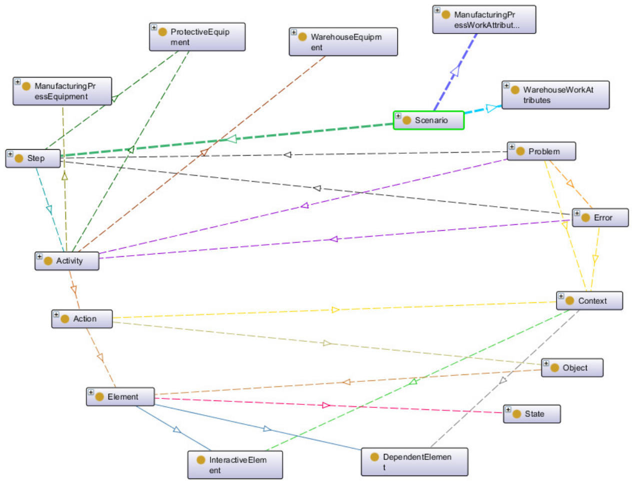

Fig. 2: Classes and properties of the scenario ontology.

The classes and properties that constitute the ontology are illustrated in Figure 2. They can be categorized into four distinct groups:

a) Training Objects: Training objects are the primary elements of every VR training scenario, e.g., forklifts, pallet trucks, and batteries. A virtual object is a tuple of an *object descriptor*, which semantically represents the physical object used in the training, and a 3D model, which visually represents the physical object. An object descriptor is an assertional box that describes individuals related to the physical training object using classes and properties specified in the scenario ontology. In an object descriptor, the physical object is represented by an individual of the *Object* class. An object individual comprises individuals of the *Element* class. Hence, it forms a hierarchy. Objects' elements have possible *states*. Elements' states can be changed during VR training. There are two types of *elements*: *interactive element* and *dependent element*. The state of an *interactive element* is changed by a trainee, whereas the state of a *dependent element* is changed as a consequence of a change of the state of an interactive element. Every object, element, and state has a name, description, and image intelligible to domain experts.

b) Virtual Equipment: Virtual equipment represents either work equipment, e.g., a toolkit from which users can select tools needed to complete the scenario, or protective equipment, e.g., helmets and gloves. A *virtual piece of equipment* is a pair of an *equipment descriptor*, which semantically represents the physical equipment, and a 3D model, which visually represents the physical equipment. An equipment descriptor is an assertional box that represents individuals related to the equipment using classes and properties specified in the scenario ontology. In an equipment descriptor, a piece of equipment is represented by an individual of a sub-class of the *Equipment* class. Different sub-classes of equipment may be specified depending on the particular domain of training. Every piece of equipment has a name and description, which are understandable to domain experts.

c) Training Scenes: Each VR training scenario is designed for a specific *VR training scene*, which comprises a *scene descriptor* and synthetic 3D content. The scene descriptor represents individuals related to the training scene using the classes and properties from the scenario ontology. A training scene is an individual of the *Scene* class, which includes individual objects of the *Object* class. Likewise, the synthetic content of the virtual scene includes the 3D models of the virtual objects. Every scene has a name and a description, which are understandable to domain experts.

In practical VR training applications, multiple virtual scenarios may be designed for slightly different virtual scenes, such as two factories with different placement of battery charging points. As a result, a scene may be a super-scene to other scenes, and each scene may inherit from another scene. The scene descriptors describe which objects are included or excluded in a scene or its sub-scenes, and every scene includes an object that is either included directly in the scene or included in its super-scene but not excluded in any scene on the inheritance path to the super-scene.

V. Training Scenario Editor

A. Architecture

The Semantic Scenario Editor is composed of a client-server system consisting of two main parts: the *Scenario Editor Server* and the *Scenario Editor Client*. The Scenario Editor Server is a Java-based program with RESTful web services using the Spring library, which accesses the *Semantic Repository* and the *Content Repository*. The system offers four services. The *Scene Service* allows for the selection of scenes that can be used to create training scenarios. Every scene can have a different set of available objects. The *Object Service* provides objects, their elements, and their respective states. The *Equipment Service* provides the available equipment for training in the application domain, which is common to all potential training scenarios. The *Workflow Service* provides information about the workflow of scenarios. The workflow consists of steps, which are divided into activities. Each activity consists of several actions as well as possible problems and errors. Such a structure allows for an easier understanding and editing of the scenario workflow. The Semantic Repository is supported by the Apache Fuseki server, which enables semantic reasoning and query processing.

The *Scenario Editor Client* is a user-friendly visual tool that training managers use. It is based on Windows Presentation Foundation. The main purpose of the tool is to permit the specification of training scenarios. The client displays scenario attributes and their possible values in different fields of visual forms (see Fig. 3). The attributes are accessed from and saved to the Semantic Repository via the Scenario Editor Server. The forms are presented in a simple layout that includes attribute names, text boxes, and drop-down lists where the manager can enter the necessary information. The drop-down lists show values obtained from the *scenario ontology*.

The general information includes the scenario title and the type of work, which may be either warehouse work or manufacturing press work. The manager also specifies whether the scenario is *elementary*, *complementary*, *regular*, *verifying*, or *ad hoc*. Finally, the manager selects the necessary pieces of protective equipment to complete the scenario from the list of all available equipment.

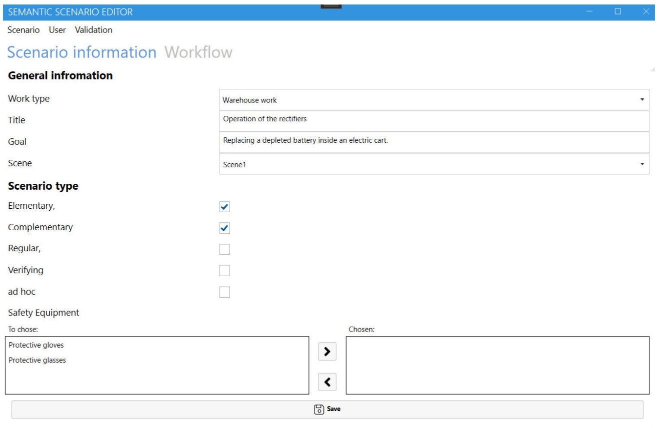

Fig. 3: General information about a scenario.

In addition to providing general information, the author also specifies a scenario's workflow, which includes steps, activities, and actions that trainees must perform. In each scenario, there must be at least one step that contains at least one activity, which in turn contains at least one action (see Section IV). Actions are linked to interactive and dependent objects' elements, as well as potential issues and errors that may arise during the action.

In the Scenario Editor, the workflow of each scenario is presented in the form of a tree, which is a widely used and easy-to-understand method for displaying hierarchical data (see Fig. 4). The tree includes scenario steps, activities, actions, problems, errors, and objects, which are represented by distinct icons. The editing manager can expand and collapse the list of sub-items for each item in the tree. In addition, there are optional sub-items for grouping actions, errors, and problems in activity and problem items. Using the toolbar and context menu, the author can visually add, modify, and delete items in the tree. Moreover, the order of the steps, activities, and actions can be changed by dragging and dropping.

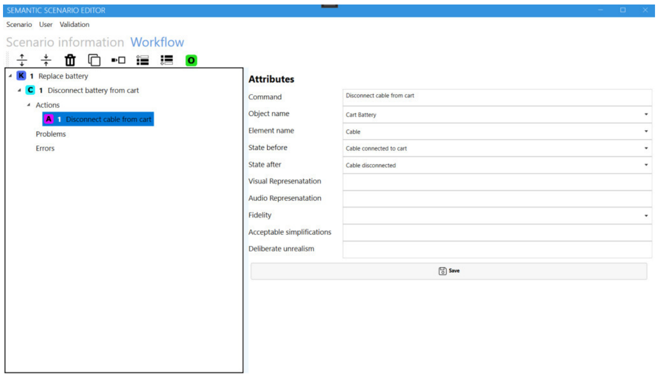

Fig. 4: Tree view of a scenario's workflow.

VI. Example VR Training Scenario

A. General information

We have implemented a prototype of a VR training system based on the semantic training scenario representation using the Unity game engine. To permit the completion of the training scenario without using the hardware controllers, the *Oculus Integration* plug-in has been utilized together with supplementary plug-ins. Thanks to the advanced capabilities of the equipment, which allows for direct tracking of users' hands, prospective trainees are not required to learn how to operate VR controllers. Instead, they can interact with objects in the virtual environment directly using their own hands, which increases the level of immersion.

The system has been developed utilizing resources provided by *Amica S.A.*, which is one of the largest producers of household equipment in Poland. These resources consist of in-depth information regarding the training process for the designated workstation, as well as comprehensive information regarding the relevant equipment. These resources became a base for creating scenes and objects used in training scenarios regarding the operation and maintenance of electric carts, forklifts and industrial presses. The elements of the scene are described by scenario descriptors, scene objects, and equipment descriptors. The main goal of the system is to teach the trainee, how to use industrial equipment safely.

B. Training scenario

In order to show the functionality of the system, we have created an example scenario that uses the developed semantic representation. The scenario has been formulated to illustrate knowledge about the procedure of replacing a depleted battery inside an electric cart. Trainees can visually inspect the condition of the cart, as per the official documentation of the real-life training session, and are also capable of operating the cart, provided that the battery is functional and all connections are properly established. All components have been designed and programmed to accurately replicate the simulated reality, including such factors as cable physics, electrical plug connections, battery charge level and depletion during operation, as well as a realistic electric cart driving system that permits trainees to execute the battery replacement procedure at any designated workstation.

The workflow for this scenario has been developed in collaboration with domain experts to enable a realistic training experience for trainees. The scenario has been created in accordance with the provided guidelines, and block diagrams have been used to illustrate the training process. Functionality facilitating the monitoring of trainees' progress during training, as well as the provision of pertinent feedback regarding positive or negative training outcomes has been incorporated.

C. Training scene

The virtual scene in which the training scenario takes place has been constructed using materials furnished by the company as well as photographs obtained during a site visit to the production hall (Fig. 5). Adequate lighting has been selected to accurately simulate the conditions present in the authentic production hall. Since the training scenario is conducted in only a small section of the hall, rendering is optimized for performance by limiting the display to the sector of the production hall utilized during the training, i.e., the section of the hall containing batteries and chargers, enabling charging and replacement of batteries in electric carts.

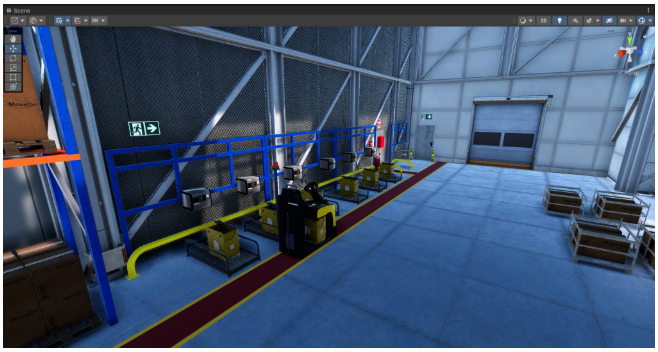

Fig. 5: Production hall in the scenario

D. Virtual objects

In industrial VR training scenarios, *virtual objects* representing real equipment are key to construction of accurate representation of a real-life training exercise. For the scenario, a number of 3D models have been developed specifically for this purpose, utilizing photographs obtained during the site visit to the production hall. The most critical virtual objects include the electric cart, battery, plugs, and charging stations (Fig. 6-9). These objects, in combination with the main *player object* representing the trainee, are employed to construct an environment that enables the execution of the scenario.

E. Scenario steps

By utilizing the aforementioned semantic representations of *Objects*, *Steps, Actions*, and the description of the scene, the initial training scenario has been created. The primary objective of the trainee is to replace a nearly depleted battery

Fig. 6: Electric cart

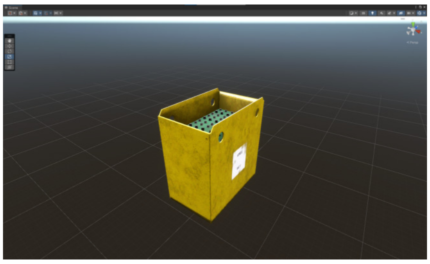

Fig. 7: Battery for the electric cart

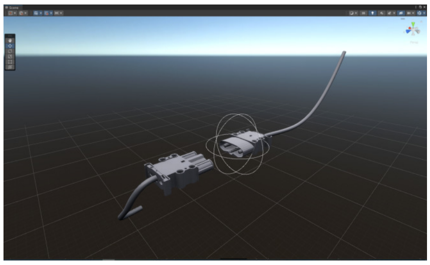

Fig. 8: Plugs of the electric cart

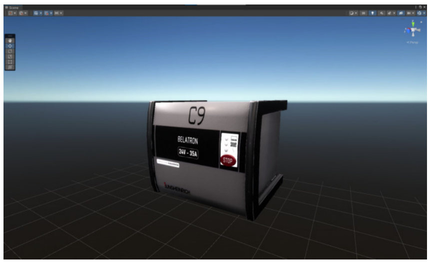

Fig. 9: Charging station for the battery

inside the electric cart with a new one from the charging station. The training scenario consists of the following steps:

1) Perform a visual inspection of the electric cart (Fig. 10).
2) Start the cart by ensuring that the battery is connected and locked and then use the electronic card to initiate the power-up sequence (Fig. 11).
3) Drive the cart into the correct position alongside the charging station (Fig. 12).
4) Replace the nearly depleted battery with a fully charged one (Fig. 13).
5) Connect the new battery unit to the cart and ensure that the blockade is properly in place (Fig. 14).
6) Conduct a visual and manual inspection to verify that the new battery unit is correctly installed and that the cart is in good working condition.
7) Drive the cart away from the charging station to finish the scenario.

At the beginning of the scenario, the trainee is situated in front of the cart and must either physically move into the correct position or use the appropriate hand gesture to teleport themselves into the desired position within the cart. The subsequent step is to verify that the cart is in good condition. The cart may appear fully functional or exhibit visual cues indicating its malfunction. This is achievable by adjusting various settings, which enable the *virtual objects* to exhibit signs of damage or malfunction. Once the visual inspection is completed, the next objective is to activate the vehicle. This step is fairly intricate and may prove challenging for inexperienced trainees, however, this is a deliberate design decision, as the scenario is intended to provide trainees with a safe and controlled environment for practice.

Activating the electric cart involves several *Steps*. Initially, the trainee must verify that the battery is properly connected and securely locked inside the cart. Subsequently, an electronic card is utilized to initiate the vehicle's power-up sequence. If the battery is not appropriately connected and/or locked, a message will be displayed in the console log, indicating an *Error*. If this situation arises, the trainee may opt to either restart the scenario by pressing the "R" key on the keyboard or attempt to correctly connect and lock the battery and then reattempt to activate the vehicle. Once the trainee completes this process, the electric cart is primed for operation.

With the activation of the cart and the trainee correctly using the steering wheel with both hands, the electric cart can be driven. A refined driving model supports precise control of the cart, paralleling the real-world scenario. Additionally, the employment of hand tracking elevates the immersive nature of the driving experience.

The subsequent objective entails driving the cart into the appropriate position, allowing for the replacement of the battery *Object*. The trainee is permitted to drive the cart to any of the charging stations where a battery is available for replacement, as indicated by the illuminated lights at the stations. The charging stations are meticulously modeled and scripted to mirror their real-life behavior, thereby enabling trainees to learn about the various color-coded markings displayed at the stations, the available actions and how to respond in an emergency.

Once the cart is positioned correctly alongside the charging station, the trainee must utilize their electronic card to power down the vehicle and remove the battery blockade to unlock the battery. To remove the battery, the trainee must execute the appropriate hand gesture to grasp and extract the battery from the cart. Next, the trainee must unlock the fully charged battery from the charging station, retrieve it, and install it into the now empty slot.

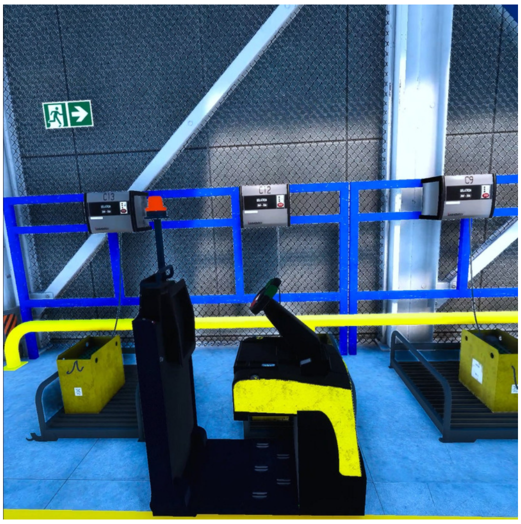

Fig. 10: Visual inspection of the electric cart

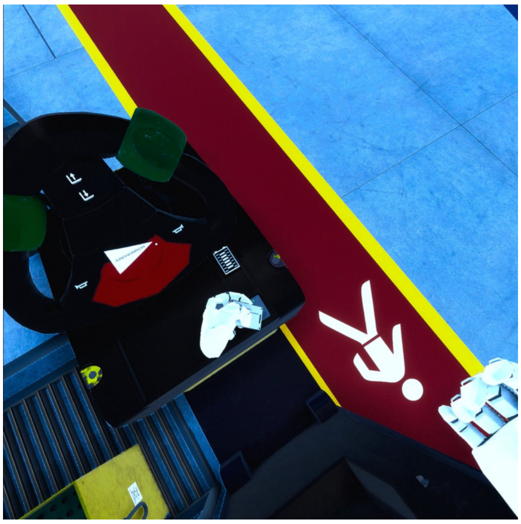

Fig. 11: Turning on the cart after visual inspection

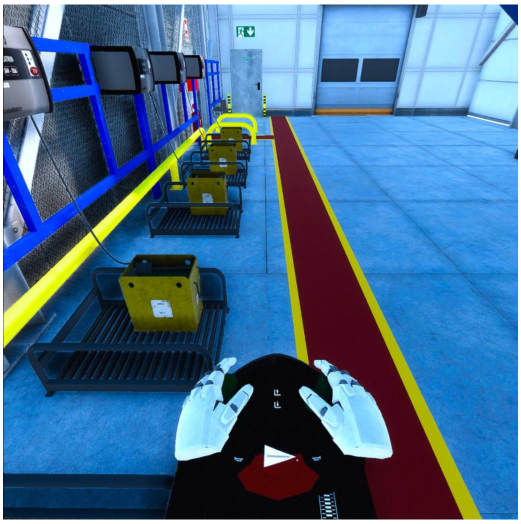

Fig. 12: Steering the cart into a station using hands

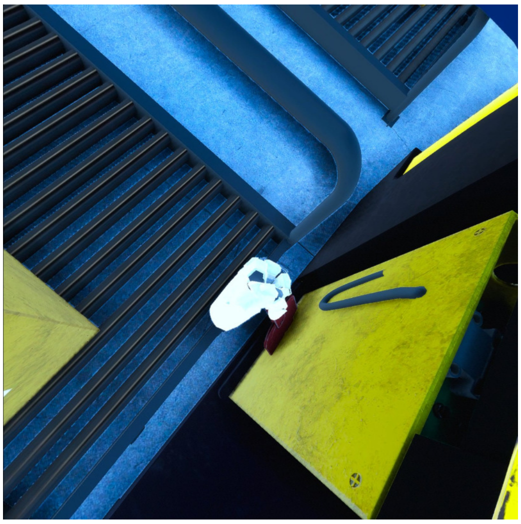

Fig. 13: Unlocking the battery using hand tracking

Following battery insertion, the trainee should confirm that the battery is correctly connected to the cart and that the blockade is securely in place. The cart may then be turned back on, and if all connections are established correctly, the trainee can resume driving the cart. This scenario is intended to train the trainees on the proper procedure for exchanging the battery, as well as teach them how to safely operate the electric cart. Trainees may learn how to drive the cart, how to adopt appropriate safety procedures while driving, and how to react in the event of an emergency.

F. Discussion

During the development of the presented training scenario, several discussions with the domain experts were conducted to ensure that the VR scenario accurately simulates real-life situations. An example of a significant alteration that has been implemented following these consultations is the need to hold the steering wheel of the electric cart with both hands during the operation. Without access to the appropriate documentation and expert knowledge, such issues could easily be overlooked.

During the battery replacement process, a trainee can encounter various challenges such as incorrectly connecting the battery, forgetting to lock it, or encountering a malfunction in the electrical system. These scenarios have been programmed to provide the trainee with a realistic learning experience and the opportunity to learn from their mistakes in a safe and controlled environment.

Overall, the scenario has been designed to provide the trainee with a realistic learning experience, replicating real-life situations as closely as possible. The use of VR technology, combined with accurate semantic representation of objects and realistic physics, creates a highly immersive and effective training environment.

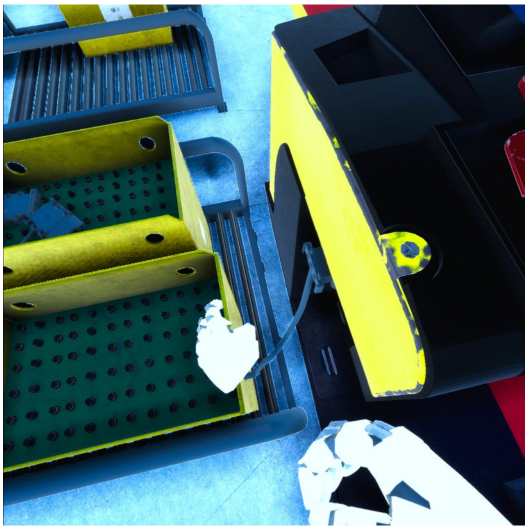

Fig. 14: Installing a new battery

VII. Conclusions

The proposed approach to the creation of VR training scenarios based on knowledge representation permits more flexible and precise modeling by utilizing domain concepts, rather than relying on low-level programming and 3D modeling. With the presented editor, trainers can easily create and modify scenarios in an efficient and intuitive manner. This makes the development of VR training environments accessible to non-technical users who can use domain terminology in the design process.

Future work includes extending the environment to allow for collaborative scenario creation by distributed users and integrating assessment of trainees' performance. Moreover, the scenario ontology will be extended to include concepts of alternative activities, which would be useful in cases when some tasks can be accomplished in multiple ways, such as in infrastructure error scenarios.

Acknowledgment

The presented research was funded by the Polish National Center for Research and Development under grant number LIDER/55/0287/L-12/20/NCBR/2021. The authors would like to thank Amica S.A. for providing training resources and insightful consultations.

References

[1] M. Dragoni, C. Ghidini, P. Busetta, M. Fruet, and M. Pedrotti, "Using ontologies for modeling virtual reality scenarios," in *The Semantic Web. Latest Advances and New Domains*, F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, and A. Zimmermann, Eds. Cham: Springer International Publishing, 2015, pp. 575–590.

[2] H. Fujita, M. Kurematsu, and J. Hakura, *Virtual Doctor System (VDS) and Ontology Based Reasoning for Medical Diagnosis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 197–214. [Online]. Available: https://doi.org/10.1007/978-3-642-33959-2_11

[3] J. Flotyński and K. Walczak, "Ontology-Based Representation and Modelling of Synthetic 3D Content: A State-of-the-Art Review," *Computer Graphics Forum*, vol. 35, p. 329–353, 2017.

[4] O. De Troyer, F. Kleinermann, B. Pellens, and W. Bille, "Conceptual modeling for virtual reality," in *Tutorials, posters, panels and industrial contributions at the 26th Int. Conference on Conceptual Modeling - ER 2007*, ser. CRPIT, J. Grundy, S. Hartmann, A. H. F. Laender, L. Maciaszek, and J. F. Roddick, Eds., vol. 83. Auckland, New Zealand: ACS, 2007, pp. 3–18.

[5] M. Gutiérrez, D. Thalmann, and F. Vexo, "Semantic virtual environments with adaptive multimodal interfaces." in *MMM*, Y.-P. P. Chen, Ed. IEEE Computer Society, 2005, pp. 277–283.

[6] E. Kalogerakis, S. Christodoulakis, and N. Moumoutzis, "Coupling ontologies with graphics content for knowledge driven visualization," in *VR '06 Proceedings of the IEEE conference on Virtual Reality*, Alexandria, Virginia, USA, Mar. 2006, pp. 43–50.

[7] M. Attene, F. Robbiano, M. Spagnuolo, and B. Falcidieno, "Characterization of 3D Shape Parts for Semantic Annotation," *Comput. Aided Des.*, vol. 41, no. 10, pp. 756–763, Oct. 2009.

[8] L. De Floriani, A. Hui, L. Papaleo, M. Huang, and J. Hendler, "A semantic web environment for digital shapes understanding," in *Semantic Multimedia*. Springer, 2007, pp. 226–239.

[9] P. Kapahnke, P. Liedtke, S. Nesbigall, S. Warwas, and M. Klusch, "ISReal: An Open Platform for Semantic-Based 3D Simulations in the 3D Internet," in *International Semantic Web Conference (2)*, 2010, pp. 161–176.

[10] S. Albrecht, T. Wiemann, M. Günther, and J. Hertzberg, "Matching CAD object models in semantic mapping," in *Proceedings ICRA 2011 Workshop: Semantic Perception, Mapping and Exploration, SPME*, 2011.

[11] M. Fischbach, D. Wiebusch, A. Giebler-Schubert, M. E. Latoschik, S. Rehfeld, and H. Tramberend, "SiXton's curse - Simulator X demonstration," in *Virtual Reality Conference (VR), 2011 IEEE*, M. Hirose, B. Lok, A. Majumder, and D. Schmalstieg, Eds., 2011, pp. 255–256. [Online]. Available: http://dx.doi.org/10.1109/VR.2011.5759495

[12] P. Drap, O. Papini, J.-C. Sourisseau, and T. Gambin, "Ontology-based photogrammetric survey in underwater archaeology," in *European Semantic Web Conference*. Springer, 2017, pp. 3–6.

[13] Trellet, M., Férey, N., Flotyński, J., Baaden, M., Bourdot, P., "Semantics for an integrative and immersive pipeline combining visualization and analysis of molecular data," *Journal of Integrative Bioinformatics*, vol. 15 (2), pp. 1–19, 2018.

[14] Y. Perez-Gallardo, J. L. L. Cuadrado, Á. G. Crespo, and C. G. de Jesús, "GEODIM: A Semantic Model-Based System for 3D Recognition of Industrial Scenes," in *Current Trends on Knowledge-Based Systems*. Springer, 2017, pp. 137–159.

[15] M. Fuchs, F. Beckert, J. Biedermann, and B. Nagel, "A collaborative knowledge-based method for the interactive development of cabin systems in virtual reality," *Computers in Industry*, vol. 136, p. 103590, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361521001974

[16] B. Youcef, M. Ahmad, and M. Mustapha, "Ontophaco: An ontology for virtual reality training in ophthalmology domain – a case study of cataract surgery," *IEEE Access*, vol. PP, pp. 1–1, 11 2021.

[17] F. Longo, G. Mirabelli, L. Nicoletti, and V. Solina, "An ontology-based, general-purpose and industry 4.0-ready architecture for supporting the smart operator (part i – mixed reality case)," *Journal of Manufacturing Systems*, vol. 64, pp. 594–612, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612522001303

[18] J. Flotyński, M. Krzyszkowski, and K. Walczak, "Semantic Composition of 3D Content Behavior for Explorable Virtual Reality Applications," in *Proceedings of EuroVR 2017, Lecture Notes in Computer Science*, J. Barbic, M. D'Cruz, M. E. Latoschik, M. Slater, and P. Bourdot, Eds. Springer, 2017, pp. 3–23.

[19] M. Maik, P. Sobociński, K. Walczak, D. Strugała, F. Górski, and P. Zawadzki, "Flexible photorealistic vr training system for electrical operators," in *Proceedings of the 27th International Conference on 3D Web Technology, New York, NY, USA*, ser. Web3D '22. Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3564533.3564561