Towards Enhanced Guiding Mechanisms in VR Training Through Process Mining

Enes Yigitbas, Sebastian Krois, Sebastian Gottschalk and Gregor Engels
Institute of Computer Science, Paderborn University, Zukuntsmeile 2, Paderborn, Germany

Keywords: Virtual Reality, Process Mining, Usability Evaluation.

Abstract: Virtual Reality (VR) provides the capability to train individuals to deal with new, complex, or dangerous situations by immersing them in a virtual environment and enabling them to learn by doing. In this virtual environment, the users usually train a sequence of different tasks. With that, most VR trainings have an underlying process that is given implicitly or explicitly. Although some training approaches provide basic guidance features, when analyzing the execution of the training, the process itself is often not considered, even if the process is one of the primary aspects to train in many cases. In this paper, we present VR-ProM, a framework that enables to use process mining techniques by supporting logging, analysis of execution logs of training sessions, and provision of guiding mechanisms to enhance VR training applications. To evaluate our framework and to investigate whether the integration of process mining techniques enables us to support the enhancement of VR-based training applications, we performed a two-staged user study based on a VR warehouse management training application. To analyze the effectiveness and subjective usability of the VR training, we performed two rounds of user studies and compared the results before and after we integrated the guiding mechanisms driven by process mining. Initial usability evaluation results show that with the help of VR-ProM the trainees made 40% fewer mistakes in the example VR training application and that the overall user satisfaction could be increased.

1 INTRODUCTION

Recent advances in Virtual Reality (VR) technology and the increased availability of VR-equipped devices enable a wide range of applications in various domains such as medicine (Gurusamy et al., 2009), robotics (Yigitbas et al., 2021), or marketing (Alcañiz et al., 2019). This work focuses on the use of VR for training which is defined as the provision of knowledge and skills in an interactive manner (Antonacopoulou, 2001). In this context, VR provides the capability to train individuals to deal with new, complex, or dangerous situations by immersing them in a virtual environment and enabling them to learn by doing.

While VR-based trainings usually rely on a process model which is given implicitly or explicitly for the task to learn, none of the existing VR-based training approaches make use of this process knowledge to improve the guidance of the training application by applying process mining techniques. As a consequence of this, the users of such VR-based trainings have to face a "one-size-fits-all" training with fixed guidance or a freestyle training approach with no guidance. However, due to learning style and experience level differences as well as task needs, it is important to consider the underlying processes and analyze their execution logs for engineering effective and user-friendly VR-based training applications. For this purpose, *Process Mining* can be seen as a promising technique to enhance the guiding mechanisms in VR-based trainings. However, the integration of process mining techniques and their implication for VR-based training applications have not been broadly researched so far. Although process mining has a big potential to analyze and optimize training processes, a systematic method how to integrate process mining techniques for VR-based trainings is missing. In addition to that, it is important to investigate whether process mining techniques are beneficial to enhance the guiding mechanisms in VR-based trainings. To the best of our knowledge, there is no approach existing that deals with the application of process mining to enhance VR-based training applications. This together leads to the research question of our work: *"How can we use process mining techniques to en-*

Yigitbas, E., Krois, S., Gottschalk, S. and Engels, G.
Towards Enhanced Guiding Mechanisms in VR Training Through Process Mining.
DOI: 10.5220/0011651600003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 2: HUCAPP*, pages 152-159
ISBN: 978-989-758-634-7; ISSN: 2184-4321

hance the guiding mechanisms in VR-based trainings?"

To answer this question, our work contains the following contributions. Firstly, we have developed a conceptual solution for the integration of process mining techniques in VR-based training applications. Secondly, we have implemented the conceptual solution as a framework, called VR-ProM, for Unity[1] projects to show its functionality in action. To evaluate our process mining solution approach, we have used an example VR-based training application from the domain of warehouse management and conducted a two-staged user study.

The rest of the paper is structured as follows. In Section 2, we present and discuss the related work. In Section 3, we describe the conceptual solution and implementation of our process mining approach for VR-based training applications. In Section 4, we present the user study and discuss the main results of the usability evaluation. In Section 5, we conclude the paper and give an outlook for future work.

2 RELATED WORK

Virtual Reality-based training applications have been discussed in the past for various application domains. In the following, we briefly draw on prior research into *Virtual Reality Trainings*, *Execution and Interaction Logging* as well as *Process Mining for Training Scenarios*.

2.1 Virtual Reality Trainings

Virtual Reality trainings have been used for several years now. For example, Liang et al. (Liang et al., 2019) developed a training game for miners, which teaches them how to spot loose rocks in underground mines to decrease rock-related hazards. Another example is given by Shen et al. (Shen et al., 2019) who simulated a marine ship to train marine engineers to work on a real ship. Furthermore, Wang et al. (Wang et al., 2017) created a surgeon training for medical students. Based on this approach, they can train surgeries in virtual reality before they start operating on a living patient.

While the above-mentioned VR-based training applications support the training process for a specific domain-relevant task, they do not make use of data logging or even if they record data, the results are not used to improve the training application by adjusting the guiding mechanisms.

[1]www.unity.com

2.2 Execution and Interaction Logging

In (Vidani and Chittaro, 2009), Vidani et al. present an approach where the combined use of task models and logs of user interactions are investigated to analyze the training process of serious games for emergency medical procedures.

Furthermore, Harms (Harms, 2019) presents an automated usability evaluation approach for VR applications. To enable logging, he presents a tool that crawls through the objects placed in a scene of the VR application to evaluate and automatically detect relevant log actions performed by the user.

A further approach where monitoring of execution logs of VR applications is considered is presented by Zahabi et al. (Zahabi and Abdul Razak, 2020). Here, the authors argue that VR trainings should be adapted based on the user's capabilities, performance, and needs. For this purpose, they provide a systematic literature review and a framework for adaptive VR-based training including performance measures, adaptive logic, and adaptive variables.

While the above-mentioned approaches enable conformance checking by taking a process or interaction log and comparing it to an existing process or task model, they do not support process discovery and enhancement which is possible using process mining.

2.3 Process Mining for Training Scenarios

Dolak et al. (Dolak, 2019) took the logs of the online education platform MOODLE[2] to analyze the students' behavior following an online course.

In the context of warehouse management, Paszkiewicz and Zbigniew (Paszkiewicz, 2013) analyzed the product management process of a mattress-producing company. The process describes how to treat pallets with freshly produced mattresses until they are shipped to the customer.

Fernández-Gallego et al. (Fernández-Gallego et al., 2013) present a learning analytics framework for 3D educational virtual worlds that focus on discovering learning flows and checking their conformance through process mining techniques. The core of this framework is an Opensim-based virtual world platform that has the ability of monitoring and registering the events generated by students and teachers.

Furthermore, Cerezo et al. (Cerezo et al., 2020) introduce a process mining approach for self-regulated learning assessment in the context of e-learning.

[2]https://moodle.com/

One approach where VR and process mining are combined is presented by Roldán et al. (Roldán et al., 2019). In this approach, the authors describe a training system for Industry 4.0 operators in complex assemblies based on VR and process mining. While the main idea of this approach is similar to ours, they are not focusing on the improvement of the VR interface by adjusting the guiding mechanisms to the needs of the end-users.

3 CONCEPT AND IMPLEMENTATION

To answer our research question and to integrate process mining techniques for the analysis of VR-based trainings, we have developed a VR process mining analysis framework called VR-ProM. A high-level architectural overview of VR-ProM is shown in Figure 1.

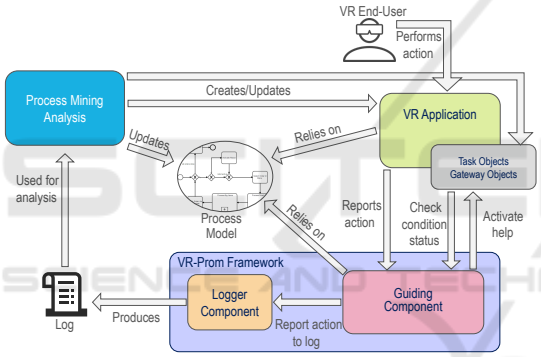

Figure 1: Architectural Overview.

In VR-ProM, we assume that there is already an existing VR-based training denoted as *VR Application* which relies on an explicit *Process Model*. The *VR End-User* performs an action in the VR-based training that is forwarded to the *Guiding Component* and logged by the *Logger Component*. In addition to this, the action is evaluated. Therefore, the *Guiding Component* needs to know where in the process the user is. To determine that, every application needs custom *Check condition status* operations which determine which path the user has to follow. When the user executed the right task, the *Guiding Component* also needs to check if the task was executed completely. If one of these conditions does not hold, the help mechanism can be activated. For every application, the help mechanisms need to be implemented individually. It also needs to be defined, when the help is to be activated (i.e. which preconditions need to hold before showing the help). That can be based on *Process Mining Analysis*, e.g. by determining how many mistakes may happen before the help is activated.

The *Logger Component* allows developers to capture the users' behavior in a format that can be used directly for *Process Mining Analysis*. The *Guiding Component* takes a process model as input. Then, it supervises if the user acts according to the specified process model. Additionally, the opportunity to easily implement guiding mechanisms into a *VR Application* is possible. Which guiding mechanisms are used and at what point they are applied can be based on the results of a *Process Mining Analysis*. The user's behavior can be continuously logged so that we can iteratively use process mining to successively improve the guiding. In the following sections, each component of the architectural overview will be described in more detail.

3.1 VR Application - Warehouse Management Training

For an illustration of the main concepts of the VR-ProM framework, we firstly describe the VR application that serves as a basis and motivational real-world example for our solution idea. In our case, we take an existing VR application from the domain of warehouse management training. In the area of logistics, warehouse management involves a complex process that consists of different order picking tasks. Due to this complex process, stock discrepancies and misplaced wares are typical problems that often occur. To overcome this problem, we have developed a VR training application that integrates an existing warehouse management system and trains typical order picking processes. In the following, the main functionality and structure of this VR-based training application will be briefly described to guarantee a better understanding of the forthcoming concepts concerning the integration of process mining. First of all, Figure 2 shows how the warehouse is structured.

The core objects in a warehouse are the items stored in the warehouse. There are different types of items. To limit the possible item types, we decided there should be *Small Items* and *Big Items*. With that, we have two item categories that need to be treated differently. The small items can be grabbed and carried by hand. We also use a *Carrying Robot* in which the user can put small items, and which can follow them through the warehouse. The robot allows each user to collect multiple small items at the same time, so they do not have to walk the same way again for every item. Small items can be *fragile* or *robust*. These attributes do not change their behavior but need to be considered when putting them into the *Picking Cart*

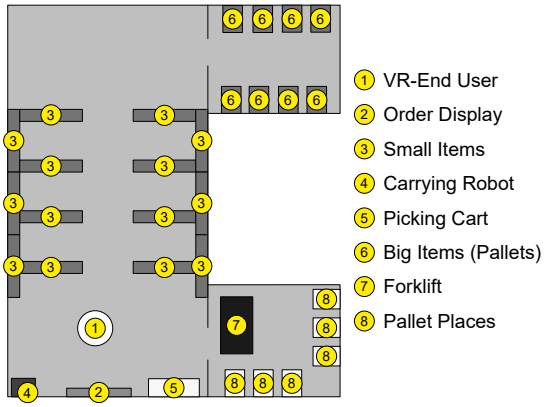

Figure 2: Floor Plan of the Warehouse Building.

for shipping. The picking carts are boxes, in which the user puts items ready to be shipped. To keep it neat, there is one picking cart for every order. A picking cart itself is divided into two sections, one for fragile items and one for robust ones. When placing items in the picking cart, the user has to place them according to their attributes. Big items are too heavy to carry by hand. They are placed on a pallet which requires the player to use a *Forklift* for moving them. To increase the immersion, the *Forklift* is controlled by the user's movement, not by pushing buttons on the controller. Therefore, it has a handle that can be grabbed. By moving the handle, the user can move the *Forklift*. The *Forklift* additionally needs two buttons to enable the up and down lifting of its forks to carry the pallets. For shipping, the pallets need to be placed in designated *Pallet Places*.

3.2 VR-ProM Framework

In the following, we describe our novel VR-ProM framework[3] in more detail by focusing on its two main components *Logger* and *Guiding*.

3.2.1 Logger Component

For analyzing the user's behavior and training performance we can log the events occurring during the training. The *Logger Component* produces a log that conforms to the XES standard[4], so we can use it with most of the existing process mining tools. To be able to easily combine traces from multiple computers into one log, the *Logger Component* writes down every playthrough in its own file and combines them to the actual log before ending the application. With that, we can use multiple traces and combine them into a

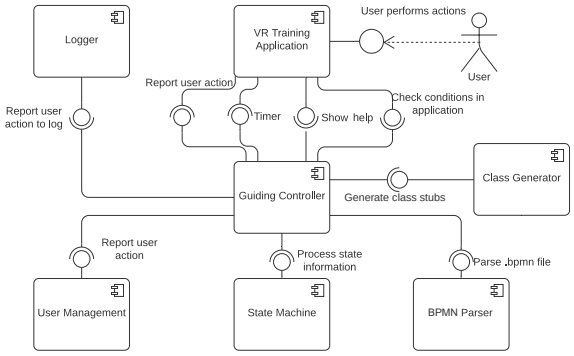

Figure 3: Component Overview.

single log later. To be able to use the *Logger Component* outside this application example, it is independent of the application. First, we need the *Logger Component* itself, which can be called to perform the logging. To ensure the last requirement, only one instance of the logger may be running at the same time. It needs at least three functions. One for initializing the logger and opening a new trace, one to close the log, and finally one for logging itself. The last one gets the event and logs it accordingly. Every event needs its own class where the class attributes are the attributes occurring in the log. Therefore, all events are an *XesEvent*. Every public field which implements the *XesAttribute* interface in the event class will be logged as an *XesAttribute* or *XesExtension* (when possible). We support primitive types and collections such as lists to also allow nested attributes.

3.2.2 Guiding Component at Design-Time

Besides the logging functionality described above, our VR-ProM framework provides a *Guiding Component* that enables assistance mechanisms in a VR-based training application to better help and assist the trainees. It is realized with the *GuidingController* script in the framework. It consists of two groups of functionalities. The first group includes features needed in the Unity editor while developing the VR-based training application. The second handles the guiding at runtime. Before integrating help mechanisms, we need to know the process we want to teach. Many tools allow the specification of process models. In the current version of the VR-ProM framework, we support process models saved as .bpmn file (e.g. created by the *Camunda Modeler*[5]). The process may contain tasks, exclusive gateways, start, and end events. To load the process model from the .bpmn file, we need a *BPMN Parser* which is used in both functionality groups.

[3]https://github.com/Quegg/VR-Prom
[4]https://xes-standard.org
[5]https://camunda.com/download/modeler/

Figure 3 shows the top-level components of the *Guiding Component* and the very basics of their communication. The *Guiding Controller* uses the *BPMN Parser* to get the process as input. Then, it creates an empty stub file for the following elements of the reference training process.

- *Task:* For every task, we need an event to log and an object which is loaded in the *VR Application* to actually interact with the trainee and all objects in the warehouse. This object will later contain the guiding mechanisms for its task.
- *Gateway:* Since a gateway is not an event that can be logged, we do not need a logging event. But we do need an object which is loaded in the scene, just like the task does. This is necessary, as the gateways have to check the state of their conditions to determine the next task to execute.
- *Error Events:* Error events are not parsed from the process. They can be added manually afterward. When adding an error event, we create a logging event and a class to be loaded in the *VR Application* as well.

After that, the developers can adjust the stubs, so they fit into their application. The core class of this project is the *GuidingController*. It handles the generation of all class stubs and manages the whole guiding during the simulation. To activate the *GuidingController*, the corresponding script needs to be added to a *GameObject*. In Unity, a *GameObject* is the most basic component which can be seen as a container holding everything being present in a scene ranging from visible 3D models to invisible scripts performing some action.

3.2.3 Guiding Component at Run-Time

In addition to the components we described above, we implemented a very lightweight *State Machine* (see Figure 3). It holds the information about the whole process and knows the task the user is currently working on. It does not need to execute anything but it has to know the current state and provide some utility functions on the process, e.g. getting all possible tasks following the current one.

Here, we first check the simplest case of the next element being a task. In that case, we do not need to check any conditions and simply return to the next task. If the next element is a gateway, we check its conditions implemented by the developers. A gateway provides a method to *CheckConditions*. It receives a list of all possible next elements and returns the name of the correct one. To use that, we first retrieve the names of all the following elements from the State Machine. Then, we pass this list to the gateway and use the returned object. If it is a task, we return this task. Otherwise, we check the conditions of the next gateway the same way until we eventually reach a task.

User management is optional and can be used to assign the playthrough results to user IDs to group or compare them later.

To show help to the user, the developers can implement custom help features (navigation arrows, object highlighting, etc.) for every task and error. If no custom help is provided, we provide generic help as a fallback. This generic help will be based on the given process and shows a simplified part of the process model depicting where the user currently is, what are the possible next tasks, and which of the next tasks is the right one. Figure 4 shows what this looks like in the simulation.

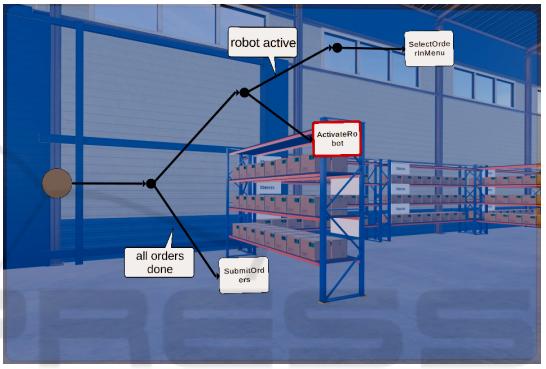

Figure 4: Exemplary Tasks to Execute During the Training Process.

To generate this view, we take the last executed task as input and display all possible next tasks together with the paths leading to them. We show the exclusive gateways as black circles which divide the path. If a path has a condition, it is shown using a speech bubble. To help the user orientate in this view, we added a red outline on the current task to execute.

3.3 Process Mining Analysis

With all the concepts described above, we almost have all parts together to integrate process mining results into VR trainings. The only thing missing is the process mining analysis itself. Using process mining, we have three different techniques at hand. The first technique, we can use, is process discovery, to see how the process is executed in real life. This is a generalization of all recorded paths in this log. So we can see, how a general user executes the process. With that, we can find out where problems occur, or the correct order is not clear to the user. The second technique is conformance checking. As its purpose is to show where the execution in the log deviates from the de-

sired log, this seems to be suitable for our approach. As, in our case, the process executions strongly deviate from each other and the given process, we can see 0% accordance of the log to the process. These results do not help to detect problems, so we will not use conformance checking in our analysis. The third technique, enhancement, is used to enhance the discovered process map, so we can also consider the timing and frequency of event occurrences. To perform the process mining analysis, we use the academic version of Disco (Günther and Rozinat, 2012). The derived results of the process mining analysis will be described in the next section which deals with the evaluation of our VR-ProM framework.

4 EVALUATION

To evaluate our framework VR-ProM and to investigate whether the integration of process mining techniques supports the enhancement of VR-based training applications, we performed an initial user study based on the described warehouse management training application. In the following, first, we describe the evaluation setup of our user study. Then, we describe the guiding mechanisms that were determined based on the application of process mining. Those guiding mechanisms were then integrated into the VR-based training to increase its efficiency, effectiveness, and user satisfaction. To check this, a second user study took place.

4.1 Evaluation Setup

To evaluate the benefit of VR-ProM, we have conducted a two-staged user study based on the VR warehouse management training application. In the first stage, the users performed the training where no guiding mechanisms were integrated. They needed to figure everything out on their own. The application logged the user's behavior, and we used these event logs to determine the values for the KPIs. Those KPIs were *Efficiency* (how long needs a user to complete the training), *Error Rate* (how often does the user do anything wrong), and the *Subjective Usability* (what is the SUS score the user gave for the training). Then, we analyzed the generated process map to identify problems and determine guiding mechanisms to avoid them. As soon as the integration was finished, a second group of users performed the training, but this time there were guiding mechanisms available. Due to the COVID-19 pandemic, we decided to perform the evaluation in a remote setting.

Due to the short-term conversion into remote evaluations, five participants completed the training in the first evaluation. They were aged between 18-46 years, two were female and three were male. They had different levels of technical knowledge concerning VR technology. In the second evaluation, 12 participants completed the training. Four were female and eight were male. They were aged between 16-56 years and had also different levels of technical knowledge about VR technology.

4.2 Guiding Mechanisms

To identify bottlenecks in the training process and thus determine possible guiding mechanisms, we performed a frequency analysis in Disco. In the following, exemplary remarkable cases are listed and will be shortly discussed.

- Some items were removed from the picking cart. One reason for that is that the users placed the wrong items into the picking cart and had to revert this for completing the order. Another reason could be that the users had to grab the item to scan it as they forgot it before.
- There were more items placed in the robot than removed. All the users completed the training with correct results (i.e. they placed all needed items in the picking carts). This means they collected some items they did not need at all. Reasons for this could be that the users did not know which item to collect or how many of them, so they took it anyway.
- The users let some items fall on the ground. This can have many reasons. For example, they did not fully understand the controls so an item falls down, or they picked the wrong item.
- The forklift's barcode was rarely scanned by the users. So the users either did not realize they need to scan it, or they forgot it.
- Often, the forklift hit an object. It is expected, that users with no or little experience in VR controls/forklift driving will hit some objects. But in five playthroughs, the users hit objects 220 times. Most likely, the users did not understand well enough how to control the forklift properly.

Concluding the insights gathered above, there are the following problems. First of all, it is not clear which items the user has to collect. Furthermore, the barcode scanner is not used properly or many barcodes are not scanned. In addition to that, when placing items in the picking cart, it is hard to distinguish between fragile and robust items. Finally, it could be observed that the forklift's main controls are not clear to the users.

The most obvious problem is, that the tasks are not executed in the correct order. Considering these problems, we came up with the following guiding mechanisms.

- Outline the items to pick up/the objects to interact with
- Outline the barcode scanner and show the buttons to activate it
- Outline the barcodes to scan
- Show the way the user has to go/drive
- Show the forklift's controls
- Show where and how to place the small items (fragile or robust) in the picking cart

When showing help, the framework will additionally show where in the process the user is, what tasks possibly follow the last one, and which one of them is the right one to execute.

4.3 Result Comparison

As our goal was to analyze whether our framework VR-ProM indeed enhanced the VR training application, we had to check if the users achieved better results in the second evaluation. To do so, we first need to clarify what better results mean. Regarding *Efficiency* and the *Error Rate*, we consider lower values as better values. These two KPIs can be summarized as the users' *Performance*. So a better *Efficiency* and *Error Rate* result in a better *Performance*. For the *Subjective Usability*, we use the SUS scale. Hence, a higher value corresponds to a better *Subjective Usability*.

First, we compare the duration shown in Figure 5. We see that in the second evaluation the users took 10% longer to complete the training, on average. However, we see that the second results vary in a broader range, as some users can take more advantage of the guiding mechanisms than others. The single dots in Figure 5 denote single values which are far apart from the others. All in all, this means, that the *Efficiency* was slightly worse in the second evaluation.

The second value to consider is *Subjective Usability*. For this, every user filled out the SUS questionnaire. With that, we determine the values shown in Figure 6. Here, we see that the maximum rating is equal. But there are fewer low ratings and the ratings overall vary less. The average SUS score increased by 6%. This growth itself is not very expressive. But with also comparing the distribution, we can conclude the *Subjective Usability* slightly increased.

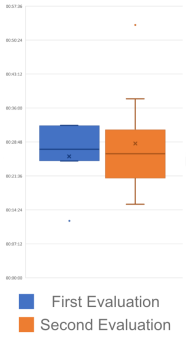

Figure 5: Duration.

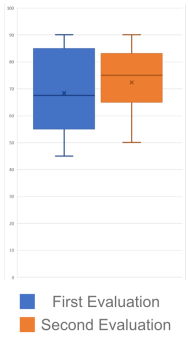

Figure 6: SUS Score.

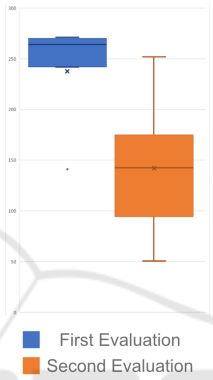

Figure 7: Error Rate Full.

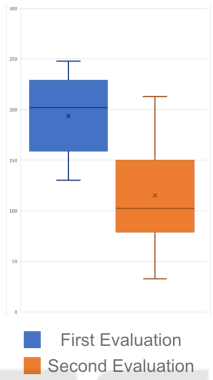

Figure 8: Error Rate.

Now, we compare the *Error Rate*. For our evaluation, there were two different error rates. Basically, an error occurs when the user did something that was not meant to do. This can be an error event (something that should never happen) as well as a normal task execution but of a task which should not be executed at the current time. *Error Rate Full* contains all these events. We excluded the *ForkLiftHitsObject Event* for our main *Error Rate*. The forklift's movement was very hard to control for the players. They often experienced motion sickness and could not concentrate on steering properly. Additionally, in a real-world warehouse, the employees are familiar with controlling the forklift. We do not want to train driving a forklift but help the user learn the process. For completeness, we provide both datasets here. We see that in both datasets, the minimum, maximum, and average error rate is much smaller in the second evaluation. For example, the average amount of errors for the main *Error Rate* decreased by 40%. However, the results of the second evaluation have a higher standard deviation in both datasets. So we can say, that we achieved a better *Error Rate* in the second evaluation.

4.4 Discussion and Threats to Validity

To summarize the main results of our usability evaluation, Regarding *Efficiency*, we found out, that the users needed on average 10% longer to complete the training with the integrated help. However, considering the *Error Rate*, we can see improvements when providing help during the training. The *Error Rate* decreased by 40% in the second playthrough compared with the first. The *Usability*, measured with the SUS scale, improved by 6%. This is only a minor change, but we can see that, in general, there were fewer low scores.

However, an important threat is the limited number of participants. Here, we need larger experiments with more heterogeneous groups to derive statistically representative results that help us to generalize our ideas for various application domains.

5 CONCLUSION AND OUTLOOK

In this paper, we have introduced the VR-ProM framework that supports the logging of VR-based training applications and produces log data in standardized XES format that can be analyzed based on existing process mining tools. Furthermore, VR-ProM provides generic and flexible guiding mechanisms to improve the help and guiding mechanisms in VR-based training applications based on the process mining results. Based on an initial evaluation we have shown the benefit of our VR-ProM framework by applying it to a VR warehouse management training application.

While this shows the potential of process mining for VR-based training applications, further steps are needed to establish the application of process mining techniques for VR technology. First of all, larger evaluations with more participants are required. Furthermore, to see the full potential of our solution ide, the help mechanisms need to be iteratively adjusted and evaluated. Finally, it would be beneficial to have a process mining solution that works out of the box and supports an automated integration in various VR-based training applications.

REFERENCES

Alcañiz, M., Bigné, E., and Guixeres, J. (2019). Virtual reality in marketing: a framework, review, and research agenda. *Frontiers in psychology*, page 1530.

Antonacopoulou, E. P. (2001). The paradoxical nature of the relationship between training and learning. *Journal of Management Studies*, 38(3):327–350.

Cerezo, R., Bogarín, A., Esteban, M., and Romero, C. (2020). Process mining for self-regulated learning assessment in e-learning. *Journal of Computing in Higher Education*, 32(1):74–88.

Dolak, R. (2019). Using process mining techniques to discover student's activities, navigation paths, and behavior in LMS moodle. In Rønningsbakk, L., Wu, T., Sandnes, F. E., and Huang, Y., editors, *ICITL*, volume 11937 of *LNCS*, pages 129–138. Springer.

Fernández-Gallego, B., Lama, M., Vidal, J. C., and Mucientes, M. (2013). Learning analytics framework for educational virtual worlds. *Procedia Computer Science*, 25:443–447.

Günther, C. W. and Rozinat, A. (2012). Disco: Discover your processes. 940:40–44.

Gurusamy, K. S., Aggarwal, R., Palanivelu, L., and Davidson, B. R. (2009). Virtual reality training for surgical trainees in laparoscopic surgery. *Cochrane database of systematic reviews*, (1).

Harms, P. (2019). Automated usability evaluation of virtual reality applications. *ACM Trans. Comput. Hum. Interact.*, 26(3):14:1–14:36.

Liang, Z., Zhou, K., and Gao, K. (2019). Development of virtual reality serious game for underground rock-related hazards safety training. *IEEE Access*, 7:118639–118649.

Paszkiewicz, Z. (2013). Process mining techniques in conformance testing of inventory processes: an industrial application. In *Int. Conf. on Business Information Systems*, pages 302–313. Springer.

Roldán, J. J., Crespo, E., Martín-Barrio, A., Peña-Tapia, E., and Barrientos, A. (2019). A training system for industry 4.0 operators in complex assemblies based on virtual reality and process mining. *Robotics and computer-integrated manufacturing*, 59:305–316.

Shen, H., Zhang, J., Yang, B., and Jia, B. (2019). Development of an educational virtual reality training system for marine engineers. *Comp. Applic. in Engineering Education*, 27(3):580–602.

Vidani, A. C. and Chittaro, L. (2009). Using a task modeling formalism in the design of serious games for emergency medical procedures. In Rebolledo-Mendez, G., Liarokapis, F., and de Freitas, S., editors, *Conf. in Games and Virtual Worlds for Serious Applications*, pages 95–102. IEEE Computer Society.

Wang, R., Yao, J., Wang, L., Liu, X., Wang, H., and Zheng, L. (2017). A surgical training system for four medical punctures based on virtual reality and haptic feedback. In *IEEE Symp. on 3D User Interfaces*, pages 215–216.

Yigitbas, E., Karakaya, K., Jovanovikj, I., and Engels, G. (2021). Enhancing human-in-the-loop adaptive systems through digital twins and VR interfaces. In *16th Int. Symp. on Software Engineering for Adaptive and Self-Managing Systems*, pages 30–40. IEEE.

Zahabi, M. and Abdul Razak, A. M. (2020). Adaptive virtual reality-based training: a systematic literature review and framework. *Virtual Reality*, 24(4).