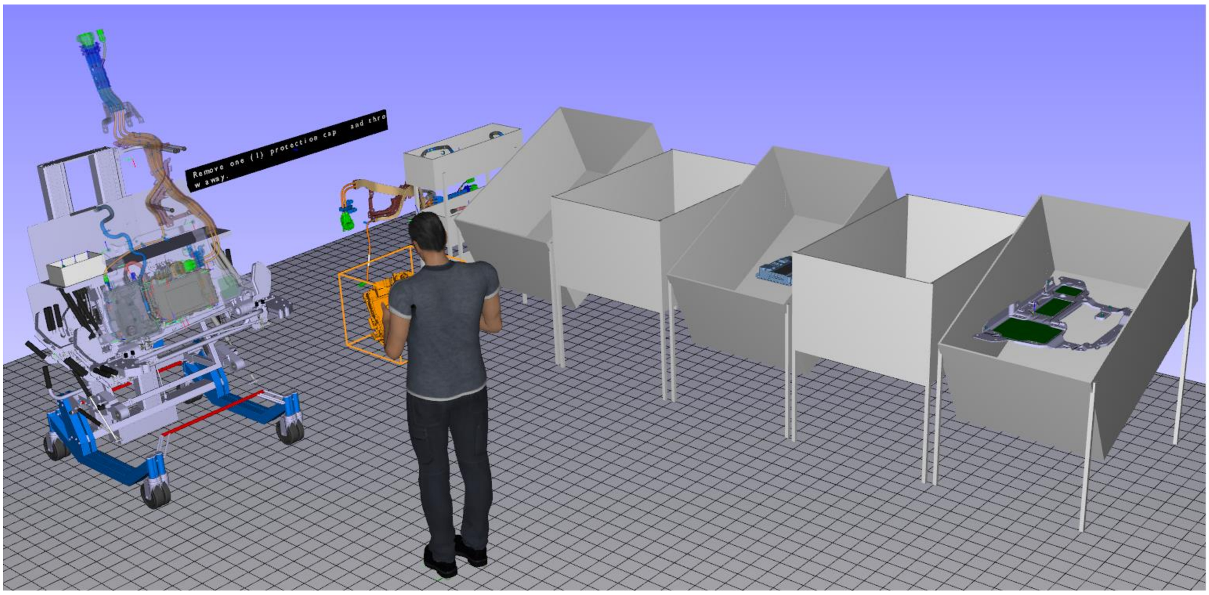
CHALMERS
UNIVERSITY OF TECHNOLOGY



Training Operators in VR

A scalable solution for efficient VR training creation

Master’s Thesis in Product Development

Geoffrey Melzani, Tony Quach

DEPARTMENT OF INDUSTRIAL AND MATERIAL SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2024
www.chalmers.se

Master’s Thesis 2024

Training Operators in VR

A scalable solution for efficient VR training creation

GEOFFREY MELZANI
TONY QUACH

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Training Operators in VR
A scalable solution for efficient VR training creation

GEOFFREY MELZANI
TONY QUACH

Supervisor: Dan Li, ME Digital, Volvo Cars Corporation
Supervisor: Puranjay Mugur, ME Digital, Volvo Cars Corporation
Supervisor: Henrik Söderlund, Industrial and Material Science, Chalmers
Examiner: Björn Johansson, Industrial and Material Science, Chalmers

Master’s Thesis 2024
Department of Industrial and Materials Science
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone + 46 (0)31-772 1000

Cover: Snapshot of VR operator training scene in IPS.

Gothenburg, Sweden 2024

Training Operators in VR
A scalable solution for efficient VR training creation
GEOFFREY MELZANI, TONY QUACH
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Industry 4.0 is revolutionizing the automotive industry at an increasing speed. Many companies are exploring different ways to make different operations, within the industrialization processes, more efficient. In that context, Virtual Reality (VR) technologies are being implemented by many players in the industry, with the aim to gain in speed-to-market. Currently, companies within different industries are exploring how to integrate VR for operator training. Therefore, this thesis, conducted at Volvo Cars Corporation (VCC), in Torslanda, Sweden, aims to identify how to reduce the effort for VR operator training scenes creation with scripting, with the goal to achieve auto generation as well as large-scale implementation of VR training.

The methodology implemented in this study is based on product development practices, with in this case, the product being VR operator training. First, the best adapted software to use was chosen: IPS, along with an HTC Vive hardware, to create and develop VR operator training and to envision its large-scale implementation. Furthermore, qualitative interviews and a literature review were conducted to acquire more knowledge, together with several tests and discussions with experts, to confirm and evaluate the findings. The topics covered were the identification of the relevant data to populate a VR operator training scene, the scripting for the auto generation of the scene itself and the learning facilitation, by considering the auto generation of pedagogical elements.

The results were that using IPS, implementing the created scripts for auto generating a VR training scene was possible, populated with all the elements required, including the assembly, the plant layout with fixtures and storage furniture, as well as pedagogical elements such as text instructions, visual cues, and motion sequences of an assembly manufacturing. Challenges on the way of the large-scale implementation of this new type of training were identified and how to try and fix them was envisioned, to draw a blueprint for the implementation of VR operator training at VCC.

Keywords: Industry 4.0, Assembly operator, VR training, Data mapping, Auto generation.

Acknowledgements

We would like to express our profound gratitude and appreciation to the individuals who have provided invaluable assistance, guidance, and support throughout the completion of our master's thesis.

First, we extend our heartfelt thanks to our supervisors at VCC, Dan Li and Puranjay Mugur. Dan, notably for his advice in project management as well as his contacts within VCC and in the industry and for the opportunities he gave us to take part in insightful meetings related to our topic. Puranjay, for his unwavering support and technical expertise that we could rely on the countless times we had software and hardware related questions, during the entire project.

A profound thank you goes to our supervisor at Chalmers, Henrik Söderlund, who was always supportive and available for us and helped with invaluable feedback for the redaction of this report. We would also like to acknowledge and thank our examiner, Björn Johansson.

Furthermore, we extend our appreciation to many people working at FCC, the provider of IPS, who were so supportive and allowed us to use the software to its full potential. Thus, we would like to give a special mention to Fredrik Norström, Product Area Lead at Industrial Path Solutions Sweden AB, who enlightened us with his experience with the software and more generally on how to deal with such project. The same goes to Jonas Kressin, Customer Success Manager at Industrial Path Solutions Sweden AB and Peter Mårdberg, from FCC who helped us to get proficient with IPS in no time and with Lua scripting and how to enable the VR possibilities within the software the best way.

Geoffrey Melzani and Tony Quach, Gothenburg, June 2024

Table of Contents

Nomenclature

AI – Artificial Intelligence – “Technology that enables computers and machines to simulate human intelligence and problem-solving capabilities.” (IBM, n.d.)

API – Application Programing Interface – “Set of rules or protocols that enables software applications to communicate with each other to exchange data, features and functionality.” (IBM, n.d.)

AR – Augmented Reality – See detailed definition Section 2.4.

AV – Augmented Virtuality – See detailed definition Section 2.4.

BOE – Bill Of Equipment – It “joins all assets, tools, devices, pieces of equipment or installations needed to carry out a manufacturing process”. (Inlean, n.d.)

BOM – Bill Of Materials – It “includes the substances or components required to generate a defined unit of finished product or a semi-processed, including their physical and chemical properties and specifications”. (Inlean, n.d.)

BOP – Bill Of Process – It “details the planned production approach for a specific product. […] The bill of process lists production line configurations, tools, machines, and equipment needed to make the product. The bill of process also contains the electronic work instructions that explain how to make it”. (Siemens, n.d.)

CAD – Computer Aided Design – “Way to digitally create 2D drawings and 3D models of real-world products before they’re ever manufactured.” (PTC, n.d.)

CC – Collaboration Context – “Container for product, plant, and process structures.” (Siemens, 2019)

FCC – Fraunhofer-Chalmers Center.

HMD – Head Mounted Display – “Type of computer display device or monitor that is worn on the head or is built in as part of a helmet. This type of display is meant for a total immersion of the user in whatever experience the display is meant for, as it ensures that no matter where the user’s head may turn, the display is positioned right in front of the user’s eyes.” (Rouse, 2017)

PLM – Product Lifecycle Management – “Holistic approach to product development. It incorporates every aspect of a product’s lifecycle – from innovation through to repurposing, reuse, and recycling.” (Technia, n.d.)

PLM system – “Suites of connected software bringing together people, processes and data, creating a seamless collaborative environment.” (Technia, n.d.)

SDK – Software Development Kit – “Set of software-building tools for a specific platform, including the building blocks, debuggers and, often, a framework or group of code libraries such as a set of routines specific to an operating system.” (IBM Cloud Education, 2021)

VCC – Volvo Cars Corporation.

VR – Virtual Reality – See detailed definition Section 2.4.

WI – Work Instruction – It “describe(s) the correct way to perform a certain task or activity. Each task is part of a larger process, so every step must be followed accordingly in order for the task to be performed properly.” (Fonseca L., 2022)

XR – Extended Reality – See detailed definition Section 2.4

1. Introduction

The fourth industrial revolution is set to transform extensively the future manufacturing and production processes with the emergence of smart factories and networked industrial environments. To achieve such novelties, the focus is on digitalization and data handling, with effectivity in mind (Raptis et al., 2019). In that context, Extended Reality (XR), which is spreading at a high pace throughout the industry, is bringing new data visualization and interaction possibilities. XR is actually an umbrella term containing distinct technologies, the two most significant ones being Augmented Reality (AR) and Virtual Reality (VR) (Rauschnabel et al., 2022). They allow new ways to handle data and are used for many purposes and for many steps of the industrialization of developed products in the manufacturing industry (Choi et al., 2015).

While being manufactured, products or parts of products may need to be assembled by human operators. These people need to be trained to ensure that products are assembled in a safe and error-free way. Traditionally, this is done in several steps, based on classroom training, and face-to-face on-site training with experienced operators (Dipesh et al., 2018). Such practices have been used for a long time, but they now show their limitations, especially when operators must work more and more with robots and automated tools or as emergency situations can never be recreated realistically enough and safely with traditional training (Patle et al., 2018). XR, and especially VR are very promising solutions for such purposes as it is shown in the literature and as some industry big players are already implementing in their operator training processes (VR Owl, n.d.). Using such high-end technology in such a new way, although being very interesting and beneficial, does not come without a cost: many barriers must be overcome to come up with a scalable solution for VR training creation. These challenges are numerous and of different natures as they may be coming from the industry and its reluctancy towards change because of settled habits and knowledge gaps (Badamasi et al., 2022), as well as from trainees themselves who may not know fully how to use VR tools (Doolani et al., 2020). Furthermore, when the creation of VR training environments was first investigated at Volvo Cars Corporation (VCC), engineering efforts needed and time required for the creation of these VR environments were the main challenges that emerged (Boström & Zamola, 2023).

1.1 Background

VCC is one of the biggest players in the automotive industry and it has strong values of safety and sustainability as it is at the forefront of the electrification of transportation (Volvo Cars, 2024). Nowadays, just like the whole automotive industry, it faces challenges in improving its speed-to-market. Virtual design and verification are important components to shortening the time it takes from finalized product designs to factories ready for the start of production. In current practices at VCC, VR is primarily used as early-stage visualization for factory design and product design (e.g., Design for Assembly, operator ergonomics). But there is now a will to use VR even more and in other product development phases. Hence, many actors in Sweden have gathered and have launched a project called PLENUM (PLENary multi-User developMent arena for industrial workspace), which this master’s thesis belongs to, that focuses on developing a multi-user platform for collaborative 3D work in the manufacturing area, with a focus on user-friendliness and interactivity (RISE, 2024). Within this project, VCC is collaborating with Chalmers University of Technology and Fraunhofer-Chalmers Center (FCC) to achieve improvement and innovation of its XR practices.

One of the subjects tackled in the PLENUM project is digitalized operator instructions and training using XR. Previous studies have been conducted on using VR to create instruction support to decrease the learning time and cost while enhancing the user experience and maintaining the same quality and the results were promising: a proof-of-concept was created (Boström & Zamola, 2023). Nevertheless, this proof-of-concept highlighted challenges, such as engineering skills needed, time spent on education and inefficient already available data utilization, to be able to create VR training environments. Furthermore, the creation of each new training scene took approximately 4-5 hours and with the current methodology, manufacturing engineers would need to master a new software (namely, Unity), which would be time consuming. Another more general challenge that emerges at VCC, when it comes to XR and new technologies in general is the acceptance, specifically for the management as well as for the engineers. All in all, these barriers would need to be considered and addressed for VR operator training to be developed further and at a larger scale at VCC.

1.2 Purpose and aim

For the recent use cases explored in VR operator training, barriers have been identified in terms of the amount of time, skills, and engineering efforts required to create the VR training scenes and to populate them with relevant training material and instructions for the operators. For a successful scaled-up implementation, such barriers need to be overcome. **Hence, the purpose of this thesis is to highlight and demonstrate how novel approaches can be leveraged to automate the creation of virtual environments for operator training.** Thanks to a facilitated creation process, people with not much VR experience would be enabled to create such VR scenes, therefore allowing for time and financial gains.

Regarding the time, skills, and engineering efforts needed to create efficiently the VR training scenes, two research questions were established and were as follow:

> ***RQ1:*** *What prerequisites are needed to implement scalable VR training scene creation and how to access the relevant data?*
> ***RQ2:*** *What effort is needed to achieve scalable VR training?*

To facilitate the learning for the operators when using VR for training, the following research question was established:

> ***RQ3:*** *To what extent can VR training pedagogical elements be auto generated?*

Furthermore, the RQ3 will cover the aspect of different possibilities to eventually achieve the auto generation of pedagogical elements, depending on the data available and their implementation in a VR training scene in an efficient way.

1.3 Delimitations

The scope of the project is quite broad. The software that should be used for training scenes creation is not determined from the start. Although, as it has been previously identified that VR is best suited to provide effective training experience, other XR technologies will be put aside from the beginning and will not be considered (Boström & Zamola, 2023). Another delimitation is that the assembly of the whole car will not be considered. In fact, only assembly sequences where the core of the work is realized by operators will be of interest. This corresponds to what is called final assembly at VCC.

Furthermore, although parts of sequences that will be used only come from VCC, the overall results and conclusions of this thesis are not specifically restricted exclusively to that case but can be seen and applicable in a more general way. Finally, in this project, VR operator training consists exclusively of picking and placing activities, realized in the assembly line, which means that the focus in such training is on learning and mastering the steps, rather than the craftmanship.

1.4 Outline of the thesis

This report first presents the theoretical frameworks on which the master's thesis is based upon, in chapter 2. Then, chapter 3 details the methodology that was used throughout the project. Chapter 4 highlights the results that were obtained to answer the research questions and chapter 5 discusses them while also providing recommendations concerning future work. Finally, a conclusion is drawn in chapter 6.

2. Theoretical frameworks

In this chapter, the theoretical framework is presented. It begins with some background on product development processes. Then, it highlights the manufacturing at VCC and the current process for operator training. Then follows a description of different applications of VR as of today, its history and characteristics as well as barriers with the use of VR in the industry and its implementation. Finally, databases mapping and how VCC handles data are described.

2.1 Product development process and methodology

Product development is a very broad and important topic in the manufacturing industry. It can be broken down into several steps. The more traditional way to develop a product is described in this section. Furthermore, some parts of this process are used in this project, therefore, they are described in more detail in what follows.

2.1.1 The product development process

Product development is a process that can vary depending on the organization and the nature of what is being developed. According to Wheelwright et al. (1992), the phases that constitute this process are the following ones: knowledge acquisition, concept investigation, basic design, prototyping, pilot production and manufacturing ramp-up. This is an old source, but the process is still similar these days as can be seen in the Figure 2.1 below.

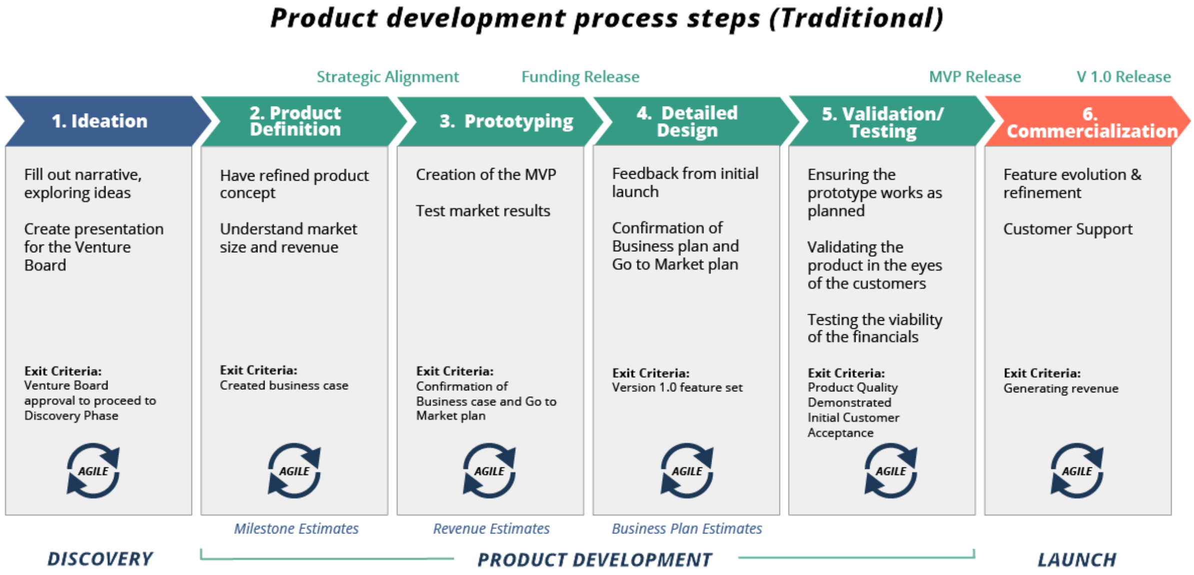

Product development process steps (Traditional)
Strategic Alignment
Funding Release
MVP Release
V 1.0 Release
1. Ideation
2. Product Definition
3. Prototyping
4. Detailed Design
5. Validation/ Testing
6. Commercialization
Fill out narrative, exploring ideas
Create presentation for the Venture Board
Exit Criteria: Venture Board approval to proceed to Discovery Phase
Have refined product concept
Understand market size and revenue
Exit Criteria: Created business case
Creation of the MVP
Test market results
Exit Criteria: Confirmation of Business case and Go to Market plan
Feedback from initial launch
Confirmation of Business plan and Go to Market plan
Exit Criteria: Version 1.0 feature set
Ensuring the prototype works as planned
Validating the product in the eyes of the customers
Testing the viability of the financials
Exit Criteria: Product Quality Demonstrated Initial Customer Acceptance
Feature evolution & refinement
Customer Support
Exit Criteria: Generating revenue
AGILE
Milestone Estimates
Revenue Estimates
Business Plan Estimates
DISCOVERY
PRODUCT DEVELOPMENT
LAUNCH

Figure 2.1: Traditional product development process (Carter, 2024)

2.1.2 Problem solving process

At the heart of the second phase, namely the concept investigation lies the problem-solving process, which generally consists of five steps, see Figure 2.2 below. The first step is to formulate the problem that needs to be solved. The next step is to, from the answers from the first step, translate it to define the criteria that are measurable. Thereafter, search for possible solutions is performed. The goal in this step is to have as many solutions as possible to explore the whole design space. Then, by evaluating and eliminating the defined solutions, the remaining one is the one that is considered as the best or more suitable solution. Finally, in the last step, the solution is further developed (Johannesson et al., 2013).

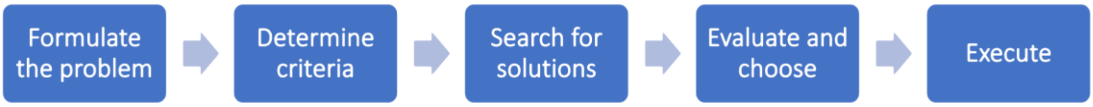


Figure 2.2: Steps used in the product development processes at companies (translated from Swedish to English) (Johanesson et al., 2013)

According to Ulrich et al. (2019), the phase Select Product Concept(s), see Figure 2.3, is the process where the defined concepts are evaluated against each other based on the identified criteria, comparing its strengths and weaknesses and further develop the remaining concept(s). Furthermore, for selecting the solution, two methods are suggested, that all solutions must go through, where the first stage is the concept screening (Pugh matrix) and the second stage, concept scoring (Kesselring matrix).

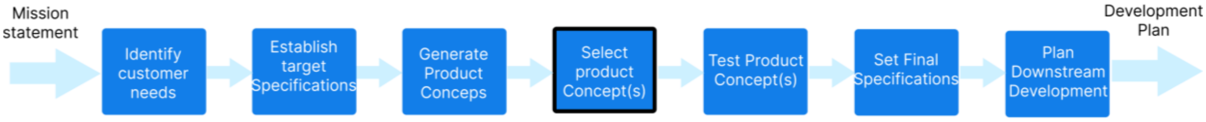


Figure 2.3: Steps used in the product development processes, where concept selection is considered throughout the whole process (Ulrich et al., 2019)

Ulrich et al. (2019) highlight that the benefits with the Pugh matrix and Kesselring matrix is that the solutions / concepts are evaluated against the defined criteria, where the selected concept is more focused toward the customer. In addition, it enables for an objective evaluation of the concepts based on the criteria, which minimizes the risk that personal and arbitrary factors could influence the decision making.

2.2 VCC manufacturing

VCC manufactures several car models which are organized in several categories, that, as of today are electric cars, hybrid cars and mild hybrids (Volvo Cars, 2024). Each model is also manufactured in different variants, depending on if the car is right-hand drive or left-hand drive as well as on other options that each car may or may not be equipped with.

When it comes to the manufacturing itself of each car model, the plants are organized in main sections, that together form the production line of the model. These sections are then geographically divided into stations and the different manufacturing steps that need to be performed to produce the car are divided into sequences. Typically, there is one sequence per station and one group of workers or operators per station. Furthermore, there is one station per car model, meaning that different car variants of the same model can be manufactured in the same stations. Finally, each sequence is broken down into several Work Instructions (WIs), that contain instructions for each and every small manufacturing activity to be performed in that sequence.

2.3 Assembly operator training

Operator training is nothing new. It has been required and therefore delivered ever since the complexity of manufacturing processes had increase. With the current trends of increase in automation and plant optimization, it is nowadays more needed than ever (Patle et al., 2018). Operator training is delivered for many purposes. On the one hand, it is to increase safety in the plant environment and on the other hand, it is to ensure that productivity requirements are met (Holuša et al., 2023). When it comes to the resources utilized to organize the training activities, operators need to be on site to be trained and the training might necessitate to have parts of the products for explanation purposes (Holuša et al., 2023). Training traditionally is done in several steps, based on a combination of classroom training, face-to-face on-site training with experienced operators and utilization of computer-based simulators (Patle et al., 2018).

At VCC, there are two main types of assembly operator training. The first one is training for operators that are completely new to the company, plant and even assembly work. In that case, they need to learn about the company, the operator craftmanship and the assembly sequences. The second type is when the training is provided to already experienced operators, when they work on a new line or when there are modifications in the sequence, they are taking part into, because of final product modifications. In that second scenario, the focus of the training is on the assembly sequence as the craftmanship is already mastered. For newcomers, the first step is an introductory week, during which explanations about the company are provided, together with information on the car models and tools that they will be using in the manufacturing plant. The next step is a 2-day off-the-line pre-training, with a focus on the specific parts of the assembly sequence the workers will be assigned to. That part of the training is paper based, and it is made for the newcomers to understand how and why things are done the way they are. The final step consists of practical on-the-line training and takes usually around 2 weeks. For that period, one trainee has one supervisor (a more experienced operator) and starts by just looking and then doing the assembly work. In the case of an individual already working as operator and starting in a new line or station, the first step is skipped and the final one might be shortened, as the usage of tools, craftmanship and the way of working are already known by the trainee. Although, it depends on the people and their abilities and in general the training process is adapted to fit as well as possible to the trainees (Valic A. & Falk M., personal communication, March 20, 2024).

2.4 XR – VR definition

Extended Reality (XR) is a generic term that regroups many technologies: both Augmented Reality (AR), Virtual Reality (VR), as well as Augmented Virtuality (AV). Mixed Reality is also used sometimes when talking about either AR or AV (Rauschnabel et al., 2022). All these terms and their relationships are presented in Figure 2.4 below.

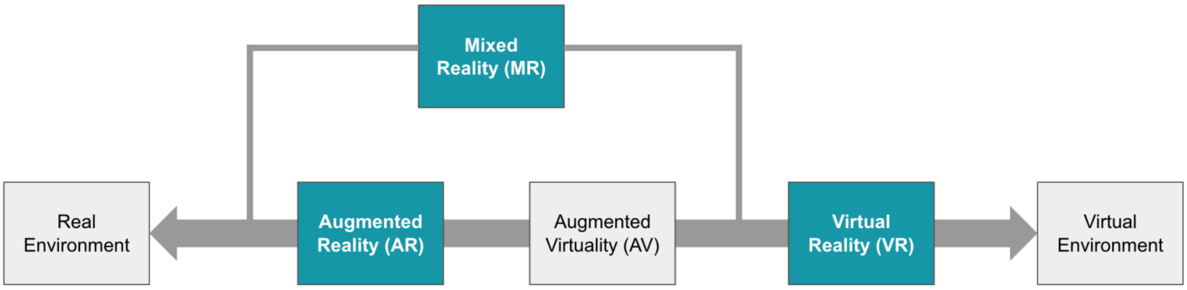


Figure 2.4: Reality – Virtuality continuum (Doolani et al., 2020)

VR, also known as Immersive Computing Technology, is a technology that recently spread and is used to visualize and interact with data of all kinds. The user can experience a virtual world with up to 3 of the 5 human senses: sight, hearing and touch (Berg et al., 2016). In practice, VR works with glasses, that show stereoscopic images on their lenses, which, when combined, allow the human brain to generate images with a relief (Healthy Mind, 2021). In the case of the HTC Vive Pro, which is the hardware solution mostly used at VCC, the full equipment kit consists of the components presented in Figure 2.5 below, and described in what follows. First, a HMD (Head Mounted Display) (also known as VR headset) conveys both the image, directly in front of the user's eyes, as well as the sound, directly to his / her ears. There are also haptic controllers, which allow the interaction with objects in the virtual environment. And finally, there is a base station, whose goal is to help tracking the location of the headset and the controllers for a more precise immersion (Vive, n.d.). Using a base station improve the tracking abilities, which can be handling up to 6 degrees of freedom (rotation and translation tracking), instead of 3 (only rotational tracking), so as walking in the real world implies also walking in the virtual environment (Rauschnabel et al., 2022).

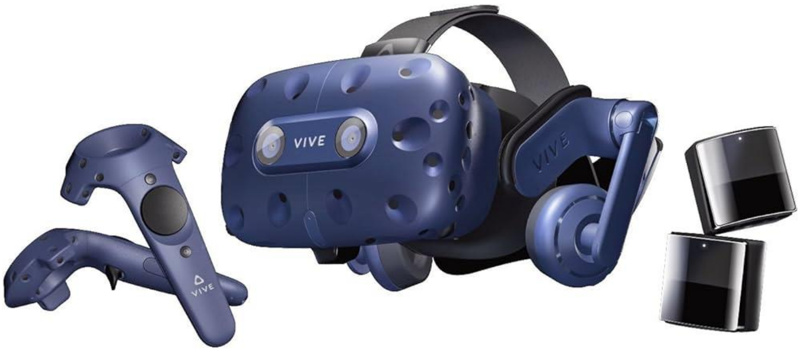


Figure 2.5: Vive Pro 2 full kit (Amazon, n.d.)

As of now, VR technologies are used more generally for gaming purposes, indeed, thanks to VR possibilities, gamers can be immersed into a fantasy world, with a stronger feeling of being into the game, compared to what is possible with regular gaming devices (Voštinár et al., 2021). Such technologies are also used in the medical field, for the treatment of psychiatric disorders, anxiety (Maples-Keller et al., 2017) or phobia. For the latter, it is realized by exposing the patient to a tailored 3D immersive environment, to provide a treatment with the exposure method (Voštinár et al., 2021). Finally, VR is also used in the industry, and this is further developed in the sub-section that follows.

2.5 VR in the industry

This section presents the upbringing of VR as well as the evolution and use cases of this technology in the industry before focusing on VR employee training and finally, pedagogy of such training is discussed in the last sub-section.

2.5.1 History of VR in the industry

In 1990, the "VR Lab" was created at the NASA with the goal of training astronauts for high-dexterity repair tasks on the Hubble Space Telescope. It was deemed very promising because it allowed to visualize the telescope entirely, in a realistic scale, which was previously not possible considering its size (Garcia et al., 2020). Figure 2.6 below presents one of the first training session, where an astronaut is wearing VR glasses and gloves. This was the first time VR was developed and used and it represents the inception of the technology.

Figure 2.6: One of the first VR astronaut training session (Garcia et al., 2020)

Nowadays in the industry, VR is mostly used as a tool for visualization of different types of assets. It allows to have more of a feeling of being close to the data that is displayed and can therefore provide people with a better understanding. For instance, it gives a better sense of the actual aesthetic of a product, presented as a 3D model (Berg et al., 2016). Furthermore, it is a good tool for communication, specifically when it comes to explaining and presenting complex data to people that are not proficient in the domain at stake (Berg et al., 2016). Therefore, it benefits the engineering work for all types of products, but it also benefits the final users of the products as it helps working on the user experience. In fact, with VR, it is possible to play the role of the user at an early stage of the development of a product and thus to focus, early on, on areas such as ergonomics and reachability. This is for instance done by PSA Peugeot Citroën, that uses VR to improve the placement of controls for a better driving experience with their cars (Berg et al., 2016).

2.5.2 VR training

Another use case for VR is employee training, similarly to what it was originally created for at the NASA (Garcia et al., 2020). In that case, it is used as an immersive digital simulation tool, to recreate real-life workplace scenarios (Belova, 2023). It is something that is still new, however, big players have already started to investigate it, for instance, in the automotive industry, BMW, Audi and Volkswagen are doing safety or assembly training, see the Figure 2.7 below for an example of assembly training in VR (VR Owl, n.d.). Employee training with VR makes it possible for employees to gain hands-on experience, in a safe and game-like environment, while also being able to do it remotely. Furthermore, using VR is of great value for training situations where hazardous equipment is being used or when the location is not yet existent, for instance in the case of a new factory being built (Belova, 2023). Moreover, VR tools for employee training and education also come with benefits regarding sustainability, when compared to previous ways to provide training. In addition, it can be done remotely, therefore involving no travel requirement for trainees or trainers and it also decreases the need for physical resources as everything is digital (Holuša et al., 2023).

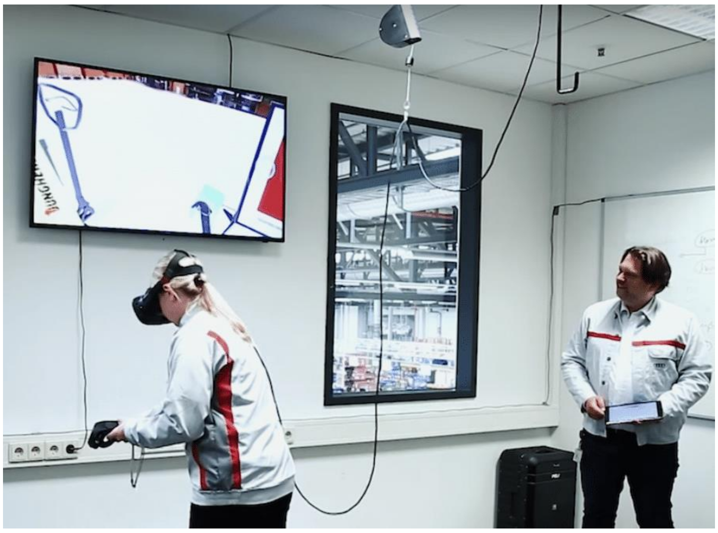


Figure 2.7: Assembly VR training at Volkswagen (VR Owl, n.d.)

When it comes to how different competitors, create, and manage their VR training activities, it is quite difficult to get such information. But, based on the picture in Figure 2.7 and similar ones that can be found online, how the trainings are being conducted can be deduced. It appears that usually one employee is being trained, with a supervisor giving advice next to him or her.

2.5.3 Pedagogical aspects of VR training

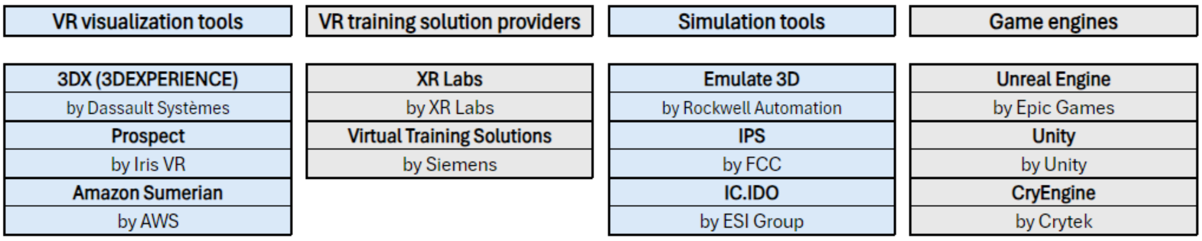
According to Fowler (2015), any type of high-quality training is divided into 3 main phases. First, there is an explanative phase, in which what is to be learned is described with an adapted representation. The second phase is exploratory: the learner should be able to interact with learning materials and immerse himself / herself in the tasks to be performed. The final phase is about testing the understanding of the learner, and this can be done with a tutorial for instance.

When it comes to assembly VR training more specifically, Philippe et al. (2020) identify that important pedagogical aspects are, for instance, descriptive texts of the instructions that should be followed, together with an audio description as well as a highlight of the elements required for a specific task. Furthermore, having several difficulty modes for the training is deemed to be beneficial for a more progressive type of learning. The difference between these modes would be the level of guidance given while performing the tasks, with a lot of them being given for a beginner mode and less for a more advanced one. (Philippe et al., 2020)

2.6 VR software solutions

There are many solutions to create VR training scenes. In the context of this project, 11 solutions were identified by the researchers, that were deemed as the most widely used, well known and promising. They were then grouped into 4 categories, see Table 2.1, where each of the identified solution is presented in its respective category. Such categorization was specifically realized, in-house, to have a better organization and overview of the solutions, their similarities, and differences. More details on each of the categories are provided in the following sub-sections.

Table 2.1: VR solutions in their categories



VR visualization tools	VR training solution providers	Simulation tools	Game engines
3DX (3DEXPERIENCE)	**XR Labs**	**Emulate 3D**	**Unreal Engine**
by Dassault Systèmes	by XR Labs	by Rockwell Automation	by Epic Games
Prospect	**Virtual Training Solutions**	**IPS**	**Unity**
by Iris VR	by Siemens	by FCC	by Unity
Amazon Sumerian		**IC.IDO**	**CryEngine**
by AWS		by ESI Group	by Crytek

Furthermore, for four of these options were conducted more thorough research, thanks to interviews of either engineers using a specific software or people working at the software provider company. This was realized to understand better the possibilities when it comes to VR training environment creation. The results from those interviews are presented in Section 4.1 and the methodology used to conduct them is presented in Section 3.1.

2.6.1 VR visualization tools

This category consists of software that are used mainly for visualization and collaboration in a realistic 3D environment. For instance, 3DX focuses on CAD geometry life-size, stereoscopic visualization, to make revision decisions easier to take (3DEXPERIENCE, n.d.). Prospect uses 3D CAD-files to create realistic environments (Prospect, n.d.), and Amazon Sumerian is used to create 3D scenes and characters more easily (Amazon Sumerian, n.d.). These VR solutions allow navigation in the virtual environment but not much interaction with what populates it is available.

2.6.2 VR training solution providers

VR training solution providers are companies that do not sell a product but sell a service. They do not provide a software that the client is free to use but they directly create the VR training scenes for their customer, based on the identified needs. For instance, Siemens, which is the PLM system's (Product Lifecycle Management) supplier at VCC, has a service of virtual service solutions that tailors VR training, by using, if possible, data from the customer's PLM system (Siemens, n.d.).

2.6.3 Simulation tools

This category is composed of software solutions that, first and foremost, are engineering tools. They are used by manufacturing engineers and integrate VR possibilities for assembly or manufacturing planning. Emulate3D is all about creation and test of innovative plant designs, to improve them, so as such designs can be implemented in the physical factory. The goal is to have a realistic digital factory in which changes can be made before changing the actual factory, so as the interruptions are limited, and the outcomes are predicted in advance (Emulate3D, n.d.). IPS (Industrial Path Solutions) is, at its core, a software for automatic verification of assembly feasibility, also dealing with robot stations and surface treatment processes (FCC, n.d.). It is used at VCC, mostly within Manufacturing, with its path planning options, for simulation of workstation ergonomics and assembly or disassembly sequence pre-studies as well as for design verification, for electrical and mechanical integration within Engineering. IC.IDO allows to realistically experience products in a virtual world. The focus is on assembly, operating and maintenance activities, with the possibilities to use tools and evaluate real-time physics for instance (ESI Group, n.d.).

2.6.4 Game engines

A game engine is a software development environment, for the development of video games, based on one or several programing languages (University of Silicon Valley, 2020). Some of them are tailored for the development of VR games, such as Unity, which is a real-time 3D development engine, made for the creation of immersive and interactive environments (Unity, n.d.). When comparing the game engines to one another, they appear to be quite similar, especially in the workflow they require for game creation. Nevertheless, it can be noted that Unity seems to be more user-friendly than Unreal Engine and CryEngine because of its interface, coding language and the availability of many plug-ins to tailor the development experience (Isar, 2018; Mohd et al., 2020).

2.7 Barriers for VR training

VR and more generally XR technologies are difficult to implement, and it is even more the case when trying to do it in large scale companies or in old industries. For instance, in their study on the construction industry in the UK, Badamasi et al. (2022) highlighted several barriers to VR adoption. The main ones were, by order of importance, the lack of in-house expertise in such new technologies, then the cultural change it represents, followed by its cost and finally the application development complexity. The barrier resulting from the cultural aspect is caused by the workers' mindset and preferences, as they are reluctant to change their methods. Sobhani et al. (2017) and Lin et al. (2020) highlighted that to create immersive virtual environments, several factors need to be taken into consideration, such as time-consumption and computational algorithms. In addition, according to them, another aspect to consider is the cost. To create a realistic environment, a great quantity of resources is needed and Schwewbel et al. (2008) point out that it can be challenging to implement, and it can be costly. Therefore, according to Schönfelder (2003), one can overcome the barriers through more costly software and complexity of the hardware but then, the cost-related issues are even more prevalent. Sagnier et al. (2020) discuss that even if there are benefits with implementing VR for visualization and educational purposes, there are some barriers that have been identified. One of the barriers is the motion sickness that the user experiences, which can result in preventing social acceptance. Another barrier is the ethical aspects that includes the procedure of data privacy and surveillance. Cameiro et al. (2023) mention that other potential barriers such as psychological aspects should be considered as they are related to mental health and establish a code of conduct, accountability, and consequences with the use of VR.

When a study on VR implementation was realized at VCC in the manufacturing division, by Boström and Zamola (2023), similar challenges emerged. Limitations were faced because of limited computing power, because of non-adapted hardware, which would need to be fixed with investments in more powerful computers. Another issue was that not much knowledge on VR simulation was available at the company beforehand, which resulted in a lot of time and effort needed to actually get started before being able to produce something valuable. Time itself was another challenge as the creation of a single VR training scene took several hours. Finally, despite being very promising, the use of VR tools for operator training was also not intuitive enough for the trainees themselves, which would require more focus on the pedagogical aspects of it. It would be helpful, for easier implementation of VR training, that the trainees had previous VR experience, but it is not very likely to be the case. Furthermore, the consumer adoption of HMDs (Head Mounted Display) is quite low, especially when compared to other older technologies that were as disruptive when they emerged (such as computers, mobile phones, etc.). This is because the development rate is quite low as well and it represents a large behavioral change in the purchasers' habits (Doolani et al., 2020).

2.8 Databases handling

The industry 4.0 puts a strong focus on data management, to take advantage of competitive leverages coming from virtualization and decentralization to name but a few (Raptis et al., 2019). Furthermore, creating VR training scenes requires different types of data and doing so automatically requires a very profound understanding of the databases organization. This section presents the systems used to organize and manage data, namely Product Lifecycle Management (PLM) systems as well as the core organization of the databases.

2.8.1 Product Lifecycle Management systems

PLM is the management of all the data related to a product, throughout its lifecycle, i.e., from its design to its disposal. PLM systems are the software solutions that are implemented to fulfill these strategic needs (Hayat et al., 2022). The main tasks of such systems are data sorting, the definition of its status (supervision and certification levels) as well as the definition of the connections between data cells. They also foster collaboration between teams working on the same product but from different perspectives and who may have conflicting objectives (Niemann et al., 2021).

One challenge that emerges with PLM systems is the interoperability and data exchange between each software used throughout the product lifecycle as each one deals with different data types. For that matter, one approach is to use standards to facilitate data exchange and to use APIs (Application Programming Interface) to help convert files when required, for compatibility (Penciuc, 2014).

2.8.2 Data mapping

Usually in big companies, such as VCC, engineering data is stored in several graph-based databases, see Figure 2.8 below for an example of a tree graph-based database. The thing is that there are several databases, for the different types of data that are created and used by the company (Segeborn, personal communication, March 12, 2024). Therefore, it gets more difficult when wanting to use related data from different databases.

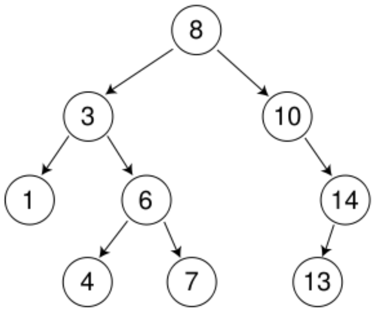


Figure 2.8: Structure of a tree graph-based database (Stack overflow, 2014)

Eiden et al. (2021) advocate that to map data from heterogeneous systems, links inside each system and across systems boundaries as well as dependencies between data objects need to be created. All these then need to be unified with a metadata model. The latter can be created in different ways. It can be done manually, or by computer programs that compare attributes of the concerned data objects or with machine learning which would suggest the mapping itself. One key aspect when implementing such a project is the knowledge representation, which should allow stakeholders from different departments to collaborate seamlessly.

3. Methodology

The methodology implemented for this thesis is covered in this chapter. An overview of the methodology is presented in Figure 3.1. As can be seen, the first step was to conduct a qualitative pre-study to acquire knowledge about existing software solutions in the market, used currently, both by engineers, in the automotive industry and by game designers, in the gaming industry. Then, interviews were utilized to identify the capabilities and possibilities with the different software solutions. After that, a requirement specification, analysis of the stakeholders, and evaluation matrices (Pugh and Kesselring matrices) were employed, where the result was a chosen software to proceed with. The next step was to get proficient with the chosen software and to develop, first, a relevant VR training, along with a quantitative study on the prerequisites of PLM data for the automatic generation. In the last step, a validation of the study was conducted, where the VR training scene that had been created was evaluated, as well as its creation process. The results from that and from the PLM data mapping were connected to the research questions, to define the possibilities and the practicalities with the scalability of VR training and future recommendations.

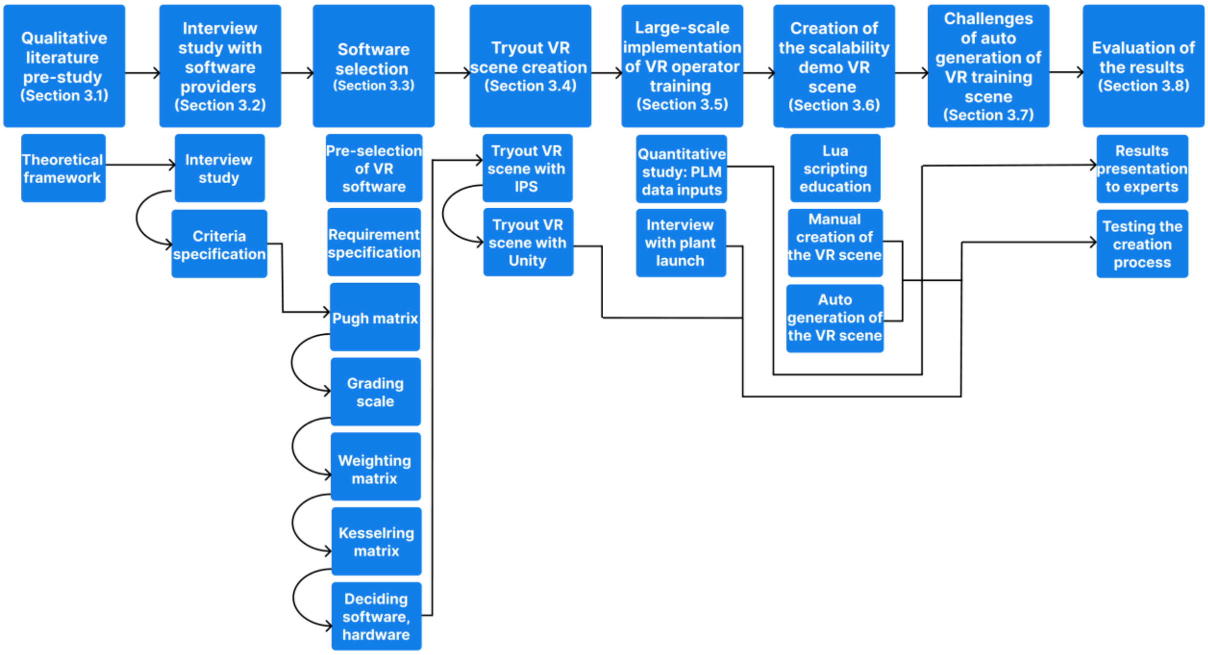

Qualitative literature pre-study (Section 3.1)
Interview study with software providers (Section 3.2)
Software selection (Section 3.3)
Tryout VR scene creation (Section 3.4)
Large-scale implementation of VR operator training (Section 3.5)
Creation of the scalability demo VR scene (Section 3.6)
Challenges of auto generation of VR training scene (Section 3.7)
Evaluation of the results (Section 3.8)
Theoretical framework
Interview study
Criteria specification
Pre-selection of VR software
Requirement specification
Pugh matrix
Grading scale
Weighting matrix
Kesselring matrix
Deciding software, hardware
Tryout VR scene with IPS
Tryout VR scene with Unity
Quantitative study: PLM data inputs
Interview with plant launch
Lua scripting education
Manual creation of the VR scene
Auto generation of the VR scene
Results presentation to experts
Testing the creation process

Figure 3.1: Methodology on the process to achieve the results

3.1 Qualitative literature pre-study

The very first step of this thesis was a literature study, to gain more knowledge on the different topics that constitute the core of the project. This is presented in Chapter 2. The topics covered are mainly within VR, its use in the industry and for training, as well as training more generally, product development and databases organization. According to Synder et al. (2019), to acquire more knowledge about a research area, the most suitable literature methodology is to combine a semi-systematic literature review with an integrative literature review. By doing so, the researchers can explore and cover different areas and research within these areas. Therefore, such a literature review was conducted. The collected information and knowledge was extracted from different databases, such as Chalmers Library, Scopus, and Elsevier, together with research papers from other sources and books.

3.2 Interview study with software providers

To gain more knowledge about the software solutions used in the market, an analysis of the relevant stakeholders was conducted (Bell et al., 2019). As a result, this gave the researchers more knowledge about the software that would be applicable in assembly training and how different non-engineering software could also be used for the same application and area. In addition, by contacting the companies and discussing their software, the researchers gained more insight and knowledge of their solutions. Furthermore, by visiting the assembly line at the plant, more knowledge was acquired about the different cases for which it would be beneficial to create VR scenes to train operators.

A qualitative method was implemented to acquire a deeper understanding of the topic based on the participant's perspectives. The participants / interviewees were identified using the snowball sampling (Dawson, 2007; Denscombe, 2014). This signifies that contacts provided by Volvo Cars and Chalmers University of Technology were used to meet with participants that had great knowledge about the different software solutions.

From the snowball sampling, semi-structured interviews were conducted. Using semi-structured interviews, the topics identified in the interview guide (presented in Appendix A) were covered and the questions established were asked in a chronological order. By gathering specific information about the topic, the interviewees' results were compared. In addition, follow-up questions were asked, depending on the interviewees' responds to the questions (Bell et al., 2019; Denscombe, 2014). The follow-up questions resulted in more dynamic interviews, that led to the gathering of more specific answers from the interviewees (Dawson, 2007; Denscombe, 2014).

The interviews were conducted via both Microsoft Teams and on-site, at predetermined locations, and the interviews lasted up to one hour. The first set of questions, see in Appendix A, namely "general thoughts" was created to acquire knowledge about the possibilities and capabilities with the interviewees' software. The next set of questions were aimed to explore the integration possibilities between PLM systems and the software, with the goal to gain more knowledge on whether it has been implemented or not and how far in the development process it had reached. Furthermore, in the last set of questions, "SDK and scripting", the aim was to explore the possibilities on how to set up the VR scene in the software as quickly as possible as well as the possibility of auto generation for some steps of the VR scene creation. The interviews were transcribed to mitigate critical information that could be missed from the interviewee's sayings. For the meeting that was conducted at a predetermined place, notes were taken based on the interviewee's presentation materials and answers.

3.3 Software selection

To select the software that would be used throughout the project for the development of VR operator training scenes, the different solutions, used in the industry were explored. As presented in Section 2.6, eleven solutions were first identified. They were the main solutions in that field at that time, the most documented and they included the ones that were recommended by VCC and Chalmers.

3.3.1 Pre-selection of VR software

Out of the four categories presented in Table 2.1, two were selected to go through the actual selection process. Furthermore, the VR training solution providers were deemed as not eligible, since with these options, the selected company would create the VR training scenes on their own and no knowledge would be gained by VCC in the end. Similarly, the VR visualization tools were also not considered as valid options since they were not meant for the user to interact with the virtual environment as much as needed, but just to observe it in a virtual world. Therefore, at this stage, 6 potential solutions were possible, these were composed of the game engines and the simulation tools. This was also justified by the fact that, from the literature review, software from both categories had been used in similar contexts and the outcomes were satisfactory.

3.3.2 Defining the requirement specification (selection criteria)

To evaluate the VR software solutions, a requirement specification was implemented. It was established from the literature review and the needs mentioned by the VCC supervisors. As a result, it mapped the essential selection criteria that should be met when wanting to create VR operator training scenes.

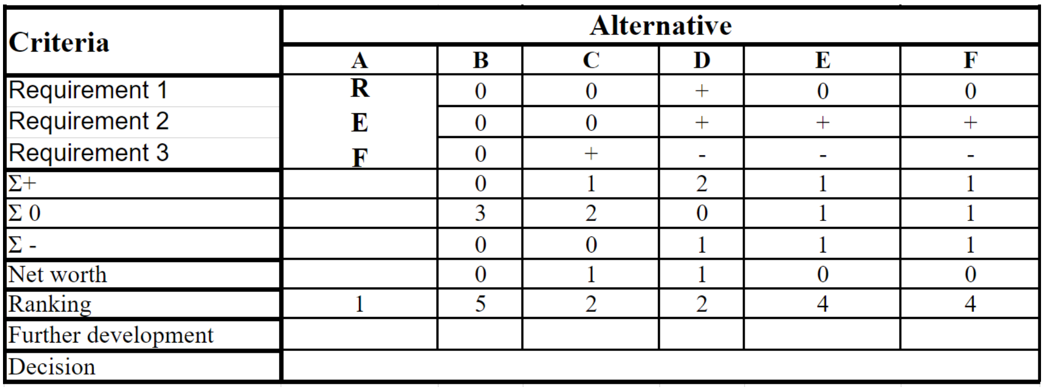
According to Ulrich et al. (2019), the least biased and more structured way to evaluate the defined solutions is to use decision-matrices (Pugh and Kesselring matrices), to narrow them down so that the remaining solution will be considered the most suitable one.

3.3.3 Pugh matrix used for the evaluation

Once the requirement specification has been defined (presented in Table 4.2), it was put into the Pugh matrix and was used as criteria for evaluating the VR software solutions. The purpose of the Pugh matrix is, with a solution defined as a reference (often the one that is the most familiar), to compare it with the other solutions for each criterion, see Table 3.1. For example, for the first criterion (Requirement 1) the solution B was considered inferior to the reference (A) and was graded a "- "(minus). On the other hand, if it was considered better than the reference solution, it would be graded a "+" (plus) and "0" if it was considered as equal compared to the reference solution.

In this thesis, two iterations of the Pugh matrices were conducted to eliminate any bias in the results and to ensure that the results converged.

Table 3.1: Pugh matrix (A as the reference solution)



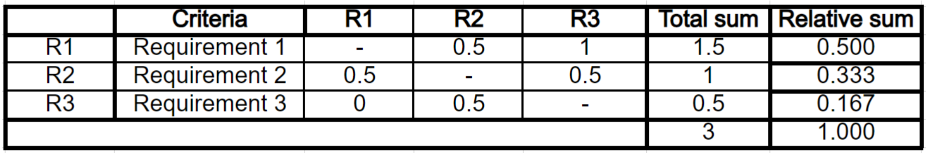
Criteria	Alternative					
	A	B	C	D	E	F
Requirement 1	R	0	0	+	0	0
Requirement 2	E	0	0	+	+	+
Requirement 3	F	0	+	-	-	-
Σ+		0	1	2	1	1
Σ 0		3	2	0	1	1
Σ -		0	0	1	1	1
Net worth		0	1	1	0	0
Ranking	1	5	2	2	4	4
Further development						
Decision						

3.3.4 Kesselring matrix used for the evaluation and ranking

For the Kesselring matrix, the ranking of the solutions is dependent on the weighting factors (defined as the w-values) and the v-grades, that are based on the score for each criterion.

A weighting matrix was implemented to evaluate the criteria's relative importance, see Table 3.2. When the comparison was performed, it was executed by comparing the criterion on each row, with the ones in each column. If the criterion was considered more important than the compared criteria (column), it was graded a 1. A 0.5 was given if it was considered as important as the compared ones, and a 0 if the criterion was less important than the criteria it was compared to. Furthermore, the relative sum was calculated by dividing the total sum for the criterion with the total sum for all criteria.

Table 3.2: Weighting matrix/pairwise comparison of the criteria (requirements)

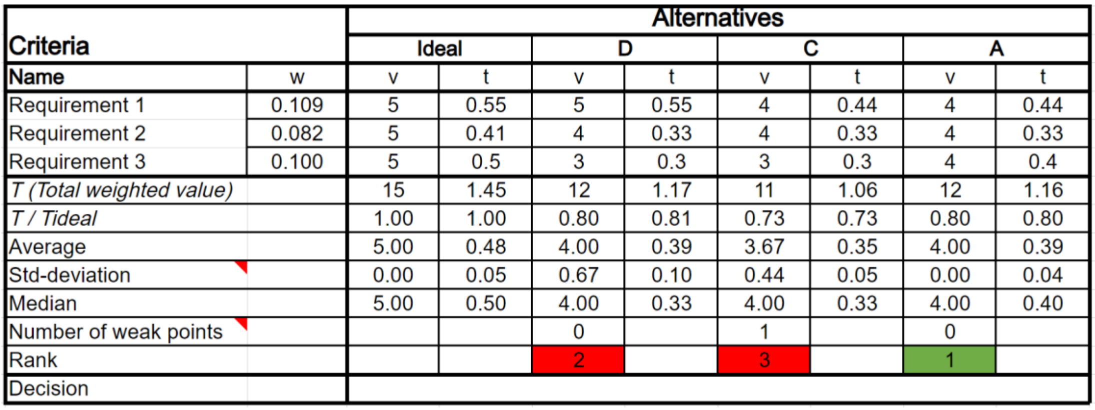


	Criteria	R1	R2	R3	Total sum	Relative sum
R1	Requirement 1	-	0.5	1	1.5	0.500
R2	Requirement 2	0.5	-	0.5	1	0.333
R3	Requirement 3	0	0.5	-	0.5	0.167
					3	1.000

A grading scale from 1-5 was established and was utilized to grade the fulfillment of each solution for each criterion. See Appendix B.2 for the grading scale. Then, by using the weighting matrix and the grading scale, the Kesselring matrix scoring was implemented.

The total value, t, was calculated by multiplying "w" with the "v" for each criterion. After the scoring, the scores were summarized and ranked and the solution with the highest score was considered as the most suitable one, see Table 3.3.

Table 3.3: Kesselring matrix



Criteria		Alternatives							
		Ideal		D		C		A	
Name	w	v	t	v	t	v	t	v	t
Requirement 1	0.109	5	0.55	5	0.55	4	0.44	4	0.44
Requirement 2	0.082	5	0.41	4	0.33	4	0.33	4	0.33
Requirement 3	0.100	5	0.5	3	0.3	3	0.3	4	0.4
T (Total weighted value)		15	1.45	12	1.17	11	1.06	12	1.16
T / Tideal		1.00	1.00	0.80	0.81	0.73	0.73	0.80	0.80
Average		5.00	0.48	4.00	0.39	3.67	0.35	4.00	0.39
Std-deviation		0.00	0.05	0.67	0.10	0.44	0.05	0.00	0.04
Median		5.00	0.50	4.00	0.33	4.00	0.33	4.00	0.40
Number of weak points				0		1		0	
Rank				2		3		1	
Decision									

3.4 Tryout VR scene creation

Following the selection of the software, a first simple VR training scene creation case was realized. This step was deemed necessary for several reasons. First, creating the scene with IPS was a good way to get used to working on that software for the purpose of VR training and it was a way to identify the barriers faced for the automation of the creation. Moreover, creating the scene with Unity was useful to identify more easily which pedagogical aspects were the most relevant ones to achieve a high-quality training.

The case itself consisted of the setup of a subwoofer in the trunk of a car. To do so, the operator need to pick it up and go and place it in the trunk.

3.4.1 Tryout VR scene with Unity

To create the VR scene with Unity, the first step was to import the CAD files into the scene. To do so, Pixyz Studio was used to convert the CAD files that were taken from Teamcenter into fbx 3D objects, which is a format that Unity can read. Then, for the creation of the sequence itself, with the interaction possibilities and the guidance elements, the free Unity asset VR Builder (Unity Asset Store, n.d.) was used, together with a tutorial on how to create a VR game with that asset (Valem Tutorials, 2022).

3.4.2 Tryout VR scene with IPS

Before creating the first case in IPS, training materials, provided by VCC, FCC and Chalmers were used to get proficient with the software. Furthermore, Puranjay, domain lead simulation engineer, was of great help to quickly understand the workflow for this kind of scene set up. He was able to help efficiently with profusion of tips and recommendations, which represented a significant gain of time. The purpose of the training material was to get familiar with the software and to establish the workflow on how to create a scene, create motions and sequences. Hence, it was then easier to move towards the scripting and automation of such workflow. Furthermore, working on the tryout VR scene with IPS also enabled to see what the limitations with this software were, concerning pedagogical elements and types of parts behaviors, together with what was more or less difficult.

3.5 Large-scale implementation of VR operator training

First, to start envisioning the large-scale implementation of VR operator training, a step back was necessary. It was deemed useful to replace it in the context of product development to which it belongs to, to set a scope on the data that would or would not be available. This was done by conducting an interview with Johan Segeborn, PLM expert at VCC. The interview was unstructured so as the interviewee was able to speak as freely as possible on the desired topic (Denscombe, 2014).

To gain knowledge on the place that VR operator training should hold in the operator training process, a semi-structured interview was realized with two persons, one plant launch manager and one launch specialist. The beforehand identified topics where covered and the questions established were asked in a chronological order. See Appendix C.6 for the transcript. Based on the inputs from the interviewees' responds, a summary as a table was created. This, together with the tests conducted with the tryout VR scene creation, allowed to confirm what a satisfying VR training scene should consist of, and the data required to create it.

Once the data required identification was accomplished, it was necessary to determine how the identified data can be gathered and how to do that automatically. That required to get to know how these different types of data were stored and managed, at that time, at VCC. To do so, several meetings were organized with Johan Segeborn, to first, get an overview of the data organization within the Manufacturing Engineering department at VCC and later, to dive deeper in the data directly relevant for VR training and their availability. Those were one-to-one, unstructured interviews. This type of interview was deemed as being the most adapted since few previous knowledge was available on the topic.

As a final step, another meeting was organized to present and get feedback on the findings and ideas that could be implemented in the future to modify the data storages to ease the data access process.

3.6 Creation of the scalability demo VR scene

After discussion, the case that was to be developed as a concrete example to study the scalability of VR operator training, was decided to be a part of the High Voltage (HV) Battery & Charging station, for Volvo V60. The reason why it was chosen was because it was identified as a critical sequence by VCC plant launch personnels. Also, it consisted of different types of parts, including flexible cables, which allow to determine the possibilities within IPS with and without scripting for such elements. Furthermore, determining which case would be used was not a crucial decision as it was just used as an example, to both create scripts for the automation, that could be used for other VR training scene creation and to compare the creation processes without and with scripting. The methodologies used to work on these two creation processes are described in Sub-sections 3.6.2 and Sub-section 3.6.3, starting by the one without scripting. Before that, the methodology used to get proficient in Lua scripting for IPS is presented in the following sub-section.

3.6.1 Lua scripting education

FCC provided training materials which had the purpose of getting familiar with the Lua API (Application Programing Interface) that IPS uses when it comes to scripting, as well as acquiring the mindset on how to create lines of codes (scripts) in a structured manner. The training included how to create objects (primitive shapes), manipulation of tree objects, and loop creation, to make it less computationally heavy. After the study of the training materials, a meeting, in a classroom setting was organized, with Jonas Kressin, Product Area Lead at IPS AB, so all the emerging and remaining questions could be answered.

More practical knowledge and know-how, more closely related to the specific use of the scripting was also acquired with a learning by doing method, with countless trials and errors. The "API for Lua scripting in IPS 2023-R2" manual, that is available from the software itself was used a lot to get to know the possibilities with Lua scripting in IPS and how to implement them and turned out to be essential. Furthermore, all along the scripting journey, people from FCC, were very supportive and very helpful whenever issues were faced.

3.6.2 Manual creation of the scalability demo VR scene

The creation of the scene began with getting the correct CAD assembly and fixture (.jt) files from Teamcenter and importing them into IPS. The storage furniture for the parts and the screws that were used in the plant in that station were not available in Teamcenter, therefore, it was created manually using primitive shapes. The knowledge acquired from the training was implemented throughout the whole process. In addition, this case involved flexible cables that included objects which needed to follow the same movement as the cable. Therefore, further education of simulating the behavior of the cables interacting with other parts was implemented with both VCC and IPS.

Once the whole training scene was created, it was realized during the testing of the case in VR that for three of the cables that needed to be flexible in relation to other rigid bodies, while IPS only allowed grabbing two parts simultaneously, the cables and rigid bodies in total accounted to be seven, meaning that all seven parts should be grabbed at the same time. This was not possible, therefore, simplification of the three cables interacting with other defined rigid bodies was necessary. The rigid portions that the operator would grab onto were defined as parents and the nodes connected to the cable were children. In contrast, the rigid bodies (clips) that were on the cable were made to be rigid bodies and connected to a grip with a node and were the children, while the node (that were flexible) were the parents. Once the simplifications were made, the motions and sequences were created. Then, the snapping zones were defined, with minor modifications of where the part would snap and the snapping distance.

3.6.3 Scripting for the scalability demo VR scene creation

The goal with the scripting was to automate as much as possible of the workflow to create a VR training scene for this specific case that was also applicable for any other cases. Therefore, it needed to be adaptable, flexible, and reusable. The time spent to create the scripts themselves was quite important, especially since it required first to get proficient at scripting with Lua for IPS. Nevertheless, the time spent for that was not considered when evaluating the time required to use them to create a VR training scene as the scripts, once created, could be reused.

At first, the objective was to create one script that would auto generate everything on its own with as few inputs as possible from the user. Nevertheless, it appeared that because of some current limitations, this was not possible. Therefore, the first step was to identify what was possible or not to auto generate, with that goal to do it as much as possible. Thus, in the end, four different scripts were created, three of them for the scene creation itself and one that to be used from within VR.

To create the main script, that was to enable the user to interact with the parts in VR, the workflow that had been identified when creating the tryout VR scene (see section 4.2.2) was used. For each of the steps of that workflow, associated lines of code were written to have it done automatically. Therefore, the scripting was, in practice, realized step after step, with checks to see if things were working or not each time.

3.7 Challenges of auto generation of VR training scene

The scalability demo VR scene was created to serve as a demonstrator but also with the intent to identify all the barriers on the way to achieve a large-scale implementation of VR operator training. Therefore, based on the final scene and the activities involved to create it, several challenges could be identified and grouped in categories to be presented in a clear way. Ideas and directions to address these challenges were then pondered upon and are presented in the discussion chapter.

3.8 Evaluation of the results

Once the scripts were completed, a test run to create the VR training scene was conducted by the researchers. In addition, the manual creation of the VR training scene was also developed, and the estimated time of creation was recorded and documented and compared.

Then, to have some insights and to validate it, a demo for a PLM / method developer, a domain lead connected operator, a domain lead simulation engineer and other member of the Manufacturing Digital Team was presented. The goal was to gain feedback on potential improvements of the auto generation aspect regarding the creation of the VR operator training scene.

4. Results

In this chapter, the results, used to answer the research questions are presented, see Figure 4.1 for the organization of the chapter. That figure also indicates which section brings material for which research question. It shall be noted that this chapter is organized similarly to the previous one, which explains the methodology that was used to get the results. By selecting a software, developing the tryout VR scene, and evaluating the results from all created VR scenes together with considering the challenges discovered along the way, the first research question was answered. By mapping the data required to create a VR scene, creating the scalability demo VR scene, and evaluating the results, the second research question could be answered. Finally, the third research question was answered based on the results from the consideration of pedagogical elements for operator training as well as the study of the emerging challenges for the auto generation of VR training scene.

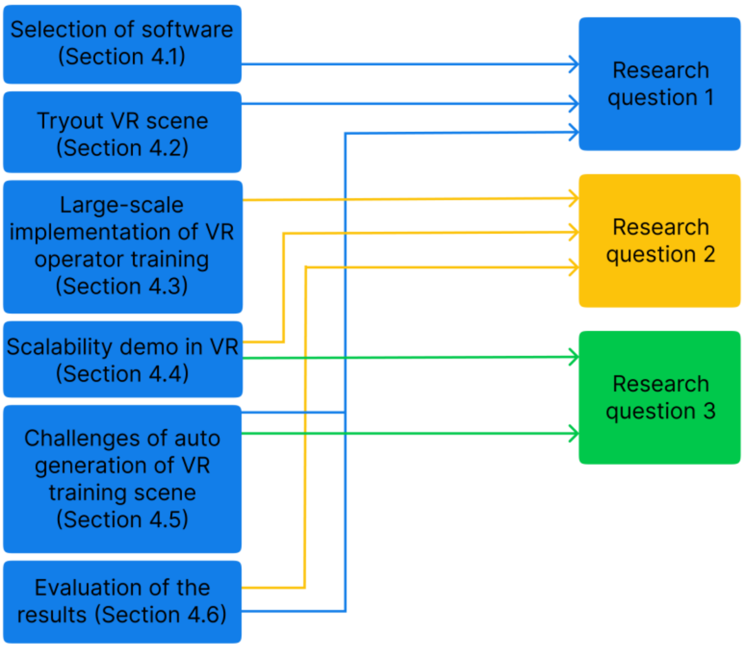

Selection of software (Section 4.1)
Tryout VR scene (Section 4.2)
Large-scale implementation of VR operator training (Section 4.3)
Scalability demo in VR (Section 4.4)
Challenges of auto generation of VR training scene (Section 4.5)
Evaluation of the results (Section 4.6)
Research question 1
Research question 2
Research question 3

Figure 4.1: Illustration of the results used to answer the research questions

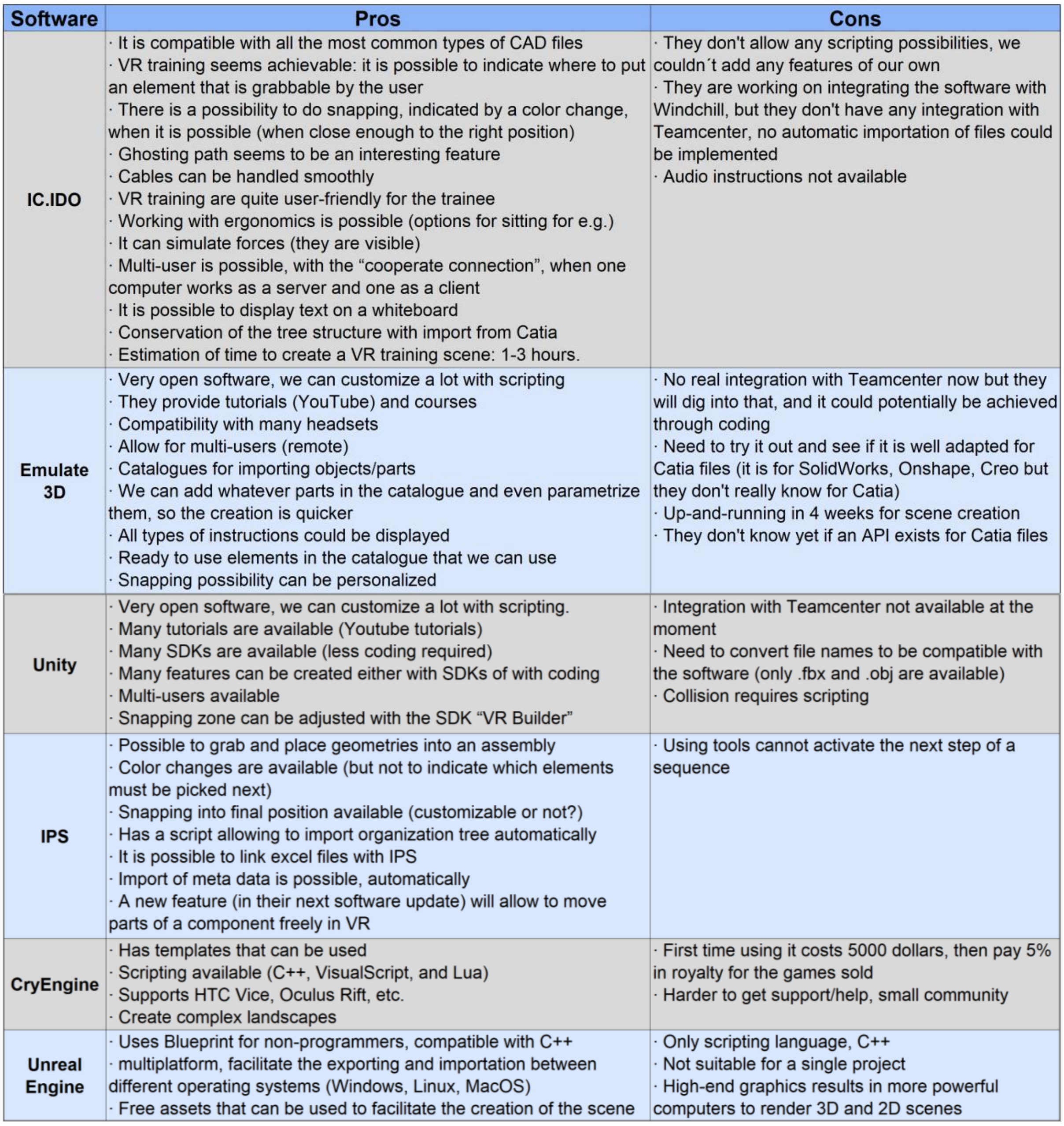
4.1 Selection of software

This chapter presents the results of the selection process for the software to be used to create VR training scenes. As a reminder, it started with collecting qualitative data and conducting interviews and based on that, a criteria specification was established. Then, to select the software, two Pugh matrices and a Kesselring matrix were implemented. The associated results are presented below.

4.1.1 Qualitative results from software interviews

The interviewees' responses were summarized and combined with the qualitative data obtained separately. At this stage, six solutions were still considered. An interview with a software provider was possible to be organized for four of them to complement the qualitative data from the literature review. For the remaining two, literature reviews were the only source of information. The results were then categorized into Pros and Cons. Table 4.1 shows these results for the following software solutions: IC.IDO, Emulate3D, Unity, IPS, CryEngine and Unreal Engine.

Table 4.1: Pros and Cons for each software

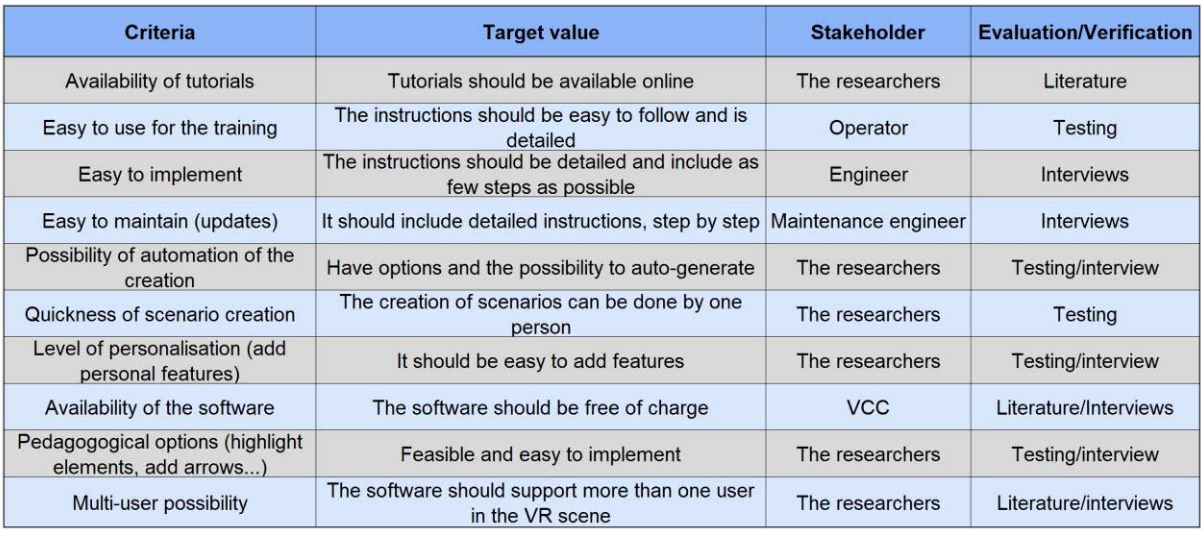


Software	Pros	Cons
IC.IDO	· It is compatible with all the most common types of CAD files · VR training seems achievable: it is possible to indicate where to put an element that is grabbable by the user · There is a possibility to do snapping, indicated by a color change, when it is possible (when close enough to the right position) · Ghosting path seems to be an interesting feature · Cables can be handled smoothly · VR training are quite user-friendly for the trainee · Working with ergonomics is possible (options for sitting for e.g.) · It can simulate forces (they are visible) · Multi-user is possible, with the "cooperate connection", when one computer works as a server and one as a client · It is possible to display text on a whiteboard · Conservation of the tree structure with import from Catia · Estimation of time to create a VR training scene: 1-3 hours.	· They don't allow any scripting possibilities, we couldn´t add any features of our own · They are working on integrating the software with Windchill, but they don't have any integration with Teamcenter, no automatic importation of files could be implemented · Audio instructions not available
Emulate 3D	· Very open software, we can customize a lot with scripting · They provide tutorials (YouTube) and courses · Compatibility with many headsets · Allow for multi-users (remote) · Catalogues for importing objects/parts · We can add whatever parts in the catalogue and even parametrize them, so the creation is quicker · All types of instructions could be displayed · Ready to use elements in the catalogue that we can use · Snapping possibility can be personalized	· No real integration with Teamcenter now but they will dig into that, and it could potentially be achieved through coding · Need to try it out and see if it is well adapted for Catia files (it is for SolidWorks, Onshape, Creo but they don't really know for Catia) · Up-and-running in 4 weeks for scene creation · They don't know yet if an API exists for Catia files
Unity	· Very open software, we can customize a lot with scripting. · Many tutorials are available (Youtube tutorials) · Many SDKs are available (less coding required) · Many features can be created either with SDKs of with coding · Multi-users available · Snapping zone can be adjusted with the SDK "VR Builder"	· Integration with Teamcenter not available at the moment · Need to convert file names to be compatible with the software (only .fbx and .obj are available) · Collision requires scripting
IPS	· Possible to grab and place geometries into an assembly · Color changes are available (but not to indicate which elements must be picked next) · Snapping into final position available (customizable or not?) · Has a script allowing to import organization tree automatically · It is possible to link excel files with IPS · Import of meta data is possible, automatically · A new feature (in their next software update) will allow to move parts of a component freely in VR	· Using tools cannot activate the next step of a sequence
CryEngine	· Has templates that can be used · Scripting available (C++, VisualScript, and Lua) · Supports HTC Vice, Oculus Rift, etc. · Create complex landscapes	· First time using it costs 5000 dollars, then pay 5% in royalty for the games sold · Harder to get support/help, small community
Unreal Engine	· Uses Blueprint for non-programmers, compatible with C++ · multiplatform, facilitate the exporting and importation between different operating systems (Windows, Linux, MacOS) · Free assets that can be used to facilitate the creation of the scene	· Only scripting language, C++ · Not suitable for a single project · High-end graphics results in more powerful computers to render 3D and 2D scenes

4.1.2 Criteria specification

In total, ten criteria were specified which cover the feasibility, implementation, and maintenance for the operator and the engineers, see Table 4.2. The defined criteria connected to the ease to create a VR scene that would facilitate the training for the operators and the possibility to auto generate it. Furthermore, it should also be feasible to personalize it as well as to maintain the VR scene when necessary. For a more thorough justification of the criteria, see Appendix B.1.

Table 4.2: Criteria specification



Criteria	Target value	Stakeholder	Evaluation/Verification
Availability of tutorials	Tutorials should be available online	The researchers	Literature
Easy to use for the training	The instructions should be easy to follow and is detailed	Operator	Testing
Easy to implement	The instructions should be detailed and include as few steps as possible	Engineer	Interviews
Easy to maintain (updates)	It should include detailed instructions, step by step	Maintenance engineer	Interviews
Possibility of automation of the creation	Have options and the possibility to auto-generate	The researchers	Testing/interview
Quickness of scenario creation	The creation of scenarios can be done by one person	The researchers	Testing
Level of personalisation (add personal features)	It should be easy to add features	The researchers	Testing/interview
Availability of the software	The software should be free of charge	VCC	Literature/Interviews
Pedagogogical options (highlight elements, add arrows...)	Feasible and easy to implement	The researchers	Testing/interview
Multi-user possibility	The software should support more than one user in the VR scene	The researchers	Literature/interviews

4.1.3 Pugh matrix

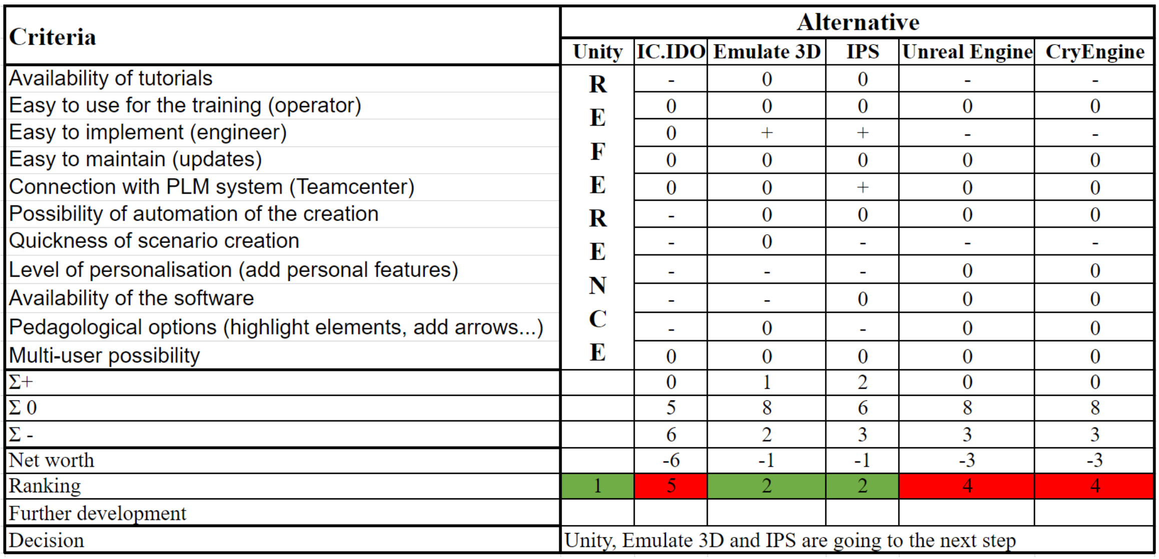
The first tool used to select the software was the Pugh matrix. For the first iteration, IPS was selected as the reference and each software was compared to it, see Table 4.3 below.

Table 4.3: First iteration of the Pugh matrix (IPS as the reference)

Criteria	**Alternative**					
	IPS	**IC.IDO**	**Emulate 3D**	**Unity**	**Unreal Engine**	**CryEngine**
Availability of tutorials	**R E F E R E N C E**	0	0	+	0	0
Easy to use for the training (operator)		0	0	+	+	+
Easy to implement (engineer)		0	+	-	-	-
Easy to maintain (updates)		-	0	0	0	0
Connection with PLM system (Teamcenter)		0	0	-	-	-
Possibility of automation of the creation		-	+	+	+	+
Quickness of scenario creation		0	0	0	-	-
Level of personalisation (add personal features)		-	0	+	+	+
Availability of the software		0	-	0	0	0
Pedagogical options (highlight elements, add arrows...)		0	+	+	+	+
Multi-user possibility		0	0	0	0	0
Σ+		0	3	5	4	4
Σ 0		8	7	4	4	4
Σ -		3	1	2	3	3
Net worth		-3	2	3	1	1
Ranking		4	2	1	3	3
Further development						
Decision	IC:IDO should be eliminated because of its poor score					

After the first iteration, it was decided that IC.IDO should be eliminated as it received the worst score and was the only one with a negative score. This is because, as was said during the interview with a technical engineer working at ESI Group (the provider of IC.IDO), there is no possibility of customization within IC.IDO, it is a rigid software. Unity was the software that received the most points; therefore, it was chosen as the reference for the second iteration.

Table 4.4: Second iteration of Pugh matrix (Unity as the reference)



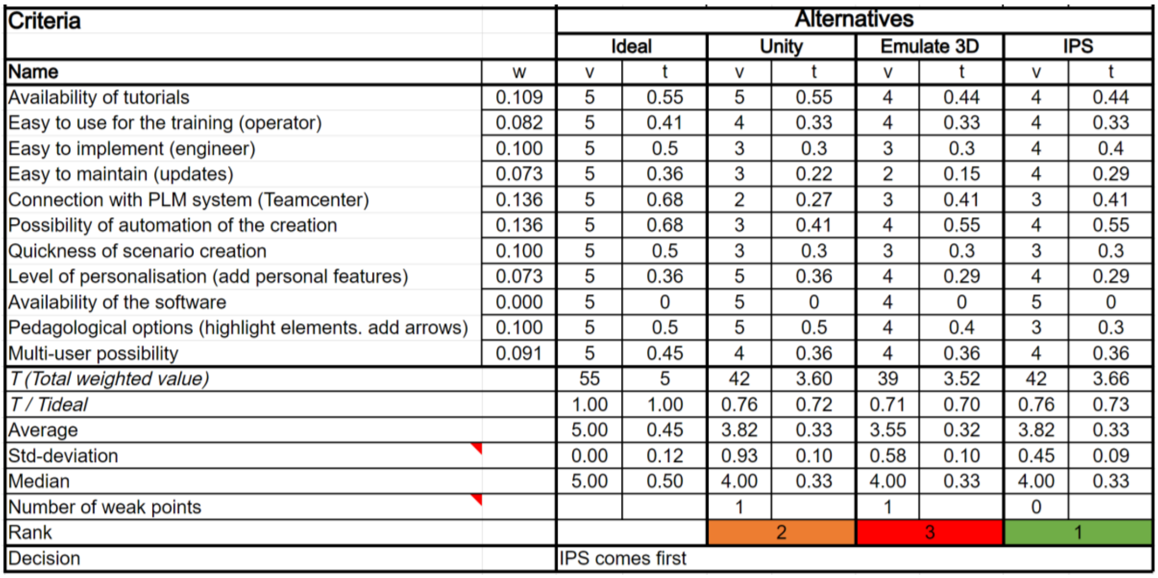
Criteria	Alternative					
	Unity	IC.IDO	Emulate 3D	IPS	Unreal Engine	CryEngine
Availability of tutorials	REFERENCE	-	0	0	-	-
Easy to use for the training (operator)		0	0	0	0	0
Easy to implement (engineer)		0	+	+	-	-
Easy to maintain (updates)		0	0	0	0	0
Connection with PLM system (Teamcenter)		0	0	+	0	0
Possibility of automation of the creation		-	0	0	0	0
Quickness of scenario creation		-	0	-	-	-
Level of personalisation (add personal features)		-	-	-	0	0
Availability of the software		-	-	0	0	0
Pedagogical options (highlight elements, add arrows...)		-	0	-	0	0
Multi-user possibility		0	0	0	0	0
Σ+		0	1	2	0	0
Σ 0		5	8	6	8	8
Σ -		6	2	3	3	3
Net worth		-6	-1	-1	-3	-3
Ranking	1	5	2	2	4	4
Further development						
Decision	Unity, Emulate 3D and IPS are going to the next step					

Looking at the ranking after the second iteration, see Table 4.4, Unreal Engine and CryEngine, that came last, were eliminated. This decision was made both because they received the worst score and because, as was discovered while researching on the software, Unity is the best suited game engine, compared to the other two (it is for instance said to be easier to learn and having more resources available online). Also, it can be noted that IC.IDO received once again a poor score, which justified the fact that it cannot be selected. Therefore, after the two iterations of the Pugh Matrix, Unity, IPS and Emulate3D were the solutions selected to go through and be used in the Kesselring matrix.

4.1.4 Kesselring matrix

The remaining VR software solutions (Unity, Emulate3D, and IPS) were put into the Kesselring matrix and using the grading scale (see Appendix B.2) and the weighting matrix (see Appendix B.3 and Table 3), the software solutions could be compared in a more precise way. The Kesselring matrix is presented in Table 4.5 below.

Table 4.5: Kesselring matrix



Criteria		Alternatives							
		Ideal		Unity		Emulate 3D		IPS	
Name	w	v	t	v	t	v	t	v	t
Availability of tutorials	0.109	5	0.55	5	0.55	4	0.44	4	0.44
Easy to use for the training (operator)	0.082	5	0.41	4	0.33	4	0.33	4	0.33
Easy to implement (engineer)	0.100	5	0.5	3	0.3	3	0.3	4	0.4
Easy to maintain (updates)	0.073	5	0.36	3	0.22	2	0.15	4	0.29
Connection with PLM system (Teamcenter)	0.136	5	0.68	2	0.27	3	0.41	3	0.41
Possibility of automation of the creation	0.136	5	0.68	3	0.41	4	0.55	4	0.55
Quickness of scenario creation	0.100	5	0.5	3	0.3	3	0.3	3	0.3
Level of personalisation (add personal features)	0.073	5	0.36	5	0.36	4	0.29	4	0.29
Availability of the software	0.000	5	0	5	0	4	0	5	0
Pedagogical options (highlight elements. add arrows)	0.100	5	0.5	5	0.5	4	0.4	3	0.3
Multi-user possibility	0.091	5	0.45	4	0.36	4	0.36	4	0.36
T (Total weighted value)		55	5	42	3.60	39	3.52	42	3.66
T / Tideal		1.00	1.00	0.76	0.72	0.71	0.70	0.76	0.73
Average		5.00	0.45	3.82	0.33	3.55	0.32	3.82	0.33
Std-deviation		0.00	0.12	0.93	0.10	0.58	0.10	0.45	0.09
Median		5.00	0.50	4.00	0.33	4.00	0.33	4.00	0.33
Number of weak points				1		1		0	
Rank				2		3		1	
Decision		IPS comes first							

Based on the results, the solutions that received the highest score were IPS and Unity. Both received the same score, but when observing the standard deviation values, Unity's score deviated more compared to IPS and it also presented a weak point (a score of 2 or below for a criterion), while the latter did not. Therefore, IPS appeared to be the best option to work towards the scalability of VR training at VCC, and it is the one that was chosen to be used in the project. More precisely, IPS would be used together with Lua, the scripting language that is associated with it, to allow its full capabilities when it comes to auto generation.

Nevertheless, it was noted that Unity scored higher when it comes to personalization and pedagogical options. Thus, this software was also used for the creation of the tryout VR scene, to take inspiration from it and to see, without having to put too much effort, what pedagogical aspects were good to have or not for the training.

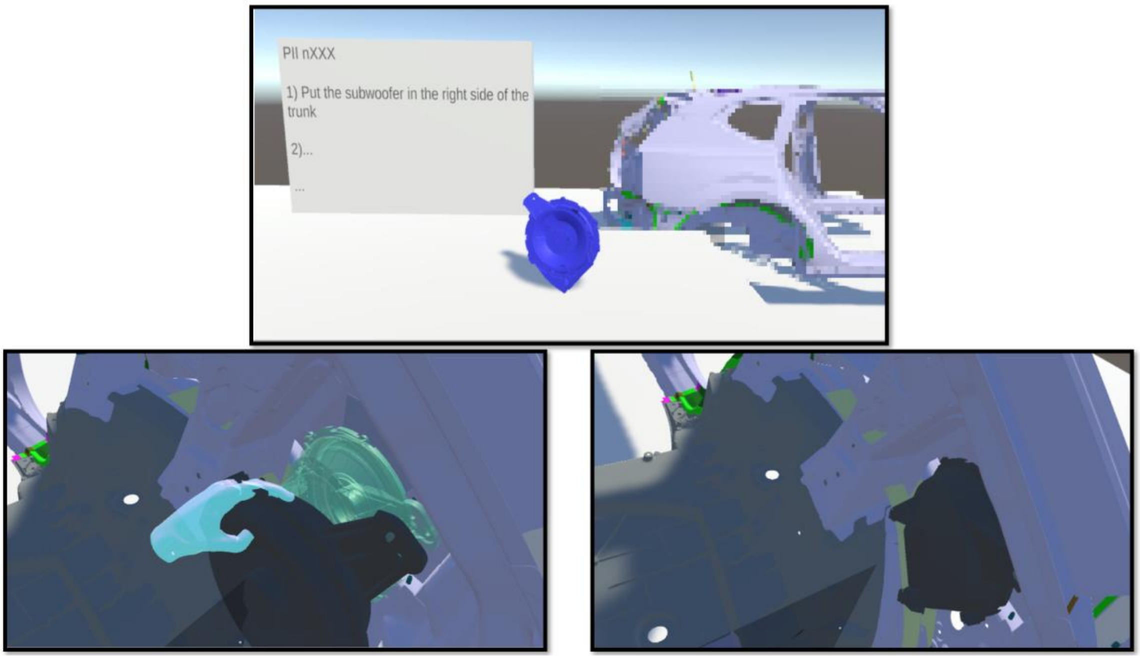
4.2 Tryout VR scene

The tryout VR scene, which is the assembly of a subwoofer in the trunk of a car, was realized both with Unity and with IPS. The results gathered when creating it with both software are presented in the following sub-sections.

4.2.1 With Unity

Realizing the training scene with Unity, although it not being the software chosen to continue with, provided several outputs. In addition, different guidance and pedagogical elements were tested, such as to define which ones were the most relevant. It was interesting to do that because these features were already included in the VR Builder construction tool, which is not the case in IPS, where it needs to be made available via scripting.
The following features, presented in Figure 4.2 below, were considered interesting:

- Change in color of the part that needs to be placed or interacted with.
- Use of a colored snapping zone to indicate where to place / assemble a given element.
- Display of the instructions in a text format, on a visible panel, rather than giving the instructions via a voice message, which might be disturbing for the trainee.



PII nXXX

1) Put the subwoofer in the right side of the trunk

2)...

...

Figure 4.2: Screenshots from the tryout VR scene in Unity

4.2.2 With IPS

Working on that first case allowed to highlight the general workflow to create VR training scenes in IPS. In that case, the focus was on allowing the picking and placing of parts and not on realism or pedagogical elements. This was executed since, to create a VR training scene that is good teaching material, creating the VR training scene itself is the very first step.

The workflow can be divided in several steps, shown in Figure 4.3 that are as follow:

1) Import the assembly:
   The CAD file, obtained from Teamcenter must be opened in IPS.
2) Identify the rigid bodies:
   Determine what sub-parts of the assembly should be defined as rigid bodies.
3) Create the rigid body path planning:
   This step is the most important and complicated. It allows to both interact with the parts and visualize them being placed in their assembled position in VR. Several sub-steps are required, which are detailed in Figure 4.3. For a better understanding, names, in dark blue, have been given to different positions.
4) Create the operation sequence parts:
   Each rigid body path planning needs to be converted into a part of operation sequence.
5) Organize the overall operation sequence:
   All the operation sequence parts must be organized in a global sequence.

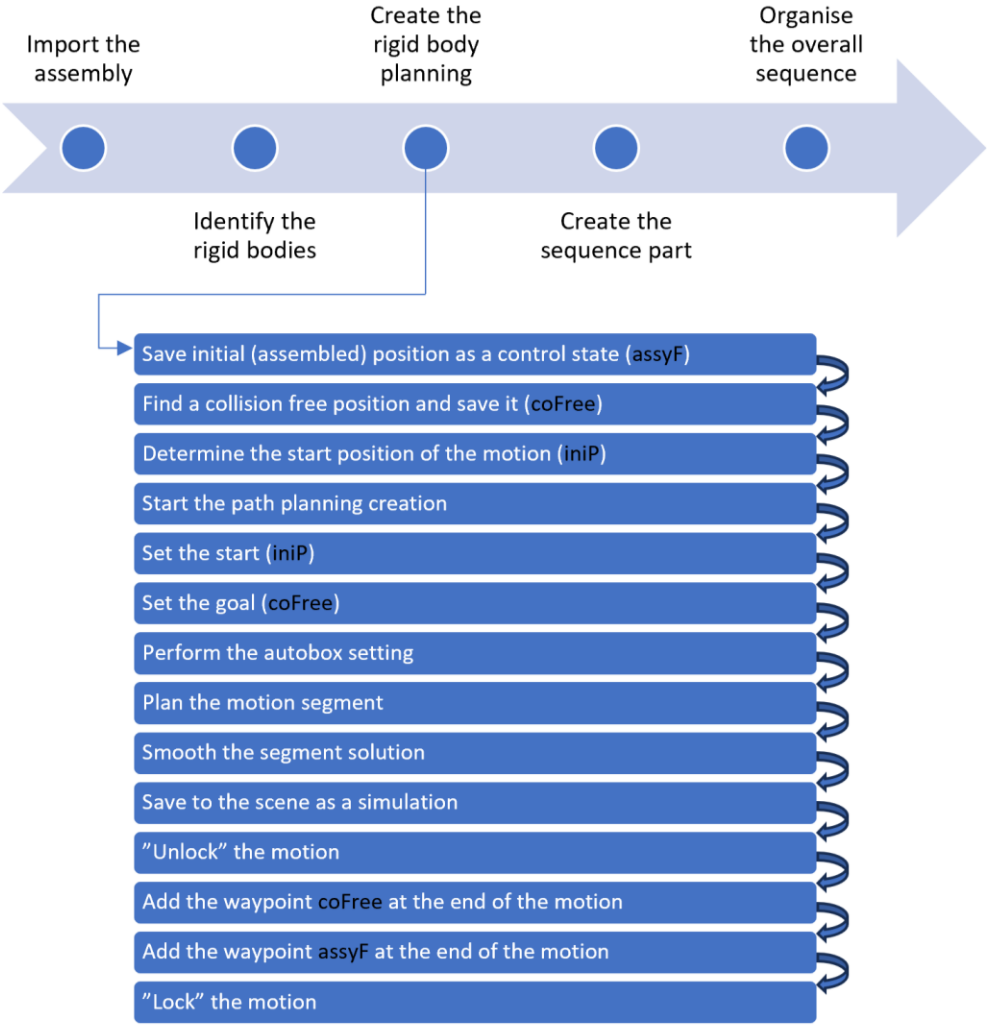

Import the assembly
Identify the rigid bodies
Create the rigid body planning
Create the sequence part
Organise the overall sequence
Save initial (assembled) position as a control state (assyF)
Find a collision free position and save it (coFree)
Determine the start position of the motion (iniP)
Start the path planning creation
Set the start (iniP)
Set the goal (coFree)
Perform the autobox setting
Plan the motion segment
Smooth the segment solution
Save to the scene as a simulation
"Unlock" the motion
Add the waypoint coFree at the end of the motion
Add the waypoint assyF at the end of the motion
"Lock" the motion

Figure 4.3: Workflow for VR interactions enabling in IPS

Once all of this had been done, the resulting VR scene was one where, first, it was possible to visualize the part that needed to be assembled, going from its stored position to its assembled position. That scene also allowed to pick and place the part, from a start position, where it is hypothetically stored, to an assembled position, where the part can snap if it is released precisely enough.

4.2.3 Directions for continuing the work

Based on the VR training scene that had been created in IPS as well as on the comparison of the latter to the one created in Unity; several focus points were determined for the continuation of the work to create a VR operator training scene in IPS that includes pedagogical elements.

First, Unity allowed to display instructions as a text in the VR environment and, after testing and in accordance with the literature (Philippe S. et al., 2020), this was deemed very useful for the trainee to understand what the different steps are. This is more difficult to achieve in IPS since there are no built-in options to create a text that is visible from within VR. Furthermore, for longer sequences, typing long text instructions can turn out to be quite time consuming, therefore, being able to import such instructions automatically is deemed of high interest and will be focused on later in the project.

Second, in Unity there was a built-in option to enable good snapping and to highlight the position in which the part should be placed. In IPS, the snapping possibilities could be enhanced by following some modifications of the part's geometry frame and a solution needed to be found to highlight, similarly, the position where the part was to be placed.

Finally, the scenes in both Unity and IPS highlighted that for a better immersion and realism, it would be beneficial to put the trainee in a realistic, assembly line-like environment. This can be achieved using point cloud environments obtained from 3D scans of real-life factories, as well as realistic fixtures, trolleys, and racks, adapted to the sequence that is to be trained.

4.3 Large-scale implementation of VR operator training

In this section, the results of when and how VR operator training should be implemented are presented, with a focus on the data required to create VR training scenes that would be good training material. Furthermore, the way to access the data is also underlined.

4.3.1 When to do VR training

At VCC, the product development process is similar to the one presented in the literature (See Figure 2.1 for instance). The following Figure 4.4 presents the main phases at VCC and the place that VR training could have. It was designed based on personal communications with Johan Segeborn (March 12, 2024).

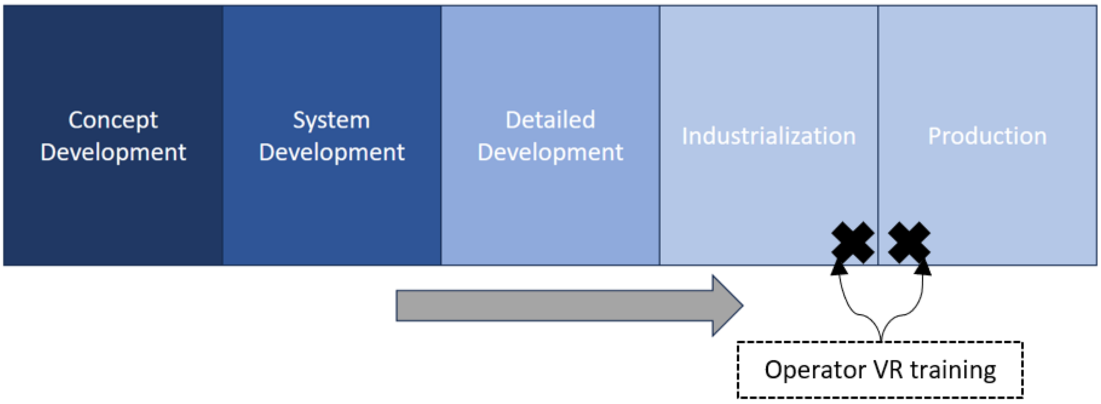


Figure 4.4: Product development process at VCC

VR Operator training would be organized and provided either late during the industrialization phase or during production, just before the production in the actual plant begins in case of a new product or a new assembly line that would be established. VR training scenes would therefore be created based on digital data produced during the detail development and early industrialization phases.

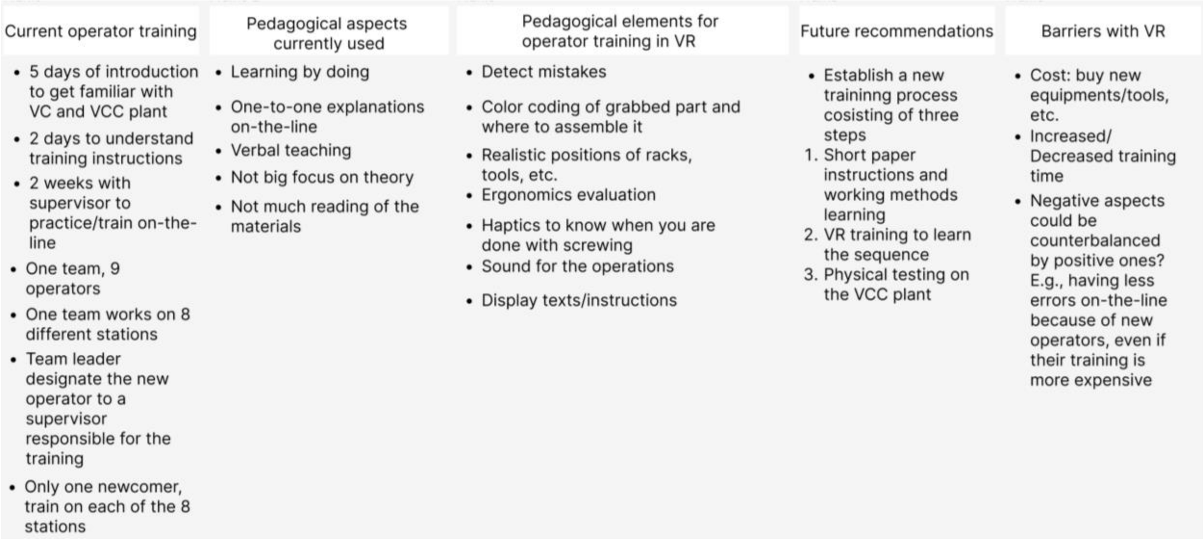
This would allow, in the case of a new product or line, to reduce the lead time resulting from ordering and receiving new physical parts by taking advantage of one of VR training strengths. The latter is the fact that no physical parts or places are needed to start training operators. Therefore, VR operator training could start earlier when it is needed, so production can also start earlier, thus the whole global product development lead time could be reduced. The second important use case is when a product is already being manufactured but necessitates some modifications, either in some of its sub-systems or in the way it is assembled. Then, the associated assembly VR training should be modified as well, and this should be easily doable. Thus, this highlights the fact that VR training scenes should be easy to modify, for instance altering its layout, the parts included or their positions.

All in all, this timing aspect of the implementation of VR operator training would restrict the data that could be used to populate the VR scenes. Furthermore, data created during the production phase would then not be available, for instance, 3D scans of the actual station in the plant could not be used as it would not necessarily be available, or documents detailing the sequence, created by plant launch, neither could be used.

4.3.2 How to do VR training

The responds from the plant launch manager and launch specialist interview contained information on the process used for operator training. Feedback on the implementation of VR operator training in the training process was also received, as well as future recommendations and limitations with operator training in VR, see Table 4.6.

Table 4.6: Summarized interviewees' responds regarding VR operator training



Current operator training	Pedagogical aspects currently used	Pedagogical elements for operator training in VR	Future recommendations	Barriers with VR
• 5 days of introduction to get familiar with VC and VCC plant • 2 days to understand training instructions • 2 weeks with supervisor to practice/train on-the-line • One team, 9 operators • One team works on 8 different stations • Team leader designate the new operator to a supervisor responsible for the training • Only one newcomer, train on each of the 8 stations	• Learning by doing • One-to-one explanations on-the-line • Verbal teaching • Not big focus on theory • Not much reading of the materials	• Detect mistakes • Color coding of grabbed part and where to assemble it • Realistic positions of racks, tools, etc. • Ergonomics evaluation • Haptics to know when you are done with screwing • Sound for the operations • Display texts/instructions	• Establish a new traininng process cosisting of three steps 1. Short paper instructions and working methods learning 2. VR training to learn the sequence 3. Physical testing on the VCC plant	• Cost: buy new equipments/tools, etc. • Increased/ Decreased training time • Negative aspects could be counterbalanced by positive ones? E.g., having less errors on-the-line because of new operators, even if their training is more expensive

From the responds, the key takeaways were that by implementing VR training, the new operator can establish the knowledge of the sequence of how the parts are assembled in systematic order so that he / she is prepared to implement it physically at the VCC plant. Moreover, it appeared that VR operator training would fit in the operator training process as a final step before the on-the-line training. That way, the operator would be able to focus on acquiring the craftmanship when training on the line as he / she would already be familiar with the sequence thanks to the VR training previously provided.

Furthermore, this interview highlighted the importance of certain pedagogical elements, presented in the third column of the Table 4.6, that would be good to implement in the VR operator training scenes. It is noteworthy that some of them corresponded to the elements that were stressed out in Sub-section 4.2.1.

4.3.3 VR operator training data mapping

All the required data for the creation of a VR operator training scene were then identified. Here is a summary of what is required:

- Assembly (CAD parts assembled),
- Tools and equipment (CAD),
- Station layout (furniture CAD),
- Plant environment (generic plant environment point cloud),
- Assembly sequence (information of the operations' order),
- Instructions (as text, to be followed by the trainee).

The next step was then to understand from where the different data can be obtained. Figure 4.5 below presents a summary of the data organization in Teamcenter: a Collaboration Context (CC) object contains a Bill Of Materials (BOM), Bill Of Equipment (BOE) and a Bill Of Process (BOP).

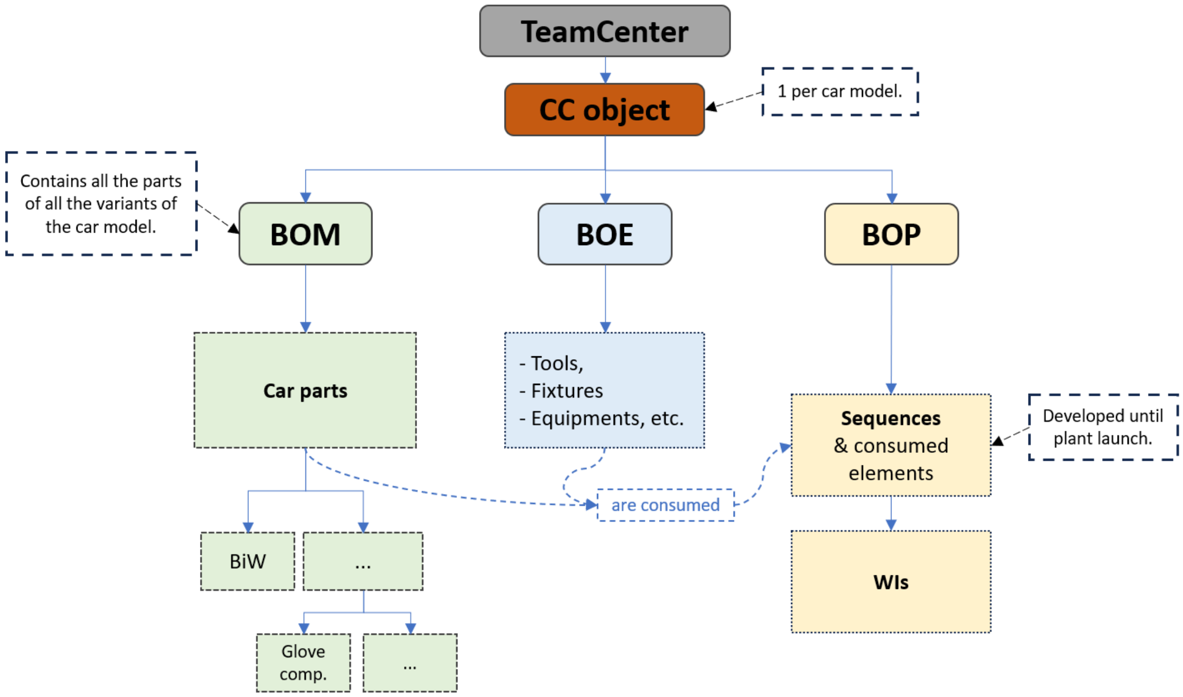


Figure 4.5: Data organization in Teamcenter

It appeared then that almost all the data identified could be gathered from the BOP. Additionally, all the CAD files (including assembly, equipment and station-specific furniture), coming from both the BOE and BOM can be accessed from the BOP, where they are consumed. Furthermore, as the sequences are detailed the WIs, the text instructions as well as the assembly sequence information can also be obtained from this database. When it comes to the generic plant environment point cloud, it is not available in the databases. Nevertheless, as it is something that is wanted to be generic, it only must be created once and for good but since it is not the focus of the project, this has been put on the side.

4.4 Scalability demo VR scene

In this section, the results obtained from creating the scalability demo VR scene are presented. It describes both VR scenes, created with and without Lua scripting as well as what they include and allow, with a focus on the creation process of the one that was partly auto generated. Thus, the first sub-section focuses on the VR training scene created without scripting and the remaining sub-sections focus on the VR training scene that was created using scripting.

4.4.1 Without Lua scripting

Based on the education received from FCC and VCC, the scalability demo VR scene was created. The parts and screws/rivets, placed on trolleys, and screws, placed on the bin that was on the fixture, were created from primitive shapes and illustrated the realistic bin that are used at the plant. Therefore, it was made as realistic as possible and as close as possible to the plant's setup, considering the limited time and resource available. Motions (see the white lines in Figure 4.6) and a sequence were created to illustrate to the operator how the parts should be assembled onto the fixture. Furthermore, the snapping zones were created for each part, highlighted in green, to highlight to the trainee where the part must be assembled on the fixture.

Flexible cables were also created with the position the same as the CAD geometries. Furthermore, some of the cables (3 of them), were interacting with rigid bodies. To mimic the realistic behavior of that, nodes on the cables were created to the grips and for the rigid bodies moving along with the cables, the rigid body was the parent, and the node was the child.
For the end parts, where the cables are connected to a rigid body, the rigid body was the parent and the node on the start / end grips were a child to replicate the reality, that the operator will grab the rigid bodies and not on the nodes. When creating the motions for the cables, the whole cable moving from where it is placed in the rack to its final assembly position was not possible. Instead, creating motions and sequence of the rigid bodies attached on the nodes of the cable was possible. A snapshot of the created VR scene is presented in Figure 4.6.

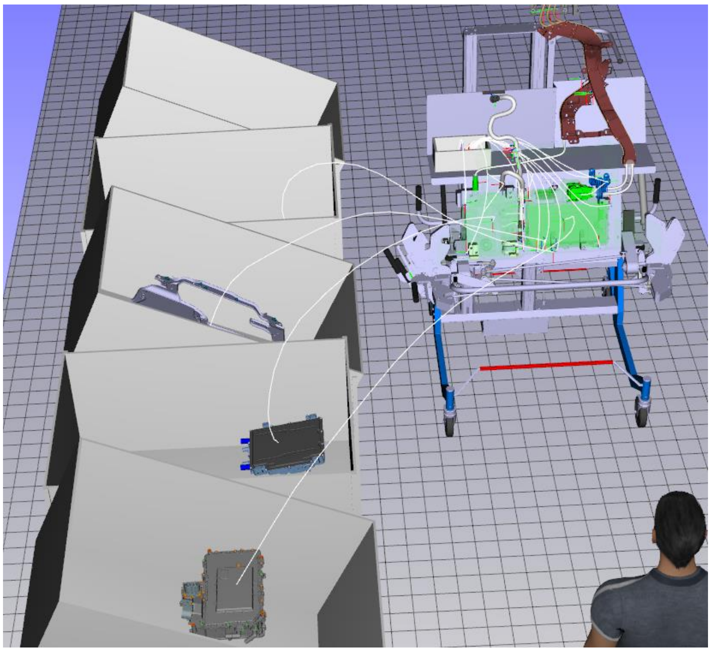


Figure 4.6: Illustration of the VR scene created without using scripts in IPS

4.4.2 Identified steps for automatic VR training scene creation

Based on the workflow previously identified and presented in Sub-section 4.2.2 and on the highlighted improvement points, presented in Sub-section 4.2.3, a new workflow was established. It was then used as a base for the whole automation process via scripting. Figure 4.7 presents it and provides more details on the most critical part of it. In that figure, the elements colored in light blue are the new steps that were identified as missing from the previous workflow and that have been implemented in the final one.

In the following sub-section, each bullet-point in Figure 4.7 and how it was dealt with, to enable as much automatic creation as possible, is detailed.

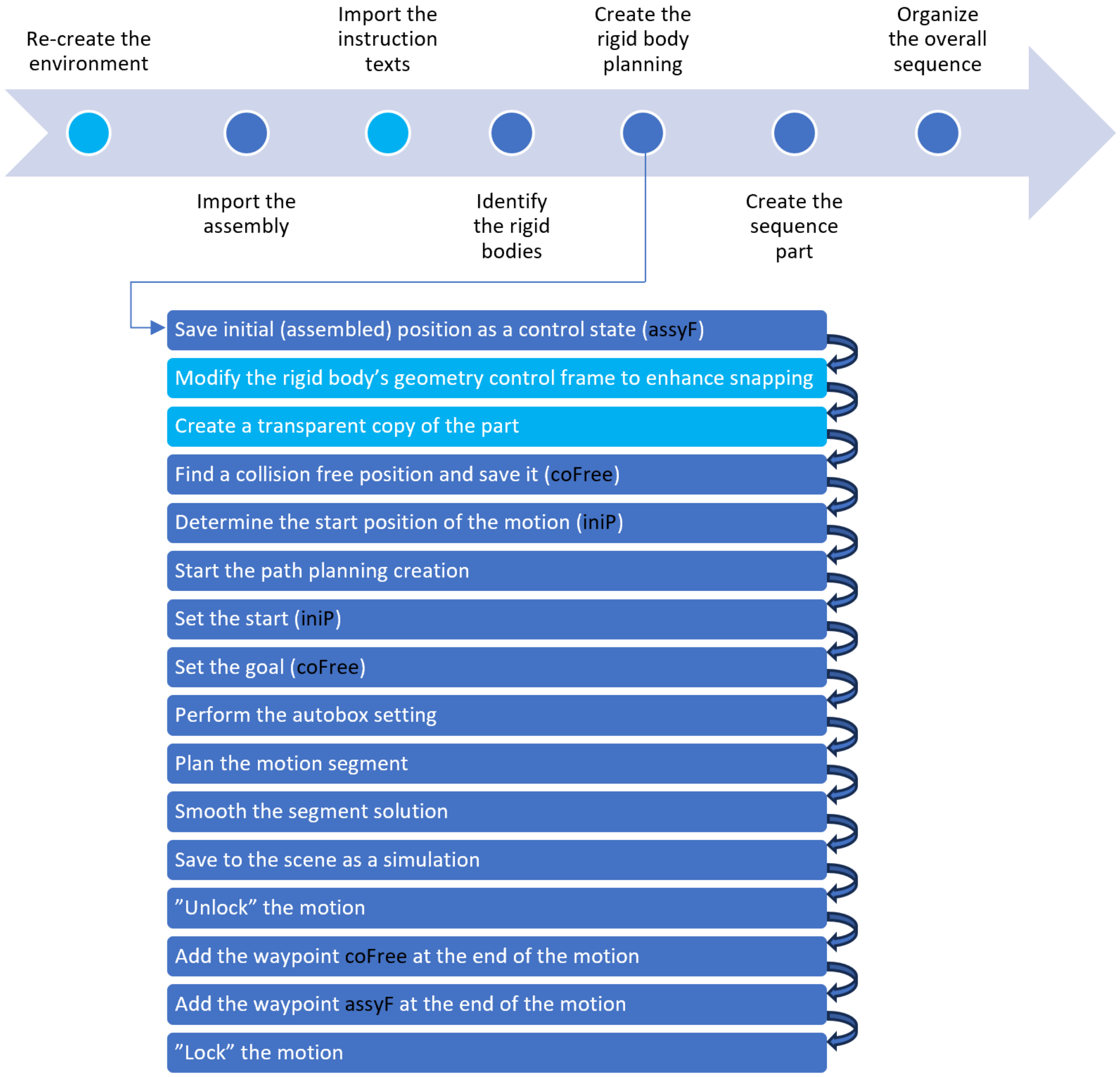

Re-create the environment
Import the assembly
Import the instruction texts
Identify the rigid bodies
Create the rigid body planning
Create the sequence part
Organize the overall sequence
Save initial (assembled) position as a control state (assyF)
Modify the rigid body's geometry control frame to enhance snapping
Create a transparent copy of the part
Find a collision free position and save it (coFree)
Determine the start position of the motion (iniP)
Start the path planning creation
Set the start (iniP)
Set the goal (coFree)
Perform the autobox setting
Plan the motion segment
Smooth the segment solution
Save to the scene as a simulation
"Unlock" the motion
Add the waypoint coFree at the end of the motion
Add the waypoint assyF at the end of the motion
"Lock" the motion

Figure 4.7: Workflow for the overall scene creation

4.4.3 Creation of the scalability demo VR scene

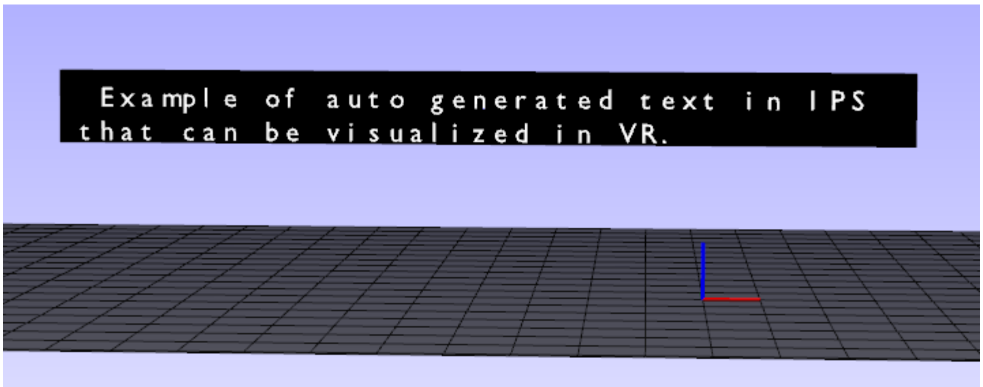
Creating a realistic environment:

First, the idea was to use a point cloud environment that would come from PlantView (which is a software where all the 3D scans of VCC's plants are stored). But it turned out that for that specific station, 3D scans were not available. Furthermore, the boxes / trolleys and racks, in which parts are stored on the line, before being assembled, were not available for that station neither. Thus, no point cloud was used, and the environment was populated with boxes / trolleys and racks that were created based on the ones present at the plant and with a trolley that was obtained from Teamcenter. See Figure 4.10 for a snapshoot of how the layout of the scene looked like.

Importing the assembly:

The assembly, with all the parts that composed it, was obtained from Teamcenter, and downloaded on the local device manually. Then, a Lua script that had been created was used to load that file to IPS and to delete the empty entities that were inside the geometry tree.

Importing the instructions:
Text instructions were also imported in the scene. See Figure 4.8 below for an example. They come from a .csv file, generated from Teamcenter that was then extracted manually to the local device. Then, a second Lua script that had been created allowed to populate the IPS scene with 3D instructions that could be read by the trainee from within VR.



Example of auto generated text in IPS that can be visualized in VR.

Figure 4.8: 3D text instruction in IPS

Furthermore, another Lua script was created, intended to be used from within VR in IPS. It allowed the trainee to display the instructions one after another by selecting it and pressing a dedicated button on the VR controller.

Identifying the rigid bodies:
Before enabling the interactions from within VR, a manual step was necessary. It was required to identify which entities in the geometry tree structure should be assigned as a rigid body or not (for instance, each screw is to be a rigid body as it needs to be picked and placed but finding each of them individually manually in the geometry tree structure was required).

Enabling interactions in VR:
A final Lua script was used, to generate the motions from the stored position to the assembled position of the parts and to enable pedagogical snapping on the assembled position. As input, it just needed the coordinates of the start position, while as outputs, was the motion itself, that could be played as well as the operation sequence part, associated with that motion. It also created the assembled-state snapping zone as well as a transparent copy of the part, to serve as a visual cue, in that assembled position and finally, it modified the definition of the rigid body, in order to enhance the snapping abilities. In Figure 4.10, the white lines represent the motions that were created.

Organizing the operation sequence:
Once the motions for all the parts were created, a final step, which had to be done manually, was to organize all the operation sequence parts (replay of individual motions) into a global operation sequence. That way, the whole assembly could be played all at once easily. This step, based on engineering knowledge consisted of going from an operation sequence that looks like the one on the left in Figure 4.9, to one that looks like the one on the right.

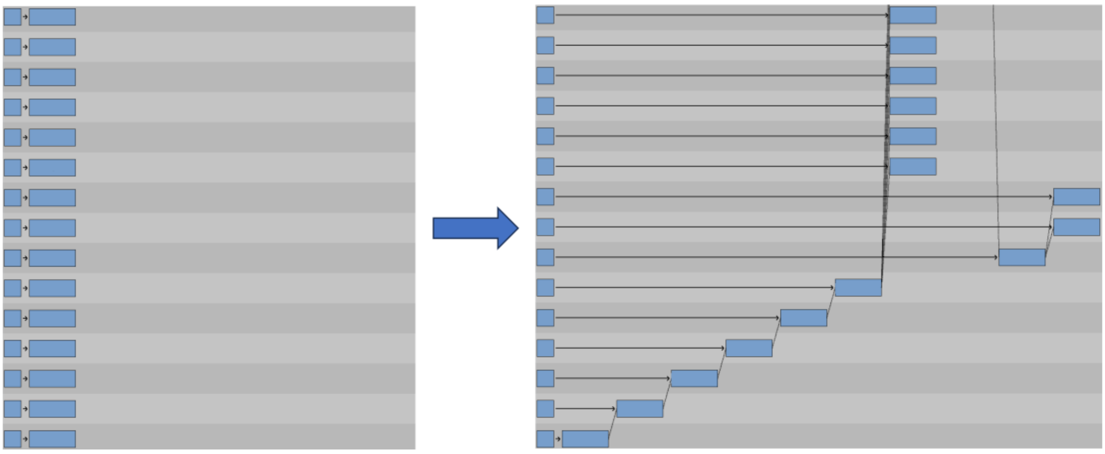


Figure 4.9: Operation sequence before and after organization

4.4.4 Outlook and use of the scalability demo VR scene

The following Figure 4.10 shows a snapshoot of the training environment that was created and that is described in the previous sub-section.

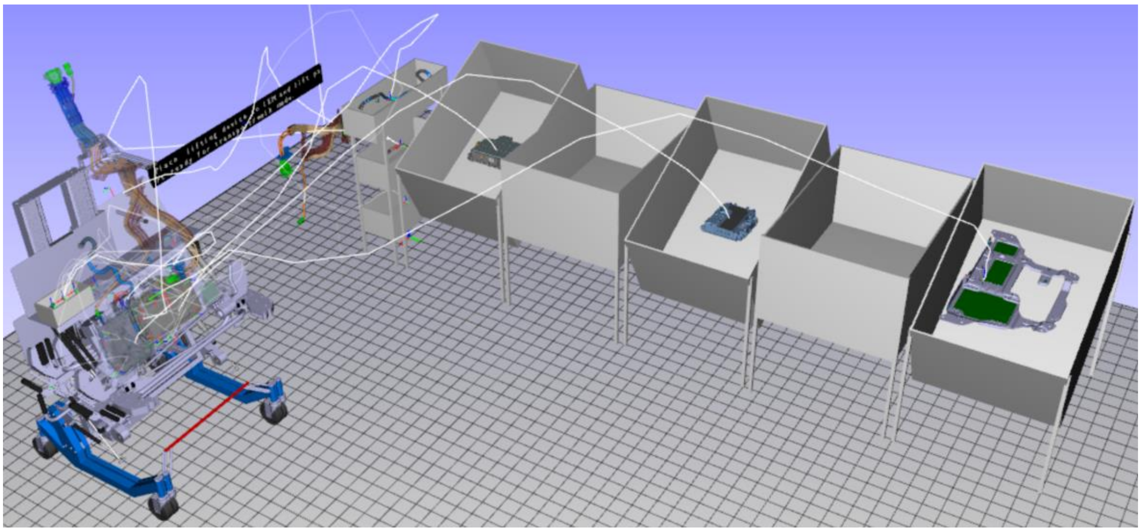


Figure 4.10: Illustration of the VR scene in IPS created with Lua scripting

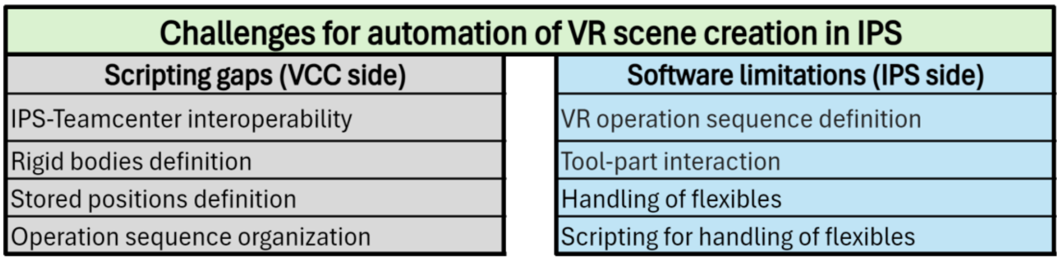
As it appears in the figure above, the scalability demo VR scene allowed the trainee to pick and place parts from their stored position to their assembled positions. The latter was indicated by a transparent copy of the part at its assembled position, in which the corresponding part could be snapped, with as much convenience as can be allowed by IPS. Furthermore, the trainee could read and follow along the instructions associated with the sequence in a seamless way. Finally, the training was to be performed in a realistic virtual environment, with the assembly being done on the actual trolley used in that station at the plant and with boxes and racks similar to the ones that were actually used.

It should be noted that some things were not done as planned because of the challenges that are presented in the following sub-section. First, all the parts were simplified as rigid bodies, irrespective of their real-life characteristics. Moreover, neither the racks and boxes, nor the tools were available in Teamcenter. Thus, the tools were not included in the scene and the station layout was recreated as well as possible with some simple CAD files.

4.5 Challenges of auto generation of VR training scene

Creating the scripts to enable the automatic creation of the scalability demo VR scene highlighted several challenges for the auto generation of VR training scene in IPS. They were divided into two groups, detailed in the sub-sections below and presented in Table 4.7 below.

Table 4.7: Summary of the identified challenges for auto generation of VR scene in IPS



Challenges for automation of VR scene creation in IPS	
Scripting gaps (VCC side)	**Software limitations (IPS side)**
IPS-Teamcenter interoperability	VR operation sequence definition
Rigid bodies definition	Tool-part interaction
Stored positions definition	Handling of flexibles
Operation sequence organization	Scripting for handling of flexibles

4.5.1 Scripting gaps

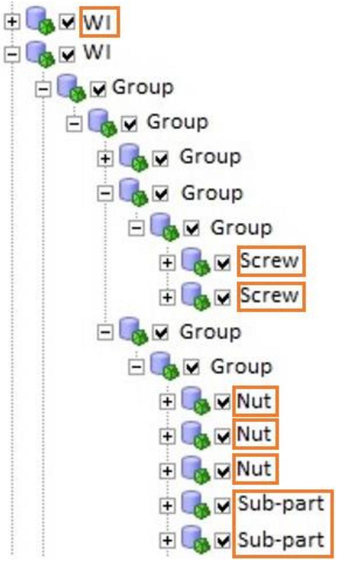
As was mentioned earlier, the initial goal with Lua scripting was to write only one script that would autogenerate the VR training scene all at once. Nevertheless, in the end 3 scripts were created to perform that. This was done that way because of gaps caused by activities that needed to be performed manually.

IPS-Teamcenter interoperability:

First, there is no interoperability between IPS and Teamcenter. Therefore, all the files extracted from the PLM system had to first be downloaded on the local desktop to be able to access them with Lua scripts. It was not possible to access any files directly from Teamcenter.

Rigid bodies definition:

Identifying elements in the geometry tree structure that should be assigned as rigid bodies could not be done automatically. In fact, the way it was organized and the names that were given to the elements composing it do not allow to implement an automatic identification. To illustrate this challenge, Figure 4.11 below presents an extract of a geometry tree structure of an assembly retrieved from Teamcenter and then imported into IPS.



```
⊞ ☑ WI
⊟ ☑ WI
    ⊟ ☑ Group
        ⊟ ☑ Group
            ⊞ ☑ Group
            ⊟ ☑ Group
                ⊟ ☑ Group
                    ⊞ ☑ Screw
                    ⊞ ☑ Screw
            ⊟ ☑ Group
                ⊟ ☑ Group
                    ⊞ ☑ Nut
                    ⊞ ☑ Nut
                    ⊞ ☑ Nut
                    ⊞ ☑ Sub-part
                    ⊞ ☑ Sub-part
```

Figure 4.11: Geometry tree structure organization (modified names) with elements that should be rigid bodies framed in orange

That Figure 4.11 highlights that it was not possible to create an algorithm that would go through the tree and create rigid bodies for the elements for which it would need to be done. The identification of these elements was possible only manually, based on engineering knowledge. First, these elements could not be identified by their hierarchical level in the tree as it appears in Figure 4.11 that they do not share a common level in the tree and as some elements must be excluded from a group before being able to assign the group as a rigid body (for instance, in the figure, the 3 nuts that have to be excluded from the last group for it to be assign as a rigid body). Second, the name of the elements could have been used to assign rigid bodies to the relevant geometries, but it was not possible either. A way to use this would have been to have a list of the names that were to be found in the geometry tree structure and then to find them with an algorithm, but such a list could not be obtained.

Stored positions definition:
Once the boxes and racks were imported to the scene, it was necessary to find manually the adequate coordinates in which the parts would be stored in them. This did not take long but it was not possible to have this data automatically. The problem was the same for CAD from Teamcenter, as there is no metadata of this type associated with the files.

Operation sequence organization:
Once the parts of the operation sequence were individually created with the script, they had to be ordered manually to form the global operation sequence (see Figure 4.9). In fact, there was no way to do that automatically. Here, once again, the names of the parts could have been used to put in order the operation sequence parts but an ordered list of names was not possible to obtain automatically and was therefore not used.

4.5.2 Software limitations

Some barriers that were identified come from the nature of the software that was used to create the VR scene. As a matter of fact, IPS, a simulation tool, has many strengths but also comes with some weaknesses that led to the following difficulties.

VR operation sequence definition:
IPS is not a game engine. Therefore, first, there was no gamification possibilities. Furthermore, from within VR, picking and placing parts could exclusively be done freely by the trainee, there was no way to restrict the actions that were possible (for instance to pick one specific part only) and no way to check that the steps of the assembly sequence were performed in a correct order. Because of that, no visual indication could be given about which part should be picked at what time, except from the text instructions that must be displayed one after the other by an action of the trainee, using the Lua script created for that purpose. There was nothing that could be done for an event to trigger another event as to follow a progressive predefined assembly sequence. Thus, it was only possible to indicate to the trainee with text instructions what should be done as well as where all the parts must be placed, with transparent copies in assembled positions.

Tool-part interactions:
For the reasons detailed in the paragraph above, no tools were imported into the scalability demo VR scene because they could be manipulated but not used as tools from within VR. For instance, it was not possible to visualize the tightening of a screw when placing a grabbed screwdriver nearby in VR. IPS did not allow to use a collision to trigger an event, for reasons that were described in the paragraph above.

Handling of flexibles:

Simulation of flexible bodies in VR is difficult and the interactions possible with such objects in VR do not feel very realistic. Additionally, from within VR in IPS, it was only possible to grab and move a cable with two ends maximum, as each end must be grabbed at the same time. Therefore, a cable with three ends for instance was simply not movable by the trainee in VR. Finally, based on the interviewees' responds, see Sub-section 4.3.2, it was deemed more pedagogical, in terms of craftmanship, to experience the behavior of the flexible cables physically rather than experiencing something not realistic in VR for the operators.

Scripting for handling of flexibles:

In the scalability demo VR scene created with Lua scripting, all the parts were simplified as rigid bodies, even the ones that were cables. Furthermore, it was possible to enable some types of interactions with cables from within VR in IPS, but it required to follow a workflow that was very case-dependent and almost unique to each cable. Because of that, it was not possible to create a unique script that would work with each type of cable. Therefore, putting a lot of efforts with scripting on cables was set aside, as it was difficult and not really beneficial.

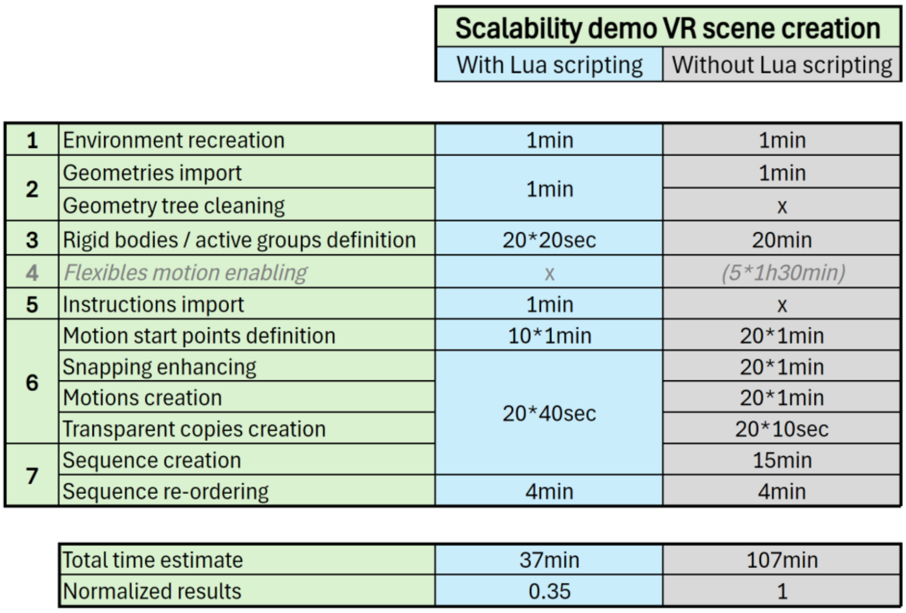
4.6 Evaluation of the results

In this section, a comparison of the results from the scalability demo VR scene created with and without scripting is being realized, with a focus on time and possibilities.

The development time taken to create the scene, as can be seen in Table 4.8, was evaluated by dividing it into 6 main points, corresponding to the 6 main steps presented Sub-section 4.4.3.

The first step is the recreation of the environment, for a better realism. The second step is to import the assembly geometry received from Teamcenter, into IPS and it also includes removing entities from the geometry tree structure that are empty (do not contain any CAD geometries). Then, the third step is creating active groups and rigid bodies to maintain a structured workflow with the purpose to facilitate the components and the assembly order. After that, because of different possibilities, either interactions with flexibles were enabled or text instructions were imported, depending on the scene. For both scenes, motions for each rigid body were created, in which the parts were going from a determined start position corresponding to where the parts were placed in the factory. In that step, pedagogical elements, such as enhancing snapping and creating transparent copies of the parts were considered. Lastly, the operation sequence parts were created, and the global operation sequence was organized so as the whole assembly could be visualized, all at once.

Table 4.8: Time evaluation of the creation processes' steps



		Scalability demo VR scene creation	
		With Lua scripting	Without Lua scripting
1	Environment recreation	1min	1min
2	Geometries import	1min	1min
	Geometry tree cleaning		x
3	Rigid bodies / active groups definition	20*20sec	20min
4	*Flexibles motion enabling*	*x*	*(5*1h30min)*
5	Instructions import	1min	x
6	Motion start points definition	10*1min	20*1min
	Snapping enhancing	20*40sec	20*1min
	Motions creation		20*1min
	Transparent copies creation		20*10sec
7	Sequence creation		15min
	Sequence re-ordering	4min	4min

Total time estimate	37min	107min
Normalized results	0.35	1

As mentioned, it can be observed that for the scene created with Lua scripting, there was no enabling of flexible motions. Whereas for the approach without Lua scripting, it was impossible (or it would have taken an unreasonable amount of time), from the software perspective, to create instructions from the WIs that were exported from Teamcenter into IPS without Lua scripting. It is also noteworthy that for the scene created with Lua scripting, the more time consuming activity was actually executing a script (which had to be done for each parts involved in the assembly sequence).

Furthermore, as appears in the final line of Table 4.8, using Lua scripts to auto generate certain aspects of the VR training scene led to creation time reduced by 65%, compared to the time needed to create the scene manually. Moreover, that final time of scene creation of 37 minutes can be reduced in the future. In fact, as mentioned in Section 4.5, limitations were faced that led to gaps in the script. Should everything be fixed in the future, i.e., if one script was able to create the scene all at once, the creation time could be decreased as low as around 15 minutes in total. This time would then correspond to the time that the script requires to be executed and to process everything as it would be what takes the longer time. If this was to be achieved, then the creation time would be decreased by 85% compared to the manual creation scenario and even less skill would be required as the manual input of data would not be necessary anymore.

Finally, it should be noted that the VR training scene was created by the researchers, who were trained and experienced with Lua scripting, using IPS and being knowledgeable in manufacturing assembly. Thus, this knowledge and experience would be a prerequisite to achieve such a low development time. Nevertheless, the effort needed to be able to use Lua scripts is really small compared to the one needed to create Lua scripts, so overall, such prerequisites are achievable for an engineer working with manufacturing or plant launch.

5. Discussion

In this chapter, first the results are discussed in accordance with the research questions and recommendations on how to improve the results are given when it is relevant. Then, the methodology used in the project is discussed. Finally, future work topics to continue in the direction set by this project and to improve the results are presented.

5.1 Discussion of Research Question 1

RQ1: *What prerequisites are needed to implement scalable VR training scene creation and how to access the relevant data?*

As it has been seen throughout the project, at VCC, all the data required to create VR operator training scenes are already available in some databases. Thus, all the prerequisites are here at VCC. The only difficulty is to be able to access the data and to do that efficiently, which is going to be the focus of the following sub-section.

Starting with the widest scope, it has been seen that VR operator training would fit well in the operator training process at VCC as a second step in a three-step training process. The first step would be a short on-paper introduction to learn the working methods, followed by VR training, that would focus on learning the assembly sequence and the last step would be an on-the-line training at the plant, with a supervisor, to acquire the specific craftmanship. Furthermore, VR operator training would possibly be provided before the manufacturing ramp-up, as the data required mainly comes from the industrialization phase. This is interesting as it would allow to start training operators earlier.

This step back is needed for a better scalable implementation of VR training as, as was underlined by the literature study, the acceptance of using such new technology is difficult to get. Therefore, all the involved stakeholders must understand the advantages and the potential of VR operator training for it to stand a chance to thrive in the long run.

When it comes to the relevant pieces of data themselves, they must be gathered from the company's PLM system, namely Teamcenter for VCC. More precisely, the BOP, present in each CC object, contains in theory everything that is needed (unless data is missing, which as was discovered during the project, can happen). Furthermore, Teamcenter already includes tools to facilitate the extraction of specific pieces of data, such as for the .csv file that was used to generate 3D text instructions in IPS. Thus, these tools could as well be exploited. Since, this way of digital data organization is common in the industry, one could say that the results of this study can be generalized. Nevertheless, things might still be a bit different in other companies for instance, therefore, the results should be taken with more caution.

5.2 Discussion of Research Question 2

RQ2: *What effort is needed to achieve scalable VR training?*

As it has been seen when working on the auto generation of the scalability demo VR scene, several challenges would need to be tackled to achieve a proper generalized automatic creation of VR training scenes and therefore a better possibility for a scalable implementation of VR training. The main identified topics of effort needed are mentioned below.

5.2.1 Data gaps

It has been seen that some data could be missing from the data bases, for instance, neither the tools CAD nor the storage elements CAD were available in the databases. This would need to be fixed, otherwise the training scenes would not be possible to populate with everything that is needed properly.

5.2.2 Data organization of CAD assembly

Figure 4.11 highlighted that the CAD tree structure of the assembly was not in its most convenient form. In fact, there is no predefined hierarchical level that could be used as a reference to determine what geometries should be rigid bodies or not. Furthermore, some geometry elements belonging to the same level in the tree structure sometimes needed to be combined to form one rigid body. These two aspects could be fixed with a better data organization. An example of a reorganization of the tree structure in Figure 4.11 is displayed in Figure 5.1 below.

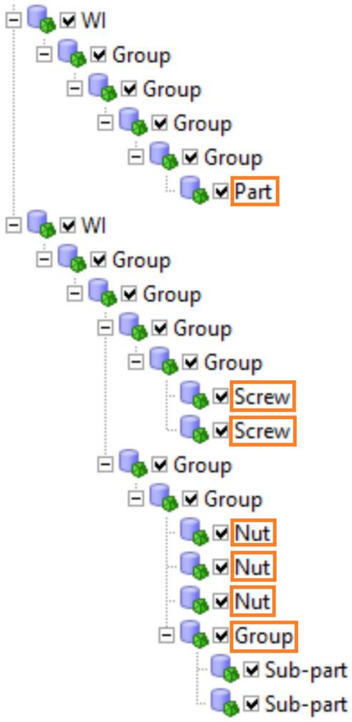


Figure 5.1: Example of tree structure data organization

This type of tree structure would allow to define the rigid bodies based on the position of the elements in the tree structure, which therefore could be done automatically more easily.

Nevertheless, this does not come without cons, as people working in the Engineering department would have to change their ways of working to implement that change, which will not be easy to achieve.

5.2.3 Denomination reworking

Another way to fix the issue with the definition of the rigid bodies and to work towards the automatic definition of the operation sequence by gathering the required data is to link the names given to the parts in the WIs step descriptions and in the CAD tree structure. As of now, the parts are named differently in the two sources, with names looking like “XX-VCCNN-XNNNNN-NN/N-HOSE XXX XXXX (view)” in the CAD tree structure and like “hose” when the same part is referred to in the corresponding WI step descriptions (and when referring to another hose in the step description, it would also be called "hose").

Thus, creating a link to associate these two different names could be beneficial. This could be done by creating a unique identification number that would be given to each part, so it could be identified easily. Another solution, as discussed in the literature study, Sub-section 2.8.2, is to find dependencies between the data objects and to unify them with a metadata model that could, for instance, be created with machine learning. In other words, a sort of dictionary would need to be created that translates each parts' name in the CAD tree structure into its name in the WI text description. Furthermore, as displayed in Figure 5.1 and 4.11, the parts are sorted by WIs, which is the same for the step descriptions so that can be used to make the process easier.

Once this is done, the order of appearance of the parts' name in the WI step descriptions can be used to automatically organize the operation sequence in IPS.

5.2.4 Storage coordinates metadata

As seen when automating the creation of the scalability demo VR scene, the coordinates of the stored positions of the parts on the racks, boxes, and storage elements in general had to be determined manually. Although, such coordinates could be included as metadata together with the CAD data. This is even more relevant as people at VCC, in the Ergonomics and Packaging departments are already working on defining such coordinates to ensure good working conditions for operators. Therefore, it would only be necessary to define a way to save these coordinates in Teamcenter, together with each CAD file and to be able to import them into IPS at the same time as the geometry itself.

5.2.5 IPS-Teamcenter interoperability

The creation of a link between IPS and Teamcenter would allow the automatic access, from IPS, with a Lua script, to data stored in Teamcenter, which would be a one-way communication. Furthermore, a two-way communication could also be beneficial as it would allow for instance to update the version of the parts in IPS when updates are made in Teamcenter.

5.2.6 IPS – extending the simulation tool's barriers

Some challenges that appeared were related to the fact that IPS is first and foremost a simulation tool and not a game engine. Because of that, the gamification of the simulation was limited, and the learning possibilities as well. To fix this, and to foster the future use of IPS as a training tool at VCC, discussions between VCC and FCC need to be conducted to communicate the needs to achieve high quality scalable VR operator training in the future.

5.3 Discussion of Research Question 3

RQ3: *To what extent can VR training pedagogical elements be auto generated?*

Based on the literature study (see Sub-section 2.5.3), the tests realized with the tryout VR scene (see Section 4.2) and the answers from interviewees (see Sub-section 4.3.2), several pedagogical elements were highlighted. Their auto generation in VR scenes is discussed in what follows.

5.3.1 Text instructions

As it has been presented in Chapter 4, the importation of the text instructions that were obtained from Teamcenter and exported as a .csv file could be achieve automatically. It was done with a script that was created to convert the column where the instructions were written and made into 3D text in IPS (see Figure 4.8). Therefore, by having text instructions in IPS, more specifically, in VR mode, the trainee could read the instructions, which facilitated the learning of the assembly sequence. Furthermore, because it contained many instructions when assembling different parts, a script was created to go to the next instructions from within VR, when the operator had finished reading and executing the assembly in accordance with the instruction. Thus, it was possible to auto generate this guidance element and it was esteemed as valuable and finally, the process to do so could be improved, as seen previously.

5.3.2 Parts' snapping

The second pedagogical element implemented was the snapping feature. It was auto generated to facilitate, for the operator, where each part should exactly be assembled, in relation to other parts. Furthermore, to make it clearer, a duplicate of the part in its assembled position, in a more transparent color, facilitated the visualization of where to assemble the part. Once the part was held close to the transparent duplicate, it was snapped into its final position. In addition, even if the operator was assembling it wrong (for example, flipping it 180 degrees in any axis), it was still snapped into the right position. The latter can be seen both as an upside or a downside because the operator would still learn the assembly sequence but not how to be precise and how to perfectly place the parts. Since VR training focuses on learning the sequence and not on acquiring any type of craftmanship, this does not appear as a problem and one can say that the auto generation of this pedagogical element was a success.

5.3.3 Assembly sequence visualization

Thirdly, to facilitate the operator training in VR, the visualization of the assembly sequence was partly auto generated by creating motions of the rigid bodies and then an operation sequence (that needed to be manually organized in a global sequence). By doing so, the operator could visually observe where the parts, that needed to be assembled, were stored and then where they should be assembled. This can be utilized to complement the text instructions, in case the operator does not fully understand them, but also to learn the assembly sequence's steps order, i.e., which part is to be assembled first and the assembly position, then which part is to be assembled next and in relation to the previous assembled ones. This pedagogical element was successfully auto generated but with the limitations detailed in Sub-section 4.5.1.

5.4 Discussion of the methodology

Throughout the thesis, the research methodologies that were implemented are from courses taken in the product development master's program at Chalmers University of Technology. They are discussed in the following sub-sections.

5.4.1 Selection methodology

To find the best solution for stakeholders / customers, the first step is to identify the customer's needs and provide a feasible solution to satisfy the defined needs. Then, matrices can be used to eliminate inferior solutions and to find the solution that is considered the most suitable one. In this study, 11 potential software solutions were identified. Commonly, many requirements and large quantities of possible solutions are identified, resulting in many solutions that are eliminated due to them not fulfilling the requirements. In this project, with 11 identified software, 2 categories (5 software solutions) were eliminated due to, from the literature review, them being inferior compared to the other software solutions and not them being bad in general. In the end, even if the two remaining solutions, Unity and IPS, were used to develop the tryout VR scene with the subwoofer, IPS was considered a better software due to the knowledge that was provided at VCC and the state of the collaboration with FCC, as well as after considering the results of the Kesselring matrix (see Sub-section 4.1.4). However, this did not affect the results since the focus was not on selecting the perfect software, but rather on developing ways for the auto generation of VR operator training scenes, together with the identification of the relevant data to do so. However, it needs to be noted that if one wanted to auto generate VR training scenes with a software different from IPS, then the findings of this project would probably need to be considered with cautious as another software could require different inputs.

5.4.2 Interviews conduction

The quantity of interviews could have affected the results. Ideally, the plan would have been to arrange an interview with all 11 software providers, to complement the literature review. In the end, solely 4 interviews were possible to realize, which allow to gather more knowledge on Emulate3D, IC.IDO, IPS, and Unity. Therefore, by having an interview to acquire more knowledge on the remaining ones, the chosen software could have differed.

In addition, when defining the pedagogical elements to facilitate operator training in VR, an interview was conducted, with a plant launch manager and launch specialist present together. By conducting one-by-one interviews with operators, other possible pedagogical elements could have been identified and explored. However, the researchers had also defined potential pedagogical elements to facilitate VR operator training beforehand and more ideas and verification of some identified pedagogical elements were provided during the interview as well. Furthermore, by letting the operator test the developed auto generated VR training scene, one could provide feedback on whether the defined pedagogical elements were valuable and if other pedagogical elements could be more suitable.

5.4.3 Developed VR operator training scene

When creating the VR operator training scene, especially when working with cables, capturing the realistic behavior of a physical cable was a difficult task. It was realized that it started to become computationally heavy when moving it and when creating the motion. Therefore, three cables that were dependent on other parts, were simplified as a rigid body to simplify the simulation as when the operator is grabbing them in VR mode, the cables inside the rigid bodies would also be grabbed. Therefore, to recreate a realistic training scene in the plant, the cables would need to be treated as flexible cables. Another obstacle that emerged was that for the three cables, when wanting to make it realistic, several rigid bodies were linked to them and because of that, it was too difficult to simulate. The solution to that was to simplify everything by creating one rigid body, reducing a lot the realism for the interaction with that part. Thus, things could have been done differently and more effort could have been put to try to fix that, but it was esteemed as secondary in the project and therefore not further studied.

5.4.4 Manufacturing knowledge acquisition

Visiting the plant and receiving video material of the assembly process, provided a better understanding on how to create a close-to-reality VR operator training scene in IPS and was deemed essential. Furthermore, to improve the created training scene, feedback from a domain lead simulation engineer was utilized. Receiving feedback from someone with experience contributed to the project by focusing more on identifying the elements that should be focused on for the creation of the VR operator training scene. This underlines that to create such VR scenes and to work towards their auto generation, a strong background is needed in manufacturing and production, as well as knowledge more related to the used technologies, including VR technologies in general and software such as IPS. Hence, this aspect should not be neglected when planning the large-scale implementation of VR operator training. The persons required for that would need to be close to the assembly processes as well as knowledgeable in simulation software and in scripting to achieve the auto generation aspect.

5.5 Ethical considerations throughout the process

According to Kazmierska (2020), regarding qualitative studies, the author highlights that because the researchers plan the process on how to approach the study, it is therefore not objective mainly since the researchers are selecting the participants for the study. Thus, the researchers want to declare that throughout the study, everything has been conducted as objectively as possible and that the subjective interferences with the results were limited as much as possible. For instance, to try to reduce the biases, well known decision matrices were used, such as the Kesselring and Pugh matrices. Although, it cannot be said that they ensure a total objectivity.

5.6 Future work: improvements of the training

In this section, further improvements of the VR operator training are presented, divided into topics including ergonomics, pedagogy, and realism of the training.

5.6.1 Ergonomics

One future work would be to auto generate VR operator training scenes that include an ergonomic perspective. This would require importing a manikin into the IPS scene and considering the different heights for both sexes. Furthermore, one would also need to consider the movements of the manikin in relation with how humans are moving. One can simulate that by adjusting the joints of the manikin and create different postures / positions that the manikin can take. Another option would be to use body trackers and record how the human moves then have the manikin to move in a similar fashion. This would allow to warn to the trainee when he / she is lifting, placing, or moving parts in an unergonomic way, so good habits can be taken from the beginning. Another interesting aspect to explore in the future would be to use AI (Artificial Intelligence) capture cameras. Using this to record an operator, performing an assembly in VR, the AI can create a stick figure of the operator and calculate the postures numerically and display it with colors (red, yellow, and green), where red indicates a bad posture and green indicates a good posture. Therefore, the engineers involved can get a real-time ergonomic data that can be used to evaluate the movements and the posture of the operator while the latter performs tasks in a safe way as there are no heavy parts involved.

5.6.2 Pedagogical considerations

Another future work would be, from the perspective of IPS, to be able to highlight the part that needs to be picked with the same color as where it is to be assembled to follow the assembly sequence's order. It could facilitate the operator training, as it would make the sequence easier to follow, especially if parts are stored close to another and are looking similar. This represents an important pedagogical element that could not be implemented.

Another future work could be to integrate videos and pictures. To complement the text instructions, a video / picture can be integrated for clarification, to explain better to the operator how parts should be assembled. Furthermore, if there are too many instructions to read, the operator can get tedious. Therefore, a possibility could be to complement or replace some with videos or pictures to better illustrate, step by step, the assembly sequence.
In addition, arrows could also be utilized to complement the highlighting of the part with colors and auto generated to locate better which part to pick. Therefore, misunderstandings can be mitigated. On the other hand, for the operator, it can be perceived as redundant to have both the arrows and the highlight of the part to be picked. Hence, further studies on creating such pedagogical elements for VR training would need to be conducted, together with tests by operators to get to know what the main stakeholder's opinion is.

The last future work towards more pedagogy in the training could be to integrate different levels of difficulty, with and without guidance. For a new operator, more pedagogical elements would be required, compared to what would be needed for an experienced one. Indeed, it can facilitate the learning of the assembly sequence for the trainee to have text instructions, and visual guidance elements indicating how to assemble the different parts. Whereas for an experienced operator, less guidance or even no guidance would be necessary for the operator to know how the parts need to be assembled. One pedagogical element that could be beneficial in that case would be the visualization of the snapping zones, otherwise, the operator would spend too much time trying to place the parts as perfectly as possible in relation to one another.

5.6.3 Realism of the VR environment

Throughout the study, it was discovered that when observing the point clouds database in PlantView, the 3D scans were too old, hence, they did not match with the current plant's layouts. Therefore, exporting point clouds from PlantView and importing them into IPS was possible, but it would bring a fake realism. Therefore, if the point clouds stored in the software were up to date, it would be possible to generate a perfectly realistic environment for the operator training, hence, facilitating the learning of the assembly sequence in a realistic environment but it was not the case. Furthermore, a study would need to be realized to determine if such level of realism is necessary or not for the trainee to learn the assembly sequence and in case of the training for the assembly of a new car model, in a new plant, the 3D scans would not be available anyway.

Instead of scanning the plant every time changes are made in a station, a possible solution could be to create the point cloud of a generic plant, which could then be used as the base for the VR training scenes. Then, this base environment would just need to be complemented with the CAD furniture associated with each station. With that, when the station's layout is changed due to, for example, the positioning of the trolleys, fixtures, tools, etc. the new relative position of each element would have to be stored in Teamcenter and could then be imported into IPS with the positions matching where it is in the plant. As a result, every time the station's layout changes, no new scans would be needed, as long as the generic 3D plant layout is saved and updated with the data for the positions of the equipment and tools stored in Teamcenter, relative to the concerned station.

Therefore, what needs to be done is, first, to create such a generic plant environment by taking into consideration how it should look to be able to use it for most of the assembly stations. The second thing to do is to implement the saving, in Teamcenter of the different stations' layout, including the positions of the storage furniture. Together with that, as discussed in Sub-section 5.2.4, for each storage element, metadata indicating where parts can be stored in / on it also needs to be saved. Once this is achieved, realistic and data-rich VR operator training environments, for each assembly station could be auto generated.

5.6.4 Sustainability considerations

Nowadays, it is important to discuss the consequences of technologies and activities, both on humans and on the environment.

When it comes to the impact of VR training on humans, it was experimented by the researchers that minor motion sickness could be experienced while wearing a VR headset for the first times. Although, this discomfort disappeared when getting use the headset, so it was not seen as a major issue. Another impact of VR training on the operators, as discussed during the interview with the persons from plant launch, is the fact that they can benefit from it as they would be more confident when they start their on-the-line training. They would already be familiar with the stations' layout and with the assembly sequences, which would facilitate their integration in the demanding environment they are entering.

VR operator training comes with pros but also with cons when pondering upon its implications on the environment. As a matter of fact, its large-scale implementation would require manufacturing an important amount of VR hardware, which does not come without an environmental cost, in term of material requirement and energy use. Nonetheless, it should be noted that VR hardware can be used several times and by several persons. It should also be kept in mind that VR operator training would replace other training activities and because of the virtual nature of this type of training, it would reduce the need of physical parts needed for training, and it would also lead to less transportation of people, such as trainees and trainers. All in all, a more thorough study on the environmental impacts of this type of activity would need to be conducted, such as a lifecycle assessment comparing the new type of training, where VR training would be implemented to the traditional operator training, without VR technologies.

6. Conclusion

The generalization of VR operator training seems inevitable nowadays in the industry 4.0, especially when looking at its advantages compared to the steps it is destined to replace in the operator training process. Nevertheless, as of now, one of the barriers to its implementation is the time necessary to create the VR training scenes to be used for operator training. That is why, working towards their auto generation appears as a crucial goal.

Hence, it has been seen in this study that using scripting to generate VR scenes could reduce their creation time by 65% as of today. It could even be decreased by up to 85% in the future, should the challenges highlighted by the study be addressed. Furthermore, VR operator training is complex to implement as it requires data of different natures, coming from many sources and involving various types of stakeholders. That is why there are many challenges to its large-scale implementation, related with the tree structure organization of CAD data, with missing links between databases, with the inability to reuse other teams' work or with software interoperability and software limitations. Another consideration to keep in mind is the broad range of skills needed to work on the large-scale implementation, as it requires scripting abilities, knowledge in VR and in the manufacturing processes. Finally, once a VR operator training is created, it also needs to be of high quality, including relevant pedagogical elements, so it gives value to the operators. Despite all these challenges, this study showed that auto generating certain elements of VR operator training scenes is already feasible, which includes the core of the mechanisms of the scene (enabling interactions with the parts in VR) as well as pedagogical elements (visual cues, text instructions). The study also showed that such practices can be improved and are overall really promising.

In conclusion, it boils down to how to create efficiently a virtual environment, including pedagogical elements, where the operator can benefit from VR training. Therefore, it is of high interest to let the operators test the auto generated VR training scenes to receive as much feedback as possible, to facilitate their learning even more.

One also needs to consider, from a scalability aspect, to achieve a certain level of social acceptance and to convince the higher management that investing in VR operator training would be beneficial. It has been demonstrated in this study that VR operator training scenes can be created efficiently and that the required data are already available. The more efforts will be put to fix the identified challenges, the most performant the creation process will become but this does not come for free. Therefore, one also needs to consider the cost required to implement VR operator training, which includes equipment, personnel, education, etc. Because when implementing VR for training operator, the company needs to have enough evidence and feel convinced that it can be beneficial to all the involved stakeholders in the long run.

7. References

Amazon. (n.d.). *HTC Vive Pro full kit Reality System*. Retrieved March 8, 2024, from https://www.amazon.se/HTC-VIVE-Full-Reality-System/dp/B07D9T6H1F

Badamasi, A. A., Aryal, K. R., Makarfi, U. U., & Dodo, M. (2022). Drivers and barriers of virtual reality adoption in UK AEC industry. *Engineering, Construction and Architectural Management*, *29*(3), 1307-1318.

Bell, E., Bryman. A., & Harley, B. (2019). *Business research methods* (5th ed.). Oxford University Press.

Berg, L. P., & Vance, J. M. (2017). Industry use of virtual reality in product design and manufacturing: a survey. *Virtual reality*, 21, 1-17.

Boström, Jim, Zamola, Sebastian (2023). *Method to create XR-supported work instructions: Taking the step toward a more sustainable plant.* http://hdl.handle.net/20.500.12380/307149

Carneiro, J. A. S., Cardoso, A., Santos, F. P., Milagre, S. T., Lamounier, E. A., & Oliveira, L. C. (2023, June). Ethical Aspects in a Virtual Reality Environment. In *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-4). IEEE.

Carter, J. (April 8, 2024). *New Product Development Process: Faster Product Development. A Roadmap to Your Next Successful Product*. TCGen. Retrieved Mai 13, 2024, from https://www.tcgen.com/product-development/process/

Choi, S., Jung, K., & Noh, S. D. (2015). Virtual reality applications in manufacturing industries: Past research, present findings, and future directions. *Concurrent Engineering*, *23*(1), 40-63.

Davies, A. (2022). *10 Great Tools for VR Development*. Retrieved March 13, 2024, from https://www.devteam.space/blog/10-great-tools-for-vr-development/

Dawson, C. (2007). *A Practical Guide to Research Methods*. 3rd ed. Oxford: How to Books Ltd.

Denscombe, M. (2014). *The Good Research Guide: For Small-Scale Social Research Projects (UK Higher Education OUP Humanities Social Sciences Study S)* (Revised). Open University Press.

Doolani, S., Wessels, C., Kanal, V., Sevastopoulos, C., Jaiswal, A., Nambiappan, H., & Makedon, F. (2020). A review of extended reality (XR) technologies for manufacturing training. *Technologies*, *8*(4), 77.

Eiden, A., Eickhoff, T., Gries, J., Göbel, J. C., & Psota, T. (2021). Supporting semantic PLM by using a lightweight engineering metadata mapping engine. *Procedia CIRP*, *100*, 690-695.

Emulate3D. (n.d.). *Every minute you lose is one you'll never get back*. Retrieved March 13, 2024, from https://www.emulate3d.com/

ESI. (n.d.). *IC.IDO Virtual Reality Engineering Software.* Retrieved March 13, 2024, from https://www.esi-group.com/products/ic-ido

FCC. (n.d.). *Industrial Path Solutions.* Retrieved March 13, 2024, from https://www.fcc.chalmers.se/software/ips/

Fonseca L. (April 21, 2022). *Writing a Work Instruction: A Complete Guide*. Venngage. Retrieved from https://venngage.com/blog/category/human-resources/

Fowler, C. (2015). Virtual reality and learning: Where is the pedagogy?. *British journal of educational technology*, 46(2), 412-422.

Garcia, A. D., Schlueter, J., & Paddock, E. (2020). Training astronauts using hardware-in-the-loop simulations and virtual reality. In *AIAA Scitech 2020 Forum* (p. 0167).

Hayat, M., & Winkler, H. (2022, December). Exploring the basic features and challenges of traditional product lifecycle management systems. In *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 0762-0766). IEEE.

Healthy Mind. (August 25, 2021). *Virtual reality headset: how does it work?.* Retrieved March 13, 2024, from https://healthymind.fr/en/virtual-reality-headset-how-does-it-work/#:~:text=VR%20glasses%20work%20by%20stereoscopy,in%20front%20of%20each%20eye.

Holuša, V.; Vanek, M.; Beneš, F.; Švub, J.; Staša, P. (2023). Virtual Reality as a Tool for Sustainable Training and Education of Employees in Industrial Enterprises. *Sustainability*, 15(17), 12886.

IBM. (n.d.). *What is an API (application programming interface)?* Retrieved April 30, 2024, from https://www.ibm.com/topics/api

IBM. (n.d.). *What is artificial intelligence (AI)?* Retrieved Mai 28, 2024, from https://www.ibm.com/topics/artificial-intelligence

IBM. (2021). *SDK vs. API: What's the Difference?* Retrieved April 30, 2024, from https://www.ibm.com/blog/sdk-vs-api/

Inlean. (n.d.). *Manufacturing Operations*. Retrieved May 16, 2024, from https://inlean.com/manufacturing-operations/

Isar, C. (2018). A glance into virtual reality development using unity. *Informatica Economica*, 22(3), 14–22.

Johannesson, H., Persson, J. G., & Pettersson, D. (2013). *Produktutveckling: effektiva metoder för konstruktion och design*. Liber.

Lin, J., Cao, L., & Li, N. (2020). How the completeness of spatial knowledge influences the evacuation behavior of passengers in metro stations: A VR-based experimental study. *Automation in Construction*, *113*, 103136.

Maples-Keller, J. L., Bunnell, B. E., Kim, S. J., & Rothbaum, B. O. (2017). The use of virtual reality technology in the treatment of anxiety and other psychiatric disorders. *Harvard review of psychiatry*, *25*(3), 103–113.

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons, Incorporated.

Mohd, T. K., Bravo-Garcia, F., Love, L., Gujadhur, M., & Nyadu, J. (2023). Analyzing strengths and weaknesses of Modern Game Engines. *International Journal of Computer Theory and Engineering*, *15*(1), 54-60.

Niemann, J., Pisla, A., Niemann, J., & Pisla, A. (2021). Teamcenter data management. *Life-Cycle Management of Machines and Mechanisms*, 381-396.

Patle, D. S., Manca, D., Nazir, S., & Sharma, S. (2019). Operator training simulators in virtual reality environment for process operators: a review. *Virtual Reality*, *23*, 293-311.

Penciuc, D., Durupt, A., Belkadi, F., Eynard, B., & Rowson, H. (2014). Towards a PLM interoperability for a collaborative design support system. *Procedia Cirp*, *25*, 369-376.

Philippe, S., Souchet, A. D., Lameras, P., Petridis, P., Caporal, J., Coldeboeuf, G., & Duzan, H. (2020). Multimodal teaching, learning and training in virtual reality: a review and case study. *Virtual Reality & Intelligent Hardware*, 2(5), 421-442.

Prospect. (n.d). *The #1 VR Platform for the Building Industry.* Retrieved March 13, 2024, from https://irisvr.com/

PTC. (n.d.). *CAD software solutions*. Retrieved April 30, 2024, from https://www.ptc.com/en/technologies/cad

Raptis, T. P., Passarella, A., & Conti, M. (2019). Data management in industry 4.0: State of the art and open challenges. *IEEE Access*, 7, 97052-97093.

Rauschnabel, P. A., Felix, R., Hinsch, C., Shahab, H., & Alt, F. (2022). What is XR? Towards a framework for augmented and virtual reality. *Computers in human behavior*, 133, 107289.

RISE. (September 15, 2022). *PLENUM: PLENary multi-User developMent arena for industrial workspace*. Retrieved January 19, 2024, from https://www.ri.se/en/what-we-do/projects/plenum-plenary-multi-user-development-arena-for-industrial-workspace

Rouse M. (January 4, 2017). *Head-Mounted Display*. Technopedia. Retrieved April 30, 2024, from https://www.techopedia.com/definition/2342/head-mounted-display-hmd

Sagnier, C., Loup-Escande, E., Lourdeaux, D., Thouvenin, I., & Valléry, G. (2020). User Acceptance of Virtual Reality: An Extended Technology Acceptance Model. *International Journal of Human–Computer Interaction*, *36*, 1007 - 993.

Schwebel, D. C., Gaines, J., & Severson, J. (2008). Validation of virtual reality as a tool to understand and prevent child pedestrian injury. *Accident Analysis & Prevention*, *40*(4), 1394-1400.

Schönfelder, R. (2003). VR/AR—Applications, Limitations and Research in the Industrial Environment. In *Human Behaviour in Design: Individuals, Teams, Tools* (pp. 253-260). Berlin, Heidelberg: Springer Berlin Heidelberg.

Siemens. (n.d.). *Bill of process*. Retrieved May 16, 2024, from https://www.sw.siemens.com/en-US/technology/bill-of-process/

Siemens. (2019). *Create a new work package*. Retrieved from https://docs.plm.automation.siemens.com/content/easyplan/2.11/WIA/en_US/easy_plan_211_work_instructions_authoring/working_with_work_packages/create_a_new_work_package.html

Siemens. (n.d.). *Virtual Training Solutions*. Retrieved March 13, 2024, from https://www.siemens.com/global/en/products/services/digital-enterprise-services/training-services/vizendo-virtual-training-solutions.html

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of business research*, *104*, 333-339.

Sobhani, A., Farooq, B., & Zhong, Z. (2017, October). Distracted pedestrians crossing behaviour: Application of immersive head mounted virtual reality. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-6). IEEE.

Stack overflow. (2014). *Best database structure to implement a graph tree.* Retrieved March 19, 2024, from https://stackoverflow.com/questions/40838726/best-database-structure-to-implement-a-graph-tree

Talentverse. (n.d). *10 Best Tools for VR Development of 2023*. Retrieved February 10, 2024, from https://www.talentverse.co/blog/best-tools-for-vr-development

Technopedia. (n.d). *What is Product Lifecycle Management (PLM)?* Retrieved April 30, 2024, from https://www.technia.se/blog/what-is-product-lifecycle-management-plm/

Ulrich, K., Eppinger, S., & Yang C, M. (2019). *Product Design and Development.* McGraw Hill.

Unity Asset Store. (n.d.). *VR Builder – Open-source toolkit for VR creation.* Retrieved March 8, 2024, from https://assetstore.unity.com/packages/tools/visual-scripting/vr-builder-open-source-toolkit-for-vr-creation-201913

Unity. (n.d.). *Products: Unity Engine*. Retrieved February 29, 2024, from https://unity.com/products/unity-engine

University of Silicon Valley. (October 20, 2020). *What is a game engine?* Retrieved from https://usv.edu/blog/what-is-a-game-engine/

Valem Tutorials. (2022). *Make VR Game Without Code – VR Builder Tutorial – PART 1*. https://www.youtube.com/watch?v=ObYdlU68MYw

Vive. (n.d.). *Base station*. Retrieved February 28, 2024, from https://www.vive.com/eu/accessory/base-station/

Volvo Cars. (2024). *Volvo car models*. Retrieved May 15, 2024, from https://www.volvocars.com/au/l/model-lineup/

Volvo Cars. (2024). *Who we are*. Retrieved April 30, 2024, from https://www.volvocars.com/intl/v/our-story

Voštinár, P., Horváthová, D., Mitter, M., & Bako, M. (2021). The look at the various uses of VR. *Open Computer Science*, *11*(1), 241-250. https://doi.org/10.1515/comp-2020-0123

VR Owl. (n.d.). *The 22 best examples of how companies use virtual reality for training.* Retrieved January 26, 2024, from https://www.vrowl.io/the-22-best-examples-of-how-companies-use-virtual-reality-for-training/

Wheelwright, S. C., & Clark, K. B. (1992). *Revolutionizing product development: quantum leaps in speed, efficiency, and quality*. Simon and Schuster.

3DEXPERIENCE. (n.d.). *Immediately improve the understanding of your design with integrated & direct access to immersive Virtual Reality inside your 3DEXPERIENCE applications.* Retrieved March 13, 2024, from https://my.3dexperience.3ds.com/welcome/compass-world/rootroles/virtual-reality-reviewer?role=true

8. Appendix

A: Interview questions for software providers

General thoughts

1. Do you think that the creation of pedagogical VR operators training scenes is achievable with this software? I.e., environments where the user can pick up pieces, use tools, with information displayed to help, elements highlighted...
2. How long does it take to create a VR environment with a detail to the nut-level?
3. Does the software only allow to grab and place pieces? Or is it possible to use tools in the simulation? E.g., simulate drilling, screwing with which the screw is moving...
4. Is the software user-friendly for developers that are not very familiar with VR?
5. Is the software compatible with different VR platforms / headwear?
6. Does the software support multi-users?

Data and PLM

7. How convenient is it to import files from Catia and other software?
8. Do you know any previous work on integration of this software with PLM systems?
9. Is it possible to achieve or work towards automatic training environment creation? E.g., automatic access to data and identification of relevant sub-systems and the link between them.

SDK and scripting

10. Are there SDKs for usual / conventional interactions (features) with objects (grab, place, highlight...)? If it is needed.
11. How free are we to create new specific actions for the user (e.g., with a scripting language associated with the software)?
12. Are there any specific frameworks or coding knowledge recommended for scalability?
13. Is it possible to create an interface that easily allows the creation of VR training scenes for someone who does not know how to code or does not have an expertise within this software? I.e., creation of a plugin / SDK.
14. How does the software handle update for VR applications from a scalability perspective?

B: Selection matrices input

B.1 Justification of the selection criteria

Availability of tutorials

As this project requires creating VR scenes in the first place and since the project team has no experience in any of the software, the availability of tutorials is crucial. The tutorials can be of different nature, such as courses or seminars organized by the software provider itself or videos on YouTube created by third parties.

Easy to use for the training (operators)

The VR scenes are created for the training of assembly operators. Therefore, they should be beneficial to them and be admirably adapted. That means that even with a limited experience in VR, operators should be able to use the scenes and they should benefit from the training.

Easy to implement (engineer)
The software can be more or less easy to use, and it can take a shorter or longer time to get familiar with it, both for the project team and for the engineers that would use it in the future, once the scalability of the solution is achieved.

Easy to maintain (updates)
The software can have updates that occur more or less often. If it is updated often, then there are less risks of bugs and there are new features that can be added more quickly, which can be beneficial for the creation of training environments.

Connection with PLM system (Teamcenter)
One of the project's goals is to integrate the chosen software with the PLM system used at VCC, i.e., Teamcenter. Therefore, a software that is already or can be easily integrated with Teamcenter should be preferred as it would allow to import more easily the data needed for the creation of the environments.

Possibility of automation of the creation
The focus of the project is on the scalability of the training of operators using VR. Since the way to achieve that is through working towards the automation of the VR scene creation, it is valuable if the software provides such opportunity, one way or another.

Quickness of scenario creation
It seems as of now that not the whole scene and all the interactions possible in it can be generated with only one click. Although in the best case it would eventually be, the manual creation of some part of the environment is to be considered and it should be as quick as possible.

Level of personalization (add personal features)
It has been identified that the different solutions do not present all the features that would be needed to create superior quality training environments. Therefore, they would have to be personalized, which is, most of the time, achievable via scripting, more or less easily and with more or less restrictions.

Availability of the software
VCC already has valid licenses for some software, for some ones, it does not and same for Chalmers. This needs to be considered for the selection of the software as it can be a barrier to quick implementation.

Pedagogical options (highlight elements, add arrows...)
For the VR training scenes to be of superior quality, guidance of the trainee should be realized throughout the sequence, thus arrows, color changes and likewise elements are needed. The different solutions make the creation of such elements more or less easy so it should be considered.

Multi-user possibility
Some parts of the assembly sequence require several operators to work together. That is why, eventually, it should be possible to have more than one operator in a VR training environment, so it is as realistic as possible. A software that provides multi-user possibilities should then be preferred.

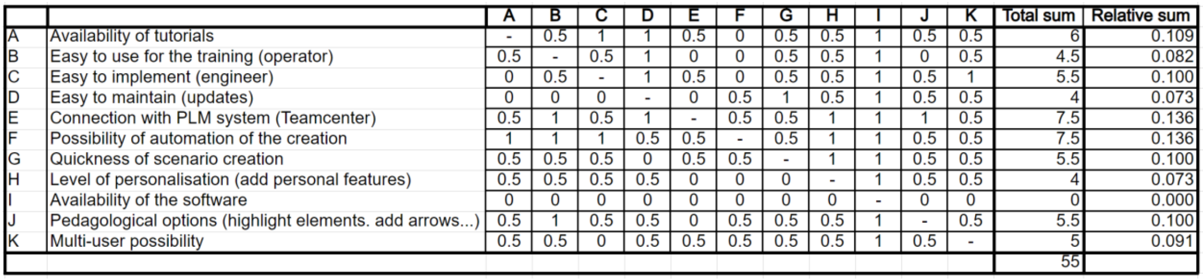
B.2 Grading scale of the selection criteria

Criteria	Grading scale (1-5)				
Availability of tutorials	Nothing available	A draft of the tutorial is available	Subscription (access to tutorial in specific area)	Subscription (tutorials in most areas)	Full access (all tutorials available)
Easy to use for the training (operator)	Very difficult (need experience/supervisor to help them)	Difficult	Neutral	Easy	Very easy (no experience in VR needed)
Easy to implement (engineer)	> 9 weeks	7-8 weeks	5-6 weeks	3-4 weeks	< 2 weeks (to be up and running)
Easy to maintain (updates)	> 6 months	5 months	4 months	3 months	< 2 months
Connection with PLM system (Teamcenter)	Impossible	Difficult to achieve	Neutral (not explored, might be doable)	Easily doable	Already available / possible
Possibility of automation of the creation	Impossible	Difficult to achieve	Neutral (not explored, might be doable)	Easily doable	Already available / possible
Quickness of scenario creation	> 6 hours	4-5 hours	3-4 hours	1-2 hours	< 1 hour
Level of personalization (add features)	Impossible	Difficult to achieve	Neutral (not explored, might be doable)	Easily doable	Already possible (button in the software)
Availability of the software	Very poor	Poor	Fair	Good	Excellent
Pedagogical options (highlight elements, add arrows...)	Nothing available and no possibility to add some	Restricted options, impossible to add more	Flexible	Free (one coding language compatible)	Very free (combine different coding language)
Multi-user possibility	Very poor	Poor	Fair	Good	Excellent

Further explanations on two of the grading scales are displayed below:

Criteria	Very poor	Poor	Fair	Good	Excellent
Availability of software	The software is not accessible by any means.	Licenses are not available at VCC nor at Chalmers. Moderate difficulty in obtaining licenses quickly at Chalmers.	Licenses are not available at VCC nor at Chalmers. Moderate difficulty in obtaining licenses quickly at VCC and at Chalmers.	Licenses are readily available at Chalmers but not at VCC. Obtaining licenses is easy and convenient.	Licenses are already available and used at VCC and at Chalmers. Obtaining licenses is effortless, the software is free.
Multi-user	Impossible to have more than one user in a VR scene.	It is possible to have several users in the same environment, but they cannot interact.	Basic collaboration features are present but lack some depth or customization options. Setting up and joining multi-user sessions is a bit difficult.	Collaboration features enable interaction and teamwork but with some limitations. Setting up and joining multi-user sessions is rather easy.	Advanced collaboration features facilitate interaction, fostering teamwork. Setting up and joining multi-user sessions is effortless.

B.3 Weighting of the selection criteria



		A	B	C	D	E	F	G	H	I	J	K	Total sum	Relative sum
A	Availability of tutorials	-	0.5	1	1	0.5	0	0.5	0.5	1	0.5	0.5	6	0.109
B	Easy to use for the training (operator)	0.5	-	0.5	1	0	0	0.5	0.5	1	0	0.5	4.5	0.082
C	Easy to implement (engineer)	0	0.5	-	1	0.5	0	0.5	0.5	1	0.5	1	5.5	0.100
D	Easy to maintain (updates)	0	0	0	-	0	0.5	1	0.5	1	0.5	0.5	4	0.073
E	Connection with PLM system (Teamcenter)	0.5	1	0.5	1	-	0.5	0.5	1	1	1	0.5	7.5	0.136
F	Possibility of automation of the creation	1	1	1	0.5	0.5	-	0.5	1	1	0.5	0.5	7.5	0.136
G	Quickness of scenario creation	0.5	0.5	0.5	0	0.5	0.5	-	1	1	0.5	0.5	5.5	0.100
H	Level of personalisation (add personal features)	0.5	0.5	0.5	0.5	0	0	0	-	1	0.5	0.5	4	0.073
I	Availability of the software	0	0	0	0	0	0	0	0	-	0	0	0	0.000
J	Pedagogical options (highlight elements. add arrows...)	0.5	1	0.5	0.5	0	0.5	0.5	0.5	1	-	0.5	5.5	0.100
K	Multi-user possibility	0.5	0.5	0	0.5	0.5	0.5	0.5	0.5	1	0.5	-	5	0.091
													55	

C: Interviews transcripts

C.1 Interview transcript IC.IDO

General thoughts

1. Yes, grabbing, color-highlighting, snapping, ghosting, displaying contact forces, etc. are available already.
2. For beginners, up to three hours and for experts, around one hour.
3. As long as there are CAD files for the parts, it will work. We have buttons where you can create the operations / sequences.
4. Yes, no coding is needed. You just need to press the buttons.
5. Yes, it is compatible with HTC Vive / Vive Pro, Oculus Quest 2, etc.
6. Yes, you can look from the first-person view but also be several people in the same VR environment.

Data and PLM

7. Very convenient, our software is compatible with many file formats from Creo/Catia.
8. We have currently started exploring it with Windchill.
9. Yes, we are currently exploring that.

SDK and scripting

10. SDKs are not available.
11. Scripting is not available. We can discuss with our customers about the features they want to implement and our company can assist them with that.
12. No coding is needed, just pressing the buttons.
13. SDKs are not available. We have buttons that is easy to use to create VR scenes and manuals describing what each button does.
14. It depends, small updates from the customers side is doable but for bigger software updates, that will take some time.

C.2 Interview transcript Emulate3D

General thoughts

1. Yes, many features are already in the software and if our customers wants something tailored only for them, we can help them.

   Yes, we think you will be able to achieve what you want with help of Emulate3D.
2. It varies from person to person but for beginners, maybe three hours and for experts, around one hour.

   It is depending on if you have ready-made CAD-models We have already now demos that can be used for demonstrating purpose. By using Point-cloud specific environment can be imported. With help of machine builder CAD files authentic experience is granted.

3. Yes, that is possible but also other features, such as snapping, handling of cables, ergonomics, write on board, texts, etc.
4. Yes, no coding is needed. Operators with no experience get used quite easily to VR with this software. Collaboration with robots has been implemented on the beta-version but improvements need to be made.
5. Yes, it is compatible with several headwear, such as Oculus Quest 2, HTC Vive / Vive Pro, etc.
   Regarding CAD-platform we have very wide integration for SolidWorks, Autodesk and Creo. But can import CAD files from the known CAD-suppliers.
6. Yes, it supports multi-users, and you can also work on multi locations by utilizing cloud connectivity.

Data and PLM

7. Emulate3D supports many different file names, such as .jt, .stl, .iges, .stl, etc. it is also possible to import CAD files from the point cloud as well.
8. In general, we have very good connectivity through Windchill / Creo.
9. We have a catalogue where the user can choose the operations that they want to do.

SDK and scripting

10. We don’t have SDKs but features, such as grabbing, highlighting elements with changing in colors, snapping, etc. are available in the software.
11. Very free. You can use scripting to tailor physical behavior, manipulate automation, build in challenges etc.
12. Coding can be done with C# or a simple scripting language.
13. We have big catalogue of objects and objects can be bundled together for easy implementation.
14. If the customer wants something tailored only for them, we can help them with that.

C.3 Interview transcript Unity User

General thoughts

1. Yes! As Unity is meant for development and games, anything is possible (as long as you can code it or find plugins for it).
   Unity (and other game engines) would be the best tool to use to make any kind of customized interactions or information displays (but comes at the cost of coding).
   All kinds of interaction, information and other training elements can be achieved with very high flexibility.
2. In the case of unity most of the detailed models etc. are imported from CAD, so it all depends on how detailed the CAD models are and how well the CAD data is structured and how well you manage to convert that to Unity. In best of worlds the CAD data should have the correct translation data from the start (position, rotation, scale, pivot points etc.) then everything should fall into the correct position as soon as it is imported into Unity.
   Most likely this is not the case however, and we will need to reposition and update the data (Hence good data structure and way of working with the PLM data is key for this project)
3. Yes, using SDKs, such as “VR Builder” you can achieve many features like grabbing, snapping, changing colors, etc.

Another SDK for interaction with objects and the environment is the "XR Interaction Toolkit".

4. Some program experience or work with game engines is recommended! There are a lot of plugins available to make it easier and more drag and drop, but most likely they would need to be adapted or changed using code to fit the purpose.
   On the scale of difficulty (in my opinion) It is somewhere in between an engineering software (like IPS and ICIDO) and pure coding.
5. Yes, all the commercial and major ones (HTC, Meta, Apple, Microsoft…)
6. Not out of the box, but a lot of plugins are available to support this (most of them developed for multiplayer video games but works great for VR and engineering purposes too!)
   Will need a bit more coding to get it to work and is a bit tricky, but it is used by some of the major game titles, so it is robust. Photon is the one that I have used for the Chalmers demonstrators (https://www.photonengine.com/pun#)

Data and PLM

7. Unity is not a 3D modeling software and recommendation is not to create any object in Unity itself, but to import this from a real CAD or 3D modeling software (Blender most used together with Unity). Materials and textures in unity is very flexible and could reach good results quickly, but ideally this data would be automatically fetched from the PLM for this project. Depends on what you would like to do and how interactive the environment should be. Just to populate an environment is "easy" and drag and drop if the input data is good (correctly formatted CAD data, scale and position data of CAD data is accurate in relation to each other). Setting it up for VR is fairly easy and unity comes with VR compatibility out of the box with some minor settings and tutorials to follow to make it work correctly.
8. Not that I personally have seen, but there are a lot of publications and news articles looking into this. Your best bet would be to look into PIXYZ, it is a conversion software that works between PLM data and Unity, and some claim that they have been able to integrate this fully:https://blog.unity.com/engine-platform/uniting-real-time-3d-and-product-lifecycle-management-for-the-first-time (Most likely this takes a lot of coding and integration work, not something that would work out-of-the-box)
9. Yes, with the help of scripting, I believe it is possible. It's only your imagination and scripting that will set the boundaries.

SDK and scripting

10. Not that I'm aware of, but there is a lot of plug-ins available so would not be surprised if this exists. Again, there is a lot of plug-ins available and several that would be able to do this. The main one that I have used is VR-Builder (https://www.mindport.co/vr-builder), it makes VR training instructions with drag and drop.
11. 100% free, only your imagination and code skills set the limit. All code is written in C# and if you are a skilled coder you can most likely automate a lot of the steps and integrate a lot of functionality to the PLM using code. But it would require a lot of care and support and might not be the most robust way for scalability.
12. As mentioned before, depending on how good you are with C#, you can achieve basically everything.
13. Yes, if you are skilled with Unity you could probably developed your own interface within Unity to automate a lot of VR creation or training instruction creation etc. Similar to what VR builder have done but even more user friendly for the user to in real time in a virtual interface to import objects, interaction and instructions.

Most likely this will take a lot of coding and some good data structure from the PLM system or the source data.

14. Unity is updated frequently to be compatible with the VR features and hardware. However, this might also be a risk for scalability as newer versions of unity might need some code changes to work properly and some continuous support and bug fixes are needed.

C.4 Interview transcript IPS

General thoughts

1. Yes, almost everything is doable, it is your scripting and imagination that will set your limit.
2. It varies a lot depending on the project, from an hour to a couple of hours depending on your knowledge with the software.
3. As long as there are CAD files for the tools, a sequence of the drilling and screwing are doable.
4. In my opinion, it is user-friendly. Most of our customers don't need to script themselves because the software provides sufficient support for them to do their simulations.
5. Yes, for example, HTC Vive / Vive Pro, Oculus, Metaquest, etc.
6. Yes, you can be several people in the VR scene. Only one computer per VR "person" is available, so two or more people trying to connect to the VR with the same computer is not available at the moment.

Data and PLM

7. Very convenient, our software supports many CAD files. Also, you can use Excel to link it to IPS.
8. Yes, many of our customers recently has been demanding trying to connect their PLM system with the software.
9. Yes, only your scripting knowledge and imagination will set your boundaries.

SDK and scripting

10. Features, such as grabbing and snapping is available in the software. For personalized features, like changing colors, HMDs, etc. will require coding and CAD files.
11. It is not possible to make the manikin to move differently because we have used a mathematical model to make it consider ergonomic so that when it moves, it is from an ergonomic perspective. Apart from that, the customer is very free.
12. The scripting language that we use is Lua.
13. Yes, our idea with this software if to create motions and sequences without coding experience, which we believe has been achieved. But if the customer wants to add personalized features, then coding is required or if the customer consults with us, we can help them out.
14. For tailored updates from the customer, that is possible as long as it is a small update. If it is a big update, we will include it as our software update, which updates twice a year.

C.5 Interview with PLM / Digitalization / Method Engineer, Domain Lead Connected Operator

How is the data structured in Teamcenter?

At the ME (Manufacturing Engineering), the first step is the CC (collaboration context), which is the car project in Teamcenter. Then, the CC is divided into three parts, BoM, BoE, and BoP. In the BoM (Bill of Materials), you can find all the car parts, such as BiW (body in White), FA

(Final Assembly), glove components, etc. In the BoE (Bill of Equipment), there are the equipment needed to assemble the car, such as tools, fixtures, and equipment. In the BoP (Bill order Planning), you will find all the sequences and Taskforce 1 operations. The BoP contains all the PII's. Then, in the BoP, you use SBS-toolbox to find the TF1 & TF2 (Taskforce 1 & 2). TF1 marked in green is also called (Value added), add-ons to the car, for example assembling a part to the fixture and TF2 is marked as red and is called (Non-Value Added), for example walking.

To find the right CC-object, you need to go into the DMU-garage. In there, you can find the complete car, variants are defined. These are often the car variants that are used for approximately 3 variants (RHD auto, LHD Manual, LHD electric).

The variants (of the car) can be configured in many ways. An example of a car variant can be that it can be divided into LHD and RHD, where the LHD can contain a deluxe stereo and the RHD is just a stereo.

To release a car, it takes around 1.5 years and can be divided into 5 steps: Concept (TF1), System (TF1), Detail (TF1), Industrialization (TF1/TF2), and Production. Currently, the Virtual operator training is at the end of Industrialization, before Production.
The PII's contain much information, for example if you go down further in the tree structure, you will find how the part will be assembled in relation to other parts, total amount of screws, etc. For example, the PII C8640-0307 contains information about that 10 assemblies to A will be made with 20 screws as the first sequence followed by 30 screws as the second sequence and 40 screws as the third sequence. Then, the PII C9110-0111 can tell you that 10 assemblies from B to A need to be made and need 20 screws.

C.6 Interview with plant launch manager and launch specialist

What is your role?
Plant launch manager and launch specialist, working with teaching operators and assembling final assemblies.

What are the phases of the operator training process?

- For newcomers only, 5 days of introduction are organized. It consists of explanations about Volvo Cars, the cars and the types of tools that they will be using as operators in a VCC plant. The goal is also for them to understand how to use the tools if they are not familiar with that.
- Then there are 2 days of off-the-line pre-training with a focus on reading, to understand how and why the work is done, for the specific part of the assembly the new operator will be working on. [In practice, this might be skipped] The material that is used consists of OIS and WES.
  - OIS (Operator Instruction Sheet): sequences text descriptions.
  - WES (Work Element Sheet): how to assemble the parts and why it is done how it is.
- The final step consists of practical on-the-line training, with a supervisor. This takes usually around 2 weeks before the new operator is up and running but it might take only 1 week in the case of someone that is experienced.

How the on-the-line training is conducted:

- Operators work in teams of 9 people, which includes one team leader.

- One team works on 8 different stations
- Usually there is one newcomer per team. The team leader designates a supervisor for the new operator, who is responsible for the training.
- If there is only one newcomer, he / she trains on each of the 8 stations, if there are more newcomers, then they are “locked” to one station and train only for that one at first. This is being done to limit quality issues.
- Supervisors have special training in how to train new operators.

How long is the current training process? (Phase that would be replaced by VR training)
It varies a lot from person to person but normally, the training is around 2 weeks. The new operator will follow an experienced operator and learn what he / she is going to do for a specific station.

How many people are trained at the same time?
Usually, one operator is trained at the same time, and we have had situations where up to 2 or three operators trained at the same time for that station.

Approximately, how many sequences is it per car project?
It depends a lot on the variant of the car because for some cars, it requires fewer sequences compared to other car projects but in general, there is around 1-2 sequences per station. Also, for some variants, there can be up to different sequences simultaneously.

How many operators are working at the same station at the plant?
At the moment, there is one operator per station. A couple of years ago, there used to be 2 people per station, but it was removed to avoid any quality issues. It can occur that it requires either 2 people to lift a part to assemble or 1 operator needs to use a tool to lift the part to assemble.

How long is it to create the training? Who does it? compare with time to create VR training.
I’m not sure because based on our experience, there is not much focus on reading the materials. Instead, the experienced operator teaches the new operator the methodology on how to assemble the parts from his / her experience that is not written in the OIS. Nowadays, we teach new operators verbally how we experienced operators do it.

Tips and tricks: are there specific pedagogical elements that are used?
Maybe not the best learning conditions are provided at the moment. It depends on the supervisor responsible for the training. Also, it is hard to achieve the time constraints when you are online for the first days. Therefore, the training can really be improved.

To help to reduce error occurrences, when a newcomer does something wrong, it is good to explain why it shouldn’t be done that way and what will happen because of that error. Explaining why and the consequences helps new operators remember the correct way to do things more easily.

Reflection questions / future:

Do you think that operator training would benefit from implementing VR? How / why?

- VR training is a good idea, but it comes with pros and cons:

- Time spent to train the operator in VR is unsure
- Some parts cannot be replicated in VR
- Physics is not recreated in VR now, so it is not 100% realistic, that’s a problem.
- It is good to learn a sequence, as it is better for visualization than pictures and texts
- It is good to standardize the workflow (so everyone performs the operations the same way)
- It is good to learn the ergonomics, as it is not the same for everyone (it is not possible to get that feeling in theoretical learning on paper)
- Whenever training operators for a new variant or model or big changes are made to the line workflow, training cannot be done directly, as the training material must be ordered and received, which can take a long time. VR training could help reduce that lead time.

Are there any barriers now why VR training has not been implemented? If yes, why?

- Cost: it is expensive to implement such a new method of training, because it requires new equipment and facilities. (could save cost due to quality if people are trained in VR)
- Time (VR would not reduce the 2 weeks on the line training because muscle memory is what is difficult to remember, it would just be adding 1 week of training, so managers need to be convinced that it is valuable in the long run.
    - It might be more beneficial for experienced operators (to learn new sequences because it is the only thing they do not know as they already know how to use the tools for instance).
    - For inexperienced people, it might add more time to the training.
- For the training time, it could be reduced, since operators who trained in VR would know the sequence before doing the on-the-line training, so that would be something less to learn when they are on the line for the first times. It could also help to reduce errors and increase their efficiency when they do the operations at the beginning.

How would VR training be implemented within the training process?

It would be very nice to use the VR for instructions and working methods for both the experienced and the new operator to get a feeling for the assembly sequence. Once they are confident enough on how to assemble it, the next step will be to do it physically, at the plant.
A new training process could look something like:

1. Short on paper instructions and working methods learning
2. VR training to learn the sequence.
3. Physical training, on-the-line

What are the most ideal features that will benefit the operators during the training from a pedagogic perspective?

- Detect mistakes: good positions of elements.
- Color coding is important.
- Realistic working station environment.
- Realistic positions of racks, tools...
- Check ergonomics: watch a recording of yourself once you did the assembly to see how you were doing it, to see what was wrong or good about your position.
- Click sounds usually on the plant: good to have it, same for tightening.

- Haptics to know when you are done (with screwing)
- Positioning in VR --> match with the plant.
- Display the explanations of why something is wrong when the trainee makes mistakes, explain the consequences if not done correctly. Display that and have them to read.
    - Fixed text if there is a lot of it.
    - Small instructions can be displayed in front of the eyes.

DEPARTMENT OF INDUSTRIAL AND
MATERIAL SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY