Ostfalia
Hochschule für angewandte
Wissenschaften

Wolfenbüttel

Faculty Computer Science

Christopher Peters, 70463991

Evaluation of interactive collaboration for 3D spacecraft development in Augmented Reality

Thesis for obtaining the academic degree

Bachelor of Science in Computer Science

at Ostfalia University of Applied Sciences

Supervisor: Prof. Dr.-Ing. Reinhard Gerndt
External supervisor: M.Sc. Anna Bahnmüller

Salzgitter
Suderburg
Wolfsburg

I hereby certify that I have written this thesis independently and have not used any sources or aids other than those indicated. I assure that I have marked all statements taken verbatim or in spirit from other works as such, and that the submitted work has not been the subject of any other examination procedure, neither in its entirety nor in essential parts.

Wolfenbüttel November 8, 2021

Abstract

Planning and designing spacecrafts is a complex process. It requires experts of different domains to collaborate together. At the end of the design process, a configuration engineer has to create a Computer-Aided Design (CAD) sketch of the spacecraft. This has to be revised often due to conflicts between requirements of spacecraft components. Thus, communication is crucial for understanding each discipline's requirements. At German Aerospace Center (DLR), a room especially designed for collaborative spacecraft engineering exists called Concurrent Engineering Facility (CEF). In order to support the communication, 3D visualization is already used in the CEF, but also in general in the industry. Recent Augmented Reality (AR) headsets like the Microsoft HoloLens 2 offer natural interaction with 3D holograms placed in the user's real environment. The aim is to support interdisciplinary communication by connecting multiple Microsoft HoloLens 2 devices over network to enable real-time collaboration on an interactive 3D visualization of a spacecraft in a shared environment. For that, a prototype is implemented that fetches visualization data from the DLR's data model for spacecraft development Virtual Satellite 4. It allows manipulating individual spacecraft parts with hand-gestures. In addition, multiple devices can be connected in a network session via the network library NetMQ 4. To evaluate the prototype's usability, potential usefulness and ability to resolve interdisciplinary conflicts between requirements, a user study is conducted in the CEF. The results show that the prototype is user-friendly in general and that its tools are useful. However, its relevance for solving interdisciplinary conflicts during the spacecraft configuration process is controversial. Furthermore, the communication between users wearing a HoloLens 2 which runs the application is good. Also, its potential usefulness in the CEF in general as well as acceptance of such technology is given.

Kurzfassung

Das Designen und Planen von Raumfahrtmissionen ist ein komplexer Prozess, welcher die Kommunikation zwischen Experten verschiedener Domänen bedarf. Am Ende des Designprozesses erstellt ein Konfigurationsingenieur einen Computer-Aided Design (CAD) Entwurf, welcher allerdings des Öfteren aufgrund von Konflikten zwischen Anforderungen der verschiedenen Satellitenbauteilen überarbeitet werden muss. Daher ist Kommunikation notwendig, um die Anforderungen der verschiedenen Komponenten zu verstehen. Beim Deutschen Zentrum für Luft- und Raumfahrt (DLR) existiert ein speziell für die kollaborative Raumfahrzeugentwicklung konzipierter Raum, welcher Concurrent Engineering Facility (CEF) heißt. Um die Kommunikation zu unterstützen, wird in der CEF und generell in der Industrie 3D Visualisierung benutzt. Augmented Reality (AR) Headsets wie die Microsoft HoloLens 2 bieten natürliche Interaktionen mit 3D Hologrammen, welche in der realen Umgebung des Nutzers platziert werden. Das Ziel ist es, die interdisziplinäre Kommunikation zu unterstützen, indem eine Software für die Microsoft HoloLens 2 entwickelt wird, welche die Verbindung von mehreren HoloLens 2 Geräten über Netzwerk und somit Echtzeit-Kollaboration in einem Raum an einer interaktiven 3D Visualisierung ermöglicht. Dafür wird ein Prototyp implementiert, dessen Visualisierungsdaten vom Server der Software Virtual Satellite 4 stammen, welche beim DLR als gemeinsames Datenmodell in der Raumfahrzeugentwicklung verwendet wird. Neben der Manipulation von einzelnen Raumfahrzeugkomponenten per Handgesten ermöglicht der Prototyp die Verbindung mehrerer HoloLens 2 Geräte über Netzwerk mithilfe der Netzwerkbibliothek NetMQ 4. Um die Nutzerfreundlichkeit, den potenziellen Nutzen und die Fähigkeit dieser Software interdisziplinäre Konflikte in der CEF zu untersuchen, wird eine Nutzerstudie durchgeführt. Die Ergebnisse zeigen, dass die entwickelte Anwendung nutzerfreundlich und dessen zusätzliche Werkzeuge nutzerfreundlich sind. Trotzdem ist die Relevanz für das Verhindern von interdisziplinären Konflikten während des Konfigurationsprozesses nur mittelmäßig. Des Weiteren ist die Kommunikation zwischen Nutzern, welche eine Microsoft HoloLens 2 tragen, auf der die Anwendung ausgeführt wird, gut. Auch ist der potenzielle Nutzen in der CEF generell gegeben, genau wie die Akzeptanz einer solchen Technologie.

Contents

List of Figures

Index of abbreviations

DLR German Aerospace Center

NASA National Aeronautics and Space Administration

ESA European Space Agency

VirSat Virtual Satellite

CEF Concurrent Engineering Facility

AR Augmented Reality

MR Mixed Reality

VR Virtual Reality

HMD Head Mounted Display

OST Optical See Through

VST Video See Through

UWP Universal Windows Platform

PHMD Peripheral Head Mounted Displays

FOV Field of View

API Application Programming Interface

REST Representational State Transfer

CA Category Assignment

SEI Structural Element Instance

UUID Universally Unique Identifier

SUS System Usability Scale

1. Introduction

Concurrent engineering of spacecrafts is a complex endeavour. To support it, a specially designed room is used in the early design phase called Concurrent Engineering Facility (CEF) at the German Aerospace Center (DLR) in Bremen [1]. In this room, experts from various disciplines of space engineering collaborate together. Similar rooms can be found at National Aeronautics and Space Administration (NASA) and European Space Agency (ESA) [1].
At the end of the design phase, a configuration engineer creates a Computer-Aided Design (CAD) sketch of the whole spacecraft. To fit the requirements of each component, every expert's knowledge is needed. Thus, communication is crucial during this collaborative process. The CAD sketch must be revised often, because requirements and constraints are communicated differently in varying domains. Thus, conflicts between components appear [1].

In the industry, often visualizations are used to support collaboration in such situations. It was also observed that the CEF's engineers help themselves out with quick visualizations like drawings or hand-gestures [1].
We propose a system to resolve conflicts during spacecraft configuration with the help of interactive visualization. While wearing the Augmented Reality (AR) headset *Microsoft HoloLens 2*, engineers can create a raw draft of the spacecraft collaboratively face-to-face in 3D.

In AR, the user can interact with virtual objects displayed in his real environment. Thus, it can facilitate communication through allowing non-verbal cues while also enabling interaction with a 3D visualization. This shall avoid misinterpretation of requirements and constraints of spacecraft components between different disciplines.

The Microsoft HoloLens 2 has optical see-through displays and tracks hands as well as fingers. Thus, interaction solely with hand-gestures is possible. Furthermore, it is stand-alone and therefore needs no other hardware to work. Through its spatial awareness system, the virtual objects can be positioned and rotated in space. Therefore, these objects can be viewed from different angles without manipulating them.

In this work, a prototype application for the HoloLens 2 is developed that is evaluated later through a user study. The goal is to find out about the prototype's ability to support communication as well as its ability to resolve interdisciplinary conflicts in the configuration process. In addition, the prototype's acceptance in the CEF and its usability is researched.

First, chapter 2 describes the state of the art in the CEF, as well as Virtual Satellite 4. In addition, AR and related work to AR collaboration are depicted.
The following chapter 3 gives details about the preliminary work done before the thesis and implementation details with the focus on network communication.
Chapter 4 focuses on the execution and statistically evaluation of the user study.
The final chapter 5 summarizes the work done and derives recommendations for the future.

2. Theory

This chapter gives background information for the researched topic. First, the *Concurrent Engineering Facility* as well as the associated design process of space missions are described in section 2.1. Second, *Virtual Satellite*'s data model and its server are explained due to its relevance for the implementation in section 2.2. Last, section 2.3 Augmented Reality systems and especially the HoloLens 2 are depicted. Also, it presents the state of the art in Augmented Reality collaboration.

2.1. Concurrent Engineering Facility

Concurrent Engineering is a concept that has found its way into astronautics back in the 90s. It is meant to improve the engineering process by making the experts of different domains work together in the early phases of development [1]. Concurrent Engineering in the design phase is also referred to as *Concurrent Design*.

The Concurrent Engineering Facility (CEF) is a room specially designed for Concurrent Engineering. Examples of CEFs can be found at the National Aeronautics and Space Administration (NASA), European Space Agency (ESA) and German Aerospace Center (DLR) [1]. The ESA's equivalent is called *Concurrent Design Facility*. Figure 2.1 shows the DLR's CEF in Bremen.

The room's layout is closely inspired by ESA's *Concurrent Design Facility* [1]. It was designed with the goal of creating an atmosphere of scientific collaboration. The CEF contains up to 12 seats for engineers with additional seats for guests and customers. The seats are arranged in a semi-circle facing a wall which holds three big monitors. In addition, the CEF contains workstations at every seat.

Figure 2.1.: CEF of DLR in Bremen [2].

The studies carried out in the CEF are taking approximately one to two weeks, split into several sessions that last two to three hours. In a session, experts of different domains are working together in small groups. After that, they review or discuss the work done by other groups. These sessions are repeated until a solution for every subsystem is found. The data resulting of the CEF's studies is stored in a software called *Virtual Satellite 4*.

The configuration engineer has to create a CAD sketch of the spacecraft as a part of the process. Due to the configuration engineer not having the knowledge of other disciplines, the sketch has to be revised often due to conflicts between them [1]. This is time-consuming and therefore expensive. To prevent recreating this draft often, the idea is to let the experts create a rough visualization draft collaboratively. Then he would see how the spacecraft parts had to be placed in relation to each other. This approach shall support spotting conflicts early.

In previous work, the concurrent engineering process of a big U.S. automotive manufacturer in 1996, the engineers projected the current CAD model revision on a screen for discussing conflicts [3].
Quan et al. [4] proposed a web-based collaborative environment to enhance prod-

uct design. It contains a visualization module which allows viewing, remarking and evaluating design parts for all users.

As has already been observed in CEF studies, the CEF engineers support their communication with gestures and quick drawings. Thus, visualization already is used for providing information [1]. Regardless of being quick and easy, these do not have any relation to the data model.

We propose an Augmented Reality (AR) system for resolving conflicts during the configuration. It provides real-time collaboration in a shared space by giving every user the ability to manipulate the same visualization with his hands. Furthermore, its visualization data comes from the server of *Virtual Satellite*.
A study from 2018 observing the spacecraft design process of 15 different CEF's showed that real time collaboration and Augmented Reality are among the top three trends influencing new concurrent design tools [5].

Further previous work is depicted in section 2.3.

2.2. Virtual Satellite

Virtual Satellite (VirSat) (currently version 4, 30.09.2021) is an open-source software developed by DLR. The application is located in the domain *Model Based Systems Engineering*. It means that VirSat supports the whole development life cycle of systems-engineering in astronautics. It is the approach of the DLR for storing parameters of the spacecraft that are discussed and set in the studies done in the CEF. In the past, this was done with Microsoft Excel, but issues arose with this approach, such as usability problems during the sessions [1]. To validate the visualization of a spacecraft model in AR, the *3D-View* of VirSat can be used. The view is implemented with the *Visualization Toolkit* and creates a three-dimensional visualization from the parameters set in the data model. The view which displays a self-built test satellite can be seen in figure 2.2.

The VirSat server exposes the data models of spacecraft missions. Since the application extended in this thesis communicates with it, both will be described in the following subsections.

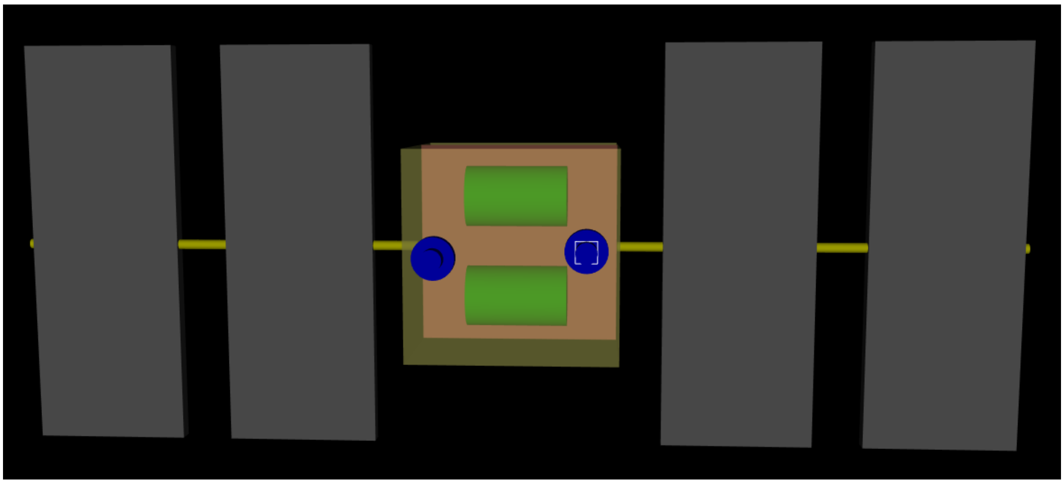


Figure 2.2.: Virtual Satellite's 3D view showing a self-created test satellite.

2.2.1. Data model

The model consists of four different tree structures:

Tree name	*Short description*
Product Tree	Definitions of equipment that should be used. *Associated to a model of a spacecraft*
Configuration Tree	Information which is related to instances of equipment. *Associated with an instance of a model*
Assembly Tree	Information which is related to equipment that will be build. *Associated with an actual built spacecraft*
Product Storage	Information which is related to supplied equipment. Can be combined. with the Assembly tree. *Associated with equipment that is present for use*

Table 2.1.: Variants of trees in Virtual Satellite [6].

Important for this project is the *Configuration Tree* because the position and rotation will be set here. Size may be inherited from components of the *Product Tree* [6].

The tree's node is a Structural Element Instance (SEI). Every node knows its children as well as its parent. One SEI can contain several Category Assignment (CA)s.

A CA contains special information of a category. The necessary category for this project is *Visualization*.

Furthermore, a CA consists of a Universally Unique Identifier (UUID), type, name and different *Properties*. The CA for visualization contains information about size, rotation, position, color, transparency, geometrical shape and the server file path to a custom STL model.
Properties have, except for the UUID, no fixed attributes. For example, the property that provides information about the position on the abscissa knows its UUID, value and the value's data type (as a string). On the other side, the property that holds the file path holds information about its UUID and the value, but no data type field.

For referencing SEIs and CAs, lite-versions are used. These contain only their UUIDs and names.

2.2.2. Server

The VirSat server is a stand-alone application that can hold different repositories with VirSat projects and allows their manipulation through a Representational State Transfer (REST) Application Programming Interface (API). A client has to authenticate itself with a username and a password through *Hypertext Transfer Protocol basic access authentication*. Additional to that, the user needs to have a fitting role that allows the manipulation of the repository he wants to access.

REST is a popular set of design constraints for distributed hypermedia systems submitted as a PhD dissertation by Roy Fielding in 2000, of which two important constraints are described below: *Client-Server* and *Statelessness* [7].

Client-Server is an architecture that enforces the principle *Seperation of Concerns* by specifying two roles: client and server [8]. The server data, which can be accessed by an implemented user interface of the client. REST additionally uses the *Request-Response* pattern. This means that the server only distributes data on request by the client. For streamlining the communication, the server often implements an API.

Statelessness means that a request contains all the information needed, so that the server doesn't have to save any context for a client in order to understand the request properly [8].

Furthermore, RESTful web services are all about resources. A resource is a unique reference to an entity. Using the various HTTP verbs, information about the states of resources can be queried or a resource's state can be changed [8]. The basic HTTP verbs are:

Verb	*Description*
GET	Read the resource representations
PUT	Create a new resource
DELETE	Delete the resource
POST	Modify the resource

Table 2.2.: Basic HTTP verbs and their meaning [9].

The representation of a resource's state can be in any kind of data format. Most common ones are *JSON* or *XML*.

The REST API of the server provides several endpoints for manipulating spacecraft models. Needed endpoints together with their verbs and tasks are shown in table 2.3.

Endpoint description	*Verb*	*Task*
Fetch a list of root SEIs	GET	Returns an array of all root SEIs in JSON
Fetch SEI	GET	Returns a full SEI in JSON for a UUID
Fetch CA	GET	Returns full CA in JSON for a UUID
Update CA	PUT	Updates an existing CA

Table 2.3.: Server's REST API endpoints needed for this project.

2.3. Augmented Reality

AR can be imagined as the middle ground between reality and virtuality. While in Virtual Reality (VR) the user is completely immersed in a virtual world, AR allows the user to see the real word supplemented with virtual objects, also called ”Holograms”.

Generally accepted is the MR Taxonomy from Paul Milgram [10], which says that Mixed Reality (MR) is a spectrum bounded by virtuality (VR) on one side and reality on the other, as can be seen in figure 2.3. Reality means any environment consisting entirely of real objects, whether it is viewed in person or only on a screen. Virtuality/Virtual Reality is any environment consisting entirely out of virtual objects e.g. conventional computer graphic simulations. In this spectrum, AR is located more in the direction of reality than virtuality.

Sometimes MR is also used as a synonym for AR in the literature. For example, Microsoft markets its AR headset *HoloLens 2* as a ”Mixed Reality Headset” [11].

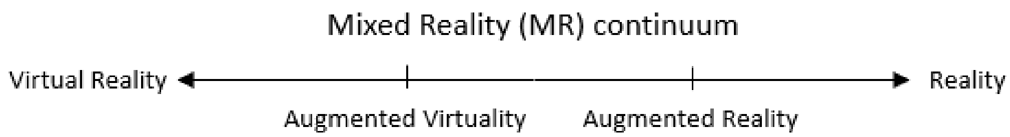


Figure 2.3.: The Mixed Reality continuum according to Paul Milgram [10].

First, various AR systems are depicted. Second, the Microsoft HoloLens 2 is described in detail.

2.3.1. Systems

AR has no consistent definition, yet there are three popular criteria developed by Azuma [12] in 1997 for identifying an AR system:

1. Combining reality and virtuality
2. Interactivity in real time

3. Virtual content is registered in a 3D space

There exist several very different systems that match these criteria, such as smartphones or tablets with appropriate software. Also, there are Head Mounted Display (HMD)s, which consist of displays mounted to a head collar and positioned in front of the user's eyes [11]. To achieve a mix out of virtuality and reality, two main approaches exist: Optical See Through (OST) displays and Video See Through (VST) displays.
In an OST-HMD, the user can see the real environment through glass, while holograms are visualized on the see-through displays. This approach has the advantage of the user's ability to see the reality through his own eyes. Thus, only the virtual objects are dependent of a particular frame rate or resolution. On the other hand, a problem is the low contrast of these displays in bright environments, which makes the holographic content hard to see in bright environments.
VST-HMDs let the user see holographic content combined with captured feed of at least one camera, that is mounted on the HMD. Main drawbacks are the restricted dynamic range and resolution.

AR HMDs contain a wide range of devices. So called *Smartglasses* display visual information, not covering the whole user's Field of View (FOV). Their purpose is providing additional visual information while not limiting the user while performing real world tasks.
Matthies et al. [13] place this technology in a new subcategory of HMDs, called Peripheral Head Mounted Displays (PHMD)s. Available examples are Google's *Glass Enterprise Edition 2* or Vuzix's *Blade*. Common interactions are touch, voice input or predefined gestures. Most available models need a connection to another device (e.g. smartphone) for their full functionality.
These are unsuitable for rendering three-dimensional virtual objects and enabling interaction with them, due to the lack of performance and interaction possibilities.

Most common HMDs, which are not PHMDs, are *Magic Leap 1* or the *Microsoft HoloLens 2*. They are bulkier but have stronger hardware as well as more advanced sensors. Furthermore, these HMDs aim for immersion [13]. In this project, the

Microsoft HoloLens 2 is used. Since it is a stand-alone system and can track hands as well as fingers, the HoloLens 2 fits the use case. Figure 2.4 shows the Google Glass Enterprise Edition 2 as well as Microsoft HoloLens 2.

Figure 2.4.: Left: *Google Glass Enteprise Edition 2* [14], right: *Microsoft HoloLens 2* [15].

2.3.2. Microsoft HoloLens 2

The Microsoft HoloLens 2 is an AR HMD firstly revealed at *GSMA Mobile World Congress* 2019. It is the successor of the Microsoft HoloLens 1 from 2016. Furthermore, it uses OST displays. It allows several different types of user interactions like eye tracking, hand tracking and voice control [16]. First, the device's specifications are described. Second, the interaction possibilities are depicted. Last, the network communication between multiple HoloLens 2 devices is presented.

Technical Environment

The HoloLens 2 contains a *Qualcomm Snapdragon 850* processor and 4GB DRAM as well as 64GB flash storage.

Furthermore, its OST displays use the *waveguide* technology where light rays are fed into a glass body in such a way that they reflect at the correct edge and hit the user's eye as an image. A separate waveguide layer is required for each color channel (RGB) [11].

Moreover, the HoloLens 2 ships with several sensors that allow different types of interactions with it. Besides an acceleration sensor, a gyroscope, a magnetometer, a microphone array and an integrated camera, the HoloLens 2 has sensors for head-tracking, eye-tracking and a depth sensor [16].

The device provides a 52° diagonal FOV. This is an improvement compared to the HoloLens 1 with a 34° FOV. In comparison, the common VR headset *HTC Vive Pro* comes with a 110° FOV.

With the help of the sensors, the user has different abilities to provide input to the HoloLens 2.

Interaction

The Microsoft HoloLens 2 provides different possibilities of interaction to the user. The built-in microphone array and Microsoft's own voice assistance service *Cortana* make it possible to recognize voice commands. In addition, the device's orientation and the user's position in the room can be determined. Furthermore, it is possible to determine the direction of the eyes as well as the hands and individual fingers. By tracking the fingers, gestures can be recognized by the system. The HoloLens 2 comes with some standard gestures.
The *Air tap* is performed by moving the thumb and index finger together and apart (see figure 2.5). It is used to click interactive holograms such as buttons. The *Air hold / Grab* is the same motion but with a small difference. Instead of moving the fingers apart right after the tap, they are hold together. While the fingers are pinched together, a target hologram is grabbed. The target will be released, if the fingers move apart after the pinch.

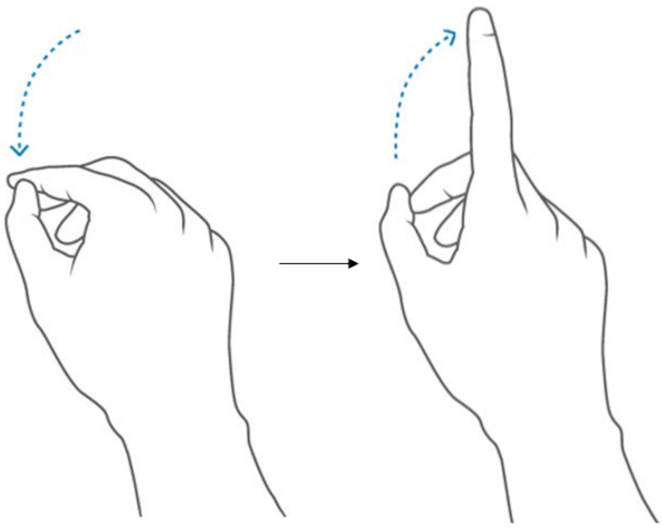


Figure 2.5.: Motion of the *Air tap* gesture.

Furthermore, the Microsoft HoloLens 2 supports, in contrast to the first generation, gesture manipulation with *6 degrees of freedom*. That means the system supports simultaneous tracking of rotation and position of the user's hands, thus allowing more natural interaction with it. For example, a user could grab a hologram and rotate and reposition it concurrently with one only one hand. Doing gestures with both hands simultaneously is supported as well.

Network communication

To Communicate between multiple HoloLens 2 via network, many ways exist.
Since the *Unity* game engine is used in the scope of this project, a library working with Unity must be used. It supports C# scripting for the Universal Windows Platform (UWP) API and can build into C++ code. The HoloLens 2 has a Wi-Fi module and implements the UWP API in C++. Furthermore, the UWP API supports creating TCP and UDP connections. Consequently, all networking libraries can be used, that have UWP support.
Many high-level networking libraries are maintained by Unity or user communities. Unity has the *Multiplayer High Level API* (MLAPI) which is well tested but deprecated. Therefore, Unity's newer *MLAPI* was observed.

The MLAPI provides an easy setup of network communication between software made in Unity. However, it has a strong focus on games, which could lead to workarounds and unneeded complexity. Additional to that, it is currently (in preview state (0.1.0) and therefore not production ready.
The use of cloud services like *Photon Unity Networking* disqualified due to the violation of data protection guidelines of the DLR.

The chosen more low level approach is *NetMQ 4* which is a `C#` implementation of *ZeroMQ*. It is well tested and used by big companies like Microsoft or Samsung. Furthermore, it is available for many programming languages. Therefore, a high level of platform independence is given. *NetMQ* uses TCP as the underlying network protocol which is preferable for this project due to its ability to detect and repair data loss in connection. Moreover, *NetMQ* puts an abstraction layer above the TCP sockets and allows different patterns of network infrastructure. The last layer provides communication between *NetMQ* and the rest of the application. The whole stack can be seen in figure 2.6.

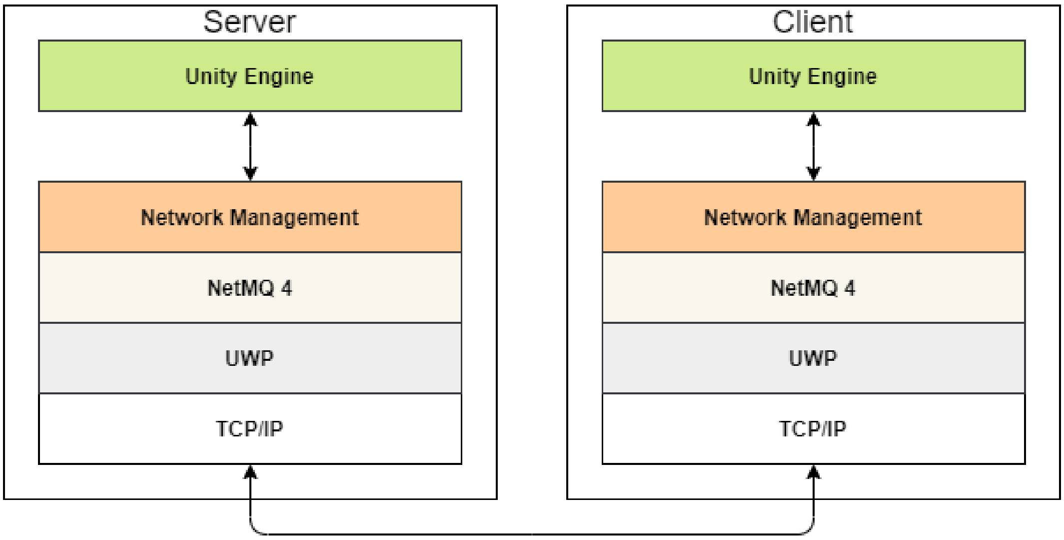


Figure 2.6.: Diagram of the used technology stack for network communication.

To provide a fast communication, the data format and its serialization process is important. This is especially true if data is sent and received frequently. Therefore, Google's *Protocol Buffers* are used for (de)serializing structured information in the application. For using it, the developer creates a *.proto* file where the data structure is defined and classes for the programming language can be generated of. The *.proto* files then are used for generating or parsing the byte stream that flows between two communicating parties. The serialized data is not human readable but its size and (de)serialization speed can dominate e.g. one of the most common data formats *JSON* [17].

To realize the communication, a client-server architecture is used, where one HoloLens 2 acts as a server, while the other devices have to connect to it as clients. The host (server) fetches the visualization data from the VirSat server at start. Then, it distributes the current state of the visualization to whatever joins the session. Messages are always sent between a client and the server. If a client wants to target a message to only one client or all participants, the message has to be sent to the host. Afterwards, the host forwards the message to the other clients.
The communication concept is shown in figure 2.7.

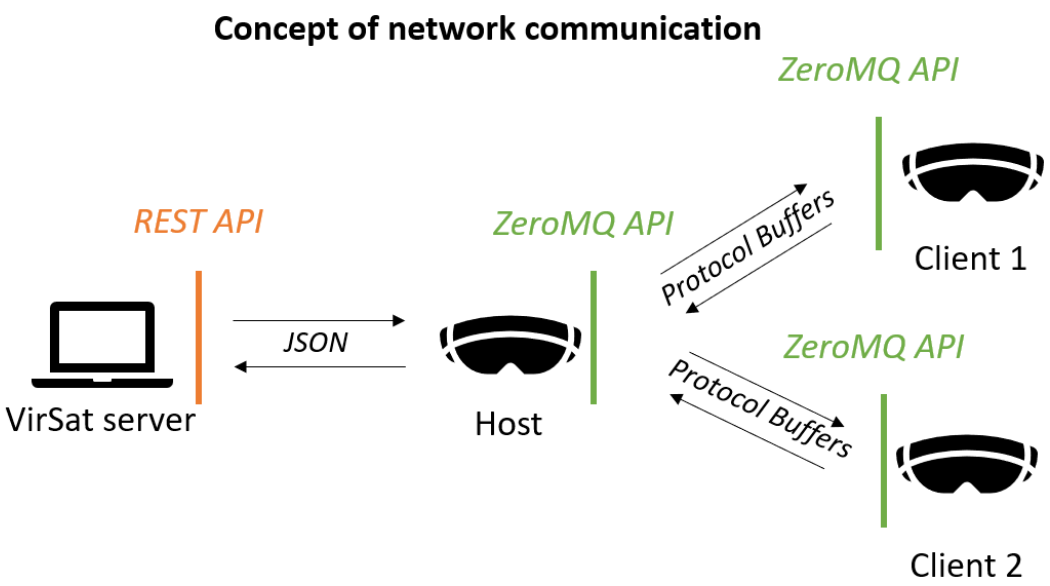


Figure 2.7.: A session's host communicates with the VirSat server in JSON by obtaining visualization data or sending back changes. The clients communicate with the host in *Google Protocol Buffers* and form a session through their *ZeroMQ* APIs.

2.3.3. Collaboration in AR

Collaboration means having a working relationship with two or more participants [18]. They share a vision and objectives. For building trust, communication is important.
It consists of verbal and non-verbal aspects [19]. For proper communication, a human needs non-verbal cues, including:

- Eye contact
- Facial expressions
- Tone of voice
- Posture and gesture
- Touch
- Intensity
- Sounds

In AR face-to-face collaboration, a user can see the other user's gestures and facial expressions of other participants (with constraints applied by the used hardware). This visibility of the real world including other people affects the communication positively [20].

One of the first projects that implemented multi-user Augmented Reality is the *Studierstube* [21] project from 1996. It contained two *Optical See Through-Head Mounted Displays*, which visualized data from Indy workstations that were connected to each other via TCP/IP. Thus, users could see a shared visualization in the real world together. Although the project found high acceptance within unskilled users, the collaboration was not evaluated.

Wang and Dunston [22] researched the effect of AR collaboration on solving a task in 2009, remote as well as face-to-face. For this, two AR *Head Mounted Displays* were used which were connected via TCP/IP. Their results show that the task solving time could be drastically reduced in both ways. The task's mental demand could be reduced as well.

Furthermore, experiments were made in 2015 with visualizing data directly coming from VirSat [23] in VR. The visualization data was associated with the data model. The experiment's conclusion was that VR can improve inter-domain communication. Since these are results with VR technology, they are only transferable to a limited extent. For example, VR creates more immersion by placing the user in a solely virtual environment. Thus, VR can create immersion more easily than VR [11]. Despite the lack of immersion in AR, it could support collaboration by improving communication through letting the user see other people.

Evidence of improvement through AR exists not only for communication, but also for a user's interaction with virtual objects. Baranowski [24] compared placing virtual objects with a HoloLens 1 with placing virtual object in VirSat. Results show, that repositioning can be better with the HoloLens 1. However, the interaction metaphors must be chosen correctly.

Due to the mentioned works, it is assumed that AR can improve the spacecraft configuration process. Through the natural communication created by AR, errors should be solved more quickly and therefore speed up the process. Furthermore, the acceptance of such a system in the CEF is researched.

3. Implementation

This chapter informs about the implementation of collaborative AR. First, section 3.1 describes Unity as the used engine and the preliminary work's functionality. The software allows fetching and manipulating visualization data from the VirSat server on the HoloLens 2.
Second, section 3.2 describes the extension's requirements and their implementation. The extension provides connection between multiple HoloLens 2 devices to observe and manipulate visualized data from VirSat in collaborative manner.

3.1. Development environment

3.1.1. Unity

Unity is primarily a gaming engine with a clean interface. The language of the engine's scripting API is C# [25]. It is used in different domains like automotive or construction. For example, it is used by Toyota to develop VR and AR tools [26].

In Unity, the developer is able to create different scenes that can be loaded simultaneously. In every scene there is a tree structure of GameObjects which act as a scene graph. A GameObject's position, rotation and size are relative to their parents position, rotation and size, unlike in VirSat, where only the position and rotation of a spacecraft part is relative to their parent and not the size.

To describe the behaviour of a GameObject, a C# script can be created that derives from the built-in class MonoBehaviour. This base class allows the implementation of event functions and prohibits the well-known instantiation of derived scripts via

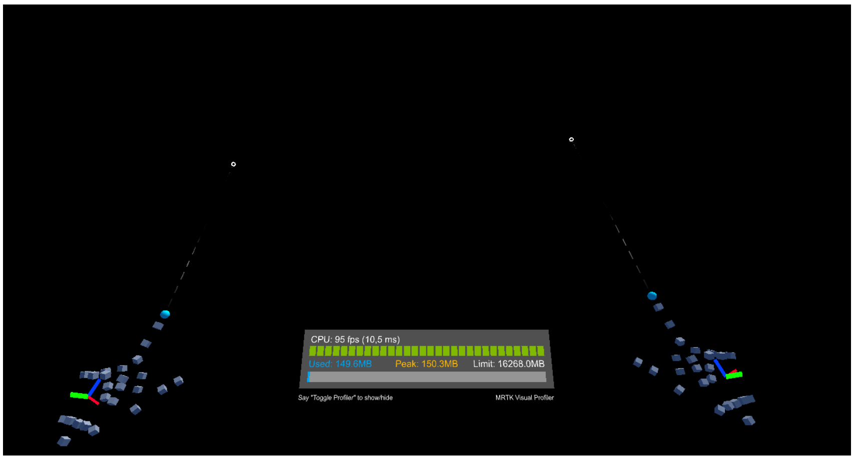


Figure 3.1.: Simulation of hands in the Unity editor.

the keyword **new**. The only way of using the script is attaching it to a GameObject which leads to the GameObject holding an instance of the script. The event functions provided by MonoBehaviour are called in the Unity main loop [25] .

An example of an event functions is `Update()` that will be called before every frame. Another one is `Start()` which will be called once before any `Update()` call [27].

The MRTK (Version 2) is an open source toolkit developed by Microsoft that is available for Unity as well as for Unreal Engine. The MRTK for Unity provides building blocks and components for easier building of Mixed Reality Applications for many platforms like Windows Mixed Reality, OpenXR or OpenVR [28].

One of the biggest features of the MRTK is its input system. It wraps up the communication with the *Microsoft Mixed Reality APIs* and allows the developer to detect the different interaction possibilities of the HoloLens 2 easily. Additionally, it comes with a visualization of the controller (hands for the HoloLens 2) for the *Play Mode* of the editor [29]. The *Play Mode* simulation is shown in figure 3.1.

3.1.2. VirSat REST Client

The implementation of AR collaboration extends a base software that was developed preliminary. It communicates with the VirSat server through its REST-API and is able to process the server's visualization data. In the following, the application's features are portrayed.

In the opening (see figure 3.2), the user can enter information of the target VirSat server he wants to communicate with. This information will be merged together to a legit URI later. Additionally, the user may choose a predefined configuration by clicking the "Choose config" button. The configurations are loaded from *.config* files located in the application's *RoamingState* folder. To fetch the model and start the process of manipulation, the *Start* button has to be pressed.

Figure 3.2.: *Connection Information* window in the opening of the *VirSat REST Client* on Microsoft HoloLens 2.

As shown in figure 3.3, the client fetches the visualization data from the VirSat server, which itself holds the model in a local git repository connected to a remote git repository.

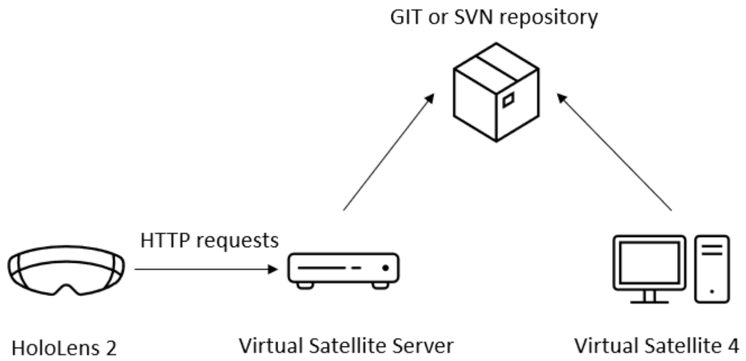


Figure 3.3.: Communication between the *VirSat REST Client* and the server.

For fetching the visualization data the client traverses the tree-structured data recursively. Therefore, it requests all root SEIs at first and continues to fetch every received SEI's children. For every SEI, the CA holding visualization data is processed. After fetching, the application maps the SEIs into a structure of Unity's GameObjects. One GameObject takes responsibility for the visualization and the ability of manipulation of one spacecraft part. Thus, the parts keep the parent-child relationships between the VirSat data model.

When the visualization is finished, the user will be able to select and manipulate any spacecraft part by focusing it. That can be done from near or far by a ray that is aligned with the user's forearm. To grab a virtual object, the user has to do the *Grab* gesture. That means moving the index finger and thumb together. While a part is focused, the part's name is displayed in the middle of the user's sight, as shown in figure 3.4. If the user starts the *Grab* gesture, the grabbed part's position can be manipulated by moving the used forearm / hand that is doing it. Furthermore, the part's rotation can be adjusted simultaneously according to the grabbing hand. Thus, a part can be manipulated with *6 degrees of freedom*.

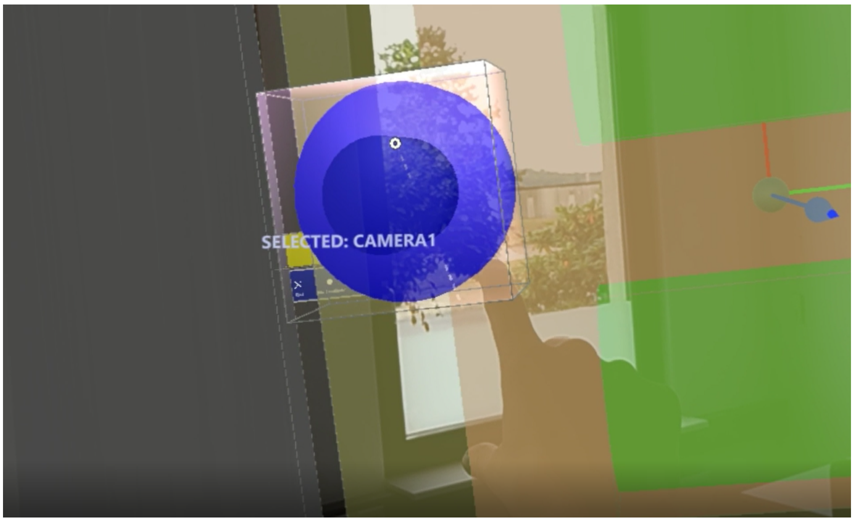


Figure 3.4.: User selects a spacecraft part in the *VirSat REST Client* using the grab gesture.

The process of manipulation is supported by a set of tools located in the *Toolbar* which appears in the user's sight (see figure 3.5).

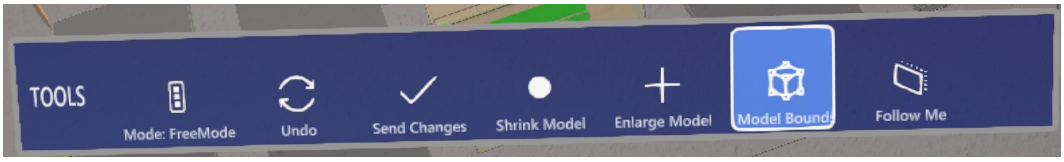


Figure 3.5.: Toolbar of the *VirSat REST Client*.

In the toolbar, the leftmost button "Mode: ..." switches between two different selection modes. On the one side, the default mode ("Free" mode) allows manipulating every spacecraft part the user can point at. On the other side, "Structure View" mode exists, to manipulate virtual objects that cannot be selected directly, for example if they are inside another virtual spacecraft part. The "StructureView" mode's activation reveals a window (see figure 3.6) following the user's sight. It shows the tree structure of the spacecraft, whereby a button is displayed for every part. These will shine in light blue and allow manipulation of the associated spacecraft part on activation. Then, every other part except the activated one is still visible but not available for manipulation. Thus, the user's arm rays will not collide with them.

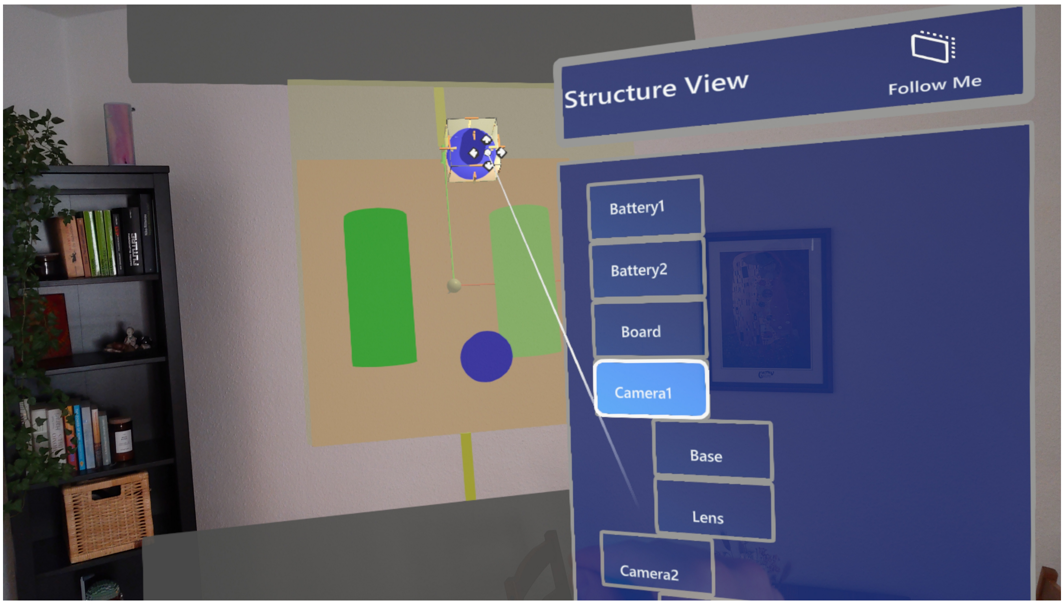


Figure 3.6.: The mode "Structure View" allows to select different virtual satellite parts directly from a list. The chosen object will be highlighted and can be manipulated. It will be useful especially if individual objects are covered by other virtual parts.

The next button "Undo" reverts the last change of a spacecraft part. If the user manipulates a part, its position and rotation in addition to a reference to itself will be saved on a stack at the beginning of the manipulation. If the button is pushed, the last saved action will be popped from a stack. Then, the spacecraft part referenced in this element is translated and rotated to the values stored in the popped element. If no actions were done, an error sound will be heard.

To save the changes on spacecraft parts to the data model in VirSat, all manipulated CAs for visualization can be sent back to the server via the button "Send Changes" located in the middle of the toolbar. On every started manipulation, a reference to the CA for visualization of the manipulated spacecraft part will be pushed on a stack. If a user pushes the button, for every CA on the stack a POST request will be sent to the server containing the CA in JSON.

The last functionality *Model bounds* can be (de)activated with the "Model bounds" button. If activated, the button will shine in light blue. Furthermore, all spacecraft parts will be wrapped up in one visual box. It can be seen in 3.7. After activation, it

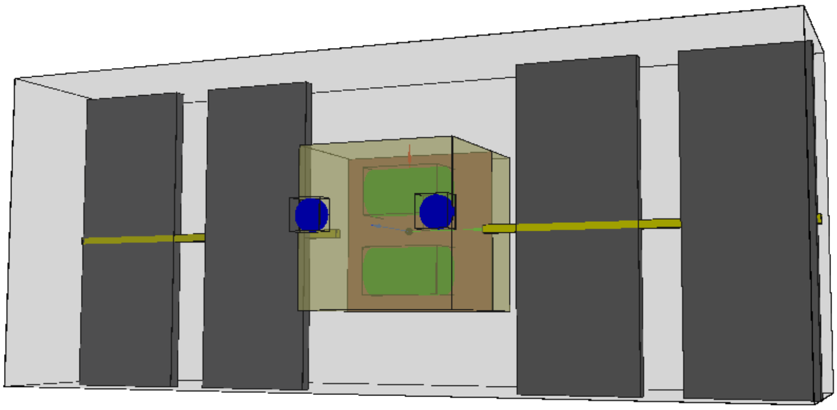


Figure 3.7.: The box around the spacecraft parts allows to manipulate the whole spacecraft's position, rotation and size without changing the spacecraft part's real measures.

can be manipulated like a spacecraft part using the *Grab* gesture.
With this functionality, the whole spacecraft can be moved, rotated and resized in the room without changing the real measures of the contained parts. Thus, the spacecraft's root coordinate system then follows the visualization. The last two buttons "Shrink Model" and "Enlarge Model" provide a way to resize the visualization, without having to use both hands.

3.2. Network communication

3.2.1. Connection & Session

Since the CEF contains at least 12 seats for every engineer, the application should allow at least 12 participants communicating with each other. To minimize frustration, it should be resilient against subtle disconnects. A lost connection should be detected and should not lead to an undiscovered asynchrony between the networking participants.
Moreover, every client should be able to see the complete spacecraft after joining the session.

For implementing a connection between multiple HoloLens 2, a mix of *NetMQ's Publish/Subscribe* and *Router/Dealer* patterns is used, as can be seen in figure 3.8.

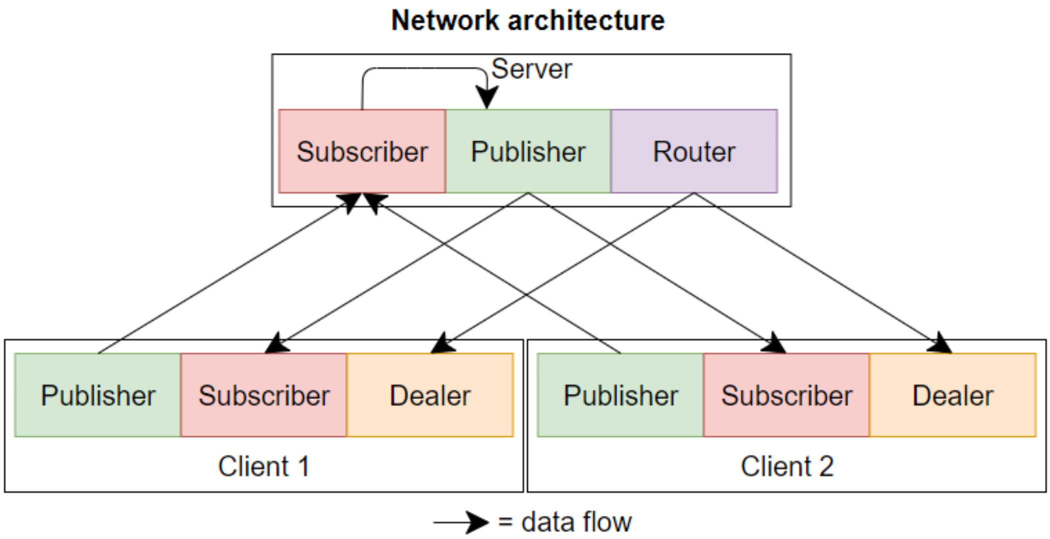


Figure 3.8.: Diagram of the application's network architecture.

Every client has a *Publisher*, *Subscriber* and *Dealer* socket. To fully join the host's session, the *Subscriber* signs up to the host's *Publisher* socket and vice versa. In addition, the client's *Dealer* connects to the host's router socket. A *Publisher* socket can only send messages to all its subscribers, while *Subscriber* only can receive messages from sockets it subscribed to. Every message the host's *Subscriber* socket receives is published by it again.

Router sockets store ID's of connected *Dealer* sockets and therefore can send messages to a specific receiver. Furthermore, *Dealer* sockets can send messages to their connected *Router* socket.

The *Publisher/Subscriber* infrastructure is used for distributing information for every participant in the session, whereas *Router/Dealer* is used for direct communication between a client and the server. If a HoloLens 2 starts a server (open NetMQ sockets), it hosts a session. Other clients then can join the session by connecting to the server (connect to the open NetMQ sockets).

NetMQ sockets do not have any default limit of maximal concurrent connected sockets. Therefore, more as 12 participants are possible. Although, more than 4 devices at once could not be tested due to the lack of hardware.

If a client joins a server's session, the server will send all visualization data to the joined client. First, all SEIs are send, then all CAs.

The process is preceded and followed by events. The first triggers a lock of the whole interaction with the HoloLens 2 on all participants, the second releases it. This avoids asynchrony by preventing applications to send updates for spacecraft parts while a connecting system is busy initially visualizing the model. If an error occurs during the sending process, an event for releasing the locked interaction will be sent regardless.

A session's host remembers a disconnected client for 1000 ms by the socket's ID. In this time, it will not recognize the action as a disconnect, if the client reconnects during this time span. If another user would end a manipulation during this time span, asynchrony could be possible. Nevertheless, this exchange is considered reasonable. The whole process of a client joining a session is shown in figure 3.9.

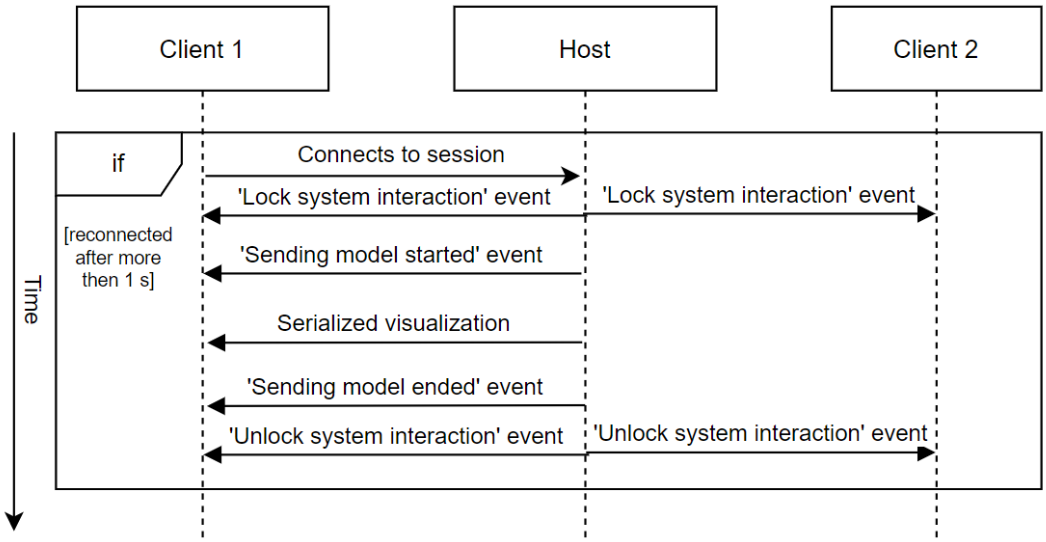


Figure 3.9.: Diagram of process "Join session": Client 2 and the host are in a session. Client 1 joins the session.

The process of detecting a connection loss revolves around so-called *heartbeats*. A client sends a heartbeat message every 300 ms to the host. If the message reaches the host, it keeps track of the arrival's timestamp. The host checks every connection's last timestamp every 100 ms. If this timestamp is older than 300 ms, all clients will be notified about a lost connection. Additional to that, every heartbeat message will be answered by the host.

If two of a client's heartbeat messages stay unanswered, it will close all sockets and will jump back to the opening menu.

3.2.2. Manipulation

To create an environment for collaboration, user must be able to do manipulations simultaneously. A solution for the case that more than one user tries to grab a virtual object must be considered. Furthermore, every user in a session should not only see the result of the manipulation, but the process too, to increase immersion.
To allow the most natural communication, the holograms has to have the same position, rotation and size in the room for everyone.

If a user starts to manipulate a spacecraft part, an event locking the spacecraft part's manipulation will be sent to the other users. After that, user's application starts publishing the spacecraft part's position and rotation every 30 ms, which means $\tilde{3}0$ updates per second. Since a human perceives a sequence of images of about more than 25 frames per second as fluid [30], a manipulation with this update-rate should be perceived as fluent by the other users too. The process is depicted in figure 3.10. However, if the application is deployed to the HoloLens 2, the manipulation seems to be less fluid than in the Unity engine's simulator. The cause of this must be further researched.

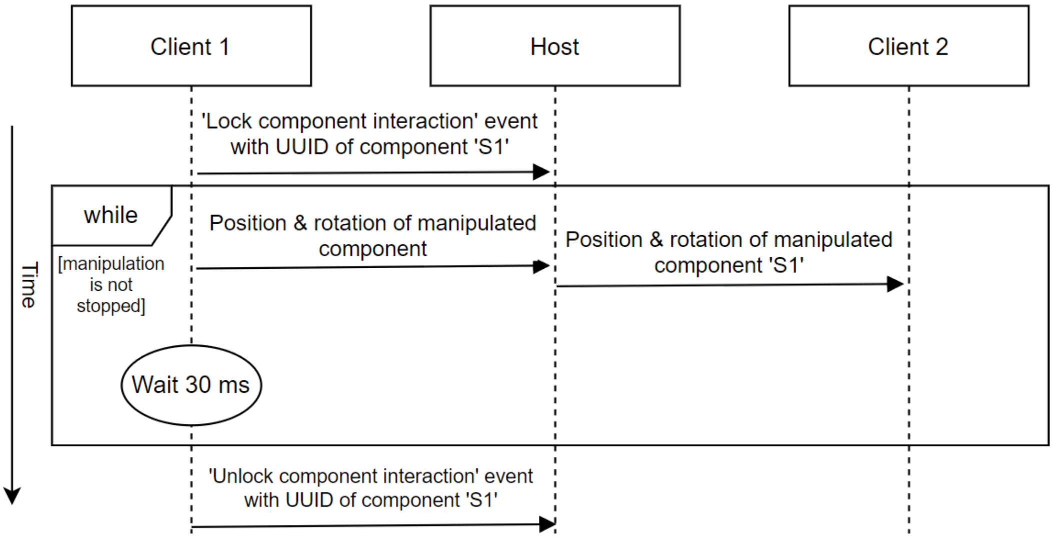


Figure 3.10.: Diagram of the process: "A client manipulates a spacecraft part". Client 1, Client 2 and the host are in a network session. Client 1 manipulates the spacecraft component called 'S1'.

If a user starts the application, its coordinate system's origin will be set where the HoloLens 2 was when it started. Thus, sharing coordinates over network can be problematic. The problem is depicted in figure 3.11.
In order for the virtual content to have the same position, rotation and size for each participant in the shared environment, their application coordinate system origins have to be synchronized.

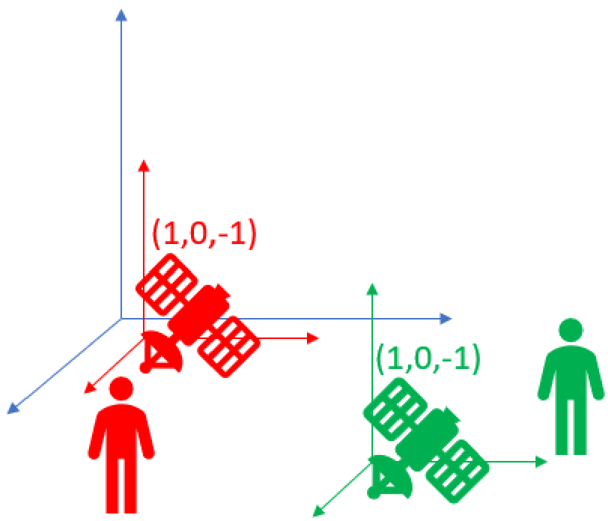


Figure 3.11.: Two people are in a shared environment (blue coordinate system). Because both started their applications on different position in the environment, their application coordinate system origins differ. Thus, their virtual spacecrafts are shown in different positions and orientations.

To achieve that, Microsoft offers its cloud service *Azure Spatial Anchors* [31]. It can persist positions of numerous holograms all over the world. In addition, these can be shared across multiple devices. However, it is a cloud service. Using it would imply a dependency to the internet, a dependency to a third-party service and a violation of DLR IT security guidelines.

Therefore, the computer vision engine *Vuforia* [32] is used. Among other functionalities, it provides detecting and tracking images in a video stream. Such an image is called *Image Target*. The tracking is done by comparing features which where extracted from the stream [33]. A feature means points in an image that can be used for identification, such as edges, corners, ridges or blobs [34].
This functionality is used on the HoloLens 2 integrated webcam stream. A QR code is used as the Image Target which resolves to "42". If the system finds the code, Vuforia will augment the hologram on it. Additional to that, it resets the virtual spacecraft's position, rotation and size relative to its new origin.
To start the tracking process, the user has to click button *Calibrate*, which can be

found on the left of the toolbar. Vuforia then tries to place the visualization on the found image. If the process starts and the user sees the hologram above the QR code, he will be able to stop the process by pushing the button again. If it is stopped, the virtual spacecraft will stay on its new position.

Figure 3.12.: A virtual satellite positioned above an Image Target, represented as a QR code.

3.2.3. Tools in collaborative work

The tools in the application described in section 3.1.2 should also work in a network session. At least one person has to be able to send changes to the server. Moreover, every session's participants virtual spacecraft should stay synchronous, if a user utilizes the *Enlarge Model*, *Shrink Model* or *Model bounds* functionality.

Every user has his own local toolbar. The contained tools implementation for network use are explained in the following.

The *Structure View* selection mode window exists for each user. Every user only sees its own window. If a user activates the *Structure View* mode, the user will only constrain his own selection, others will not be affected.

The *Undo* button reverts the user's last done change on a spacecraft part. If the affected part is currently being manipulated by another user, the operation will fail. If it succeeds, the new position and rotation will be distributed to the other users. Therefore, the tool is useful to undo recent unwanted changes. On the other side, it is not made to reset all changes done in a group. For this, everyone in the group would have to press the *Undo* button in the exact same order the manipulations happened.

If the tool *Model Bounds* is active, a user will be able to manipulate the whole virtual spacecraft's position, rotation and size in the room. Just as in the manipulation of spacecraft parts, messages with position, rotation and size will be sent every 30 ms while manipulating. Therefore, the spacecraft's position, rotation and size will be synchronous between each session's participant. Furthermore, an event is triggered before and after the described sending process on every session's participant. The first locks (disables) the interaction with the application through disabling gesture recognition. The second event unlocks (enables) the interaction with the application. These events are the same used during the transferring process of visualization data between host and client.

To send changes back to the VirSat server, every manipulated spacecraft part's associated CA is saved on a stack by a session's host. The HoloLens 2, which acts as the host is the only device in the session that has the *Send Changes* button and therefore the only one able to send information back to the VirSat server. If the sending process was successful, the stack would be cleared. Furthermore, double references to the same CAs are ignored.

The buttons *Shrink Model* and *Enlarge Model* will send a message to every session's participant with the new size, if one of it is pushed. If the spacecraft is being manipulated while one of these buttons is pushed, the action will fail.

4. Evaluation

This chapter discusses the user study's concept for evaluating the implemented prototype in section 4.1. The results are described in section 4.2. With the statistical software *R*, several statistical procedures are performed and graphs created. Last, section 4.3 discusses the results.

4.1. User study

The implemented prototype is evaluated empirically through a within-subjects user study. For getting the most representative sample, engineers participating in the CEF's processes were invited for testing the system in the DLR's CEF in Bremen. In order to evaluate it, a task had to be performed in small groups of three people.

4.1.1. Tutorial

Since it had to be assumed that no one has experience with the HoloLens 2, a tutorial was created which consisted of several parts:

- Verbal preface
- Tutorial video
- Tutorial task

First, a short introduction to the HoloLens 2 and its gesture interaction was given. Second, the application's different functions (buttons in the toolbar, manipulation of spacecraft parts, manipulation of the whole model) were shown in a video.
Last, every participant could make himself familiar with the application by solving

the tutorial task on a HoloLens 2. It consisted of moving a pink cylinder in a green box and a white sphere on the box (see figure 4.1). To successfully accomplish the tutorial, participants were encouraged to try out the presented functions and to ask questions if necessary.

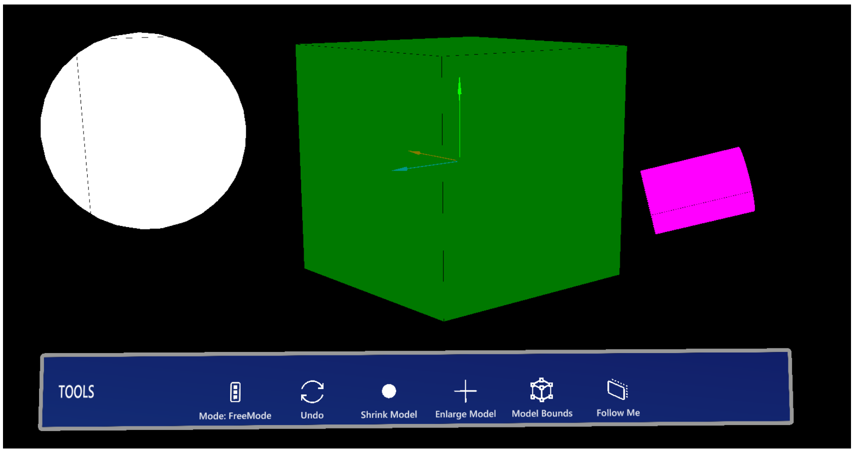


Figure 4.1.: Tutorial task's components and toolbar.

The tutorial video demonstrated a user utilizing the functions while commenting every action done from the user's perspective. A picture of a user's perspective can be seen in figure 4.2.

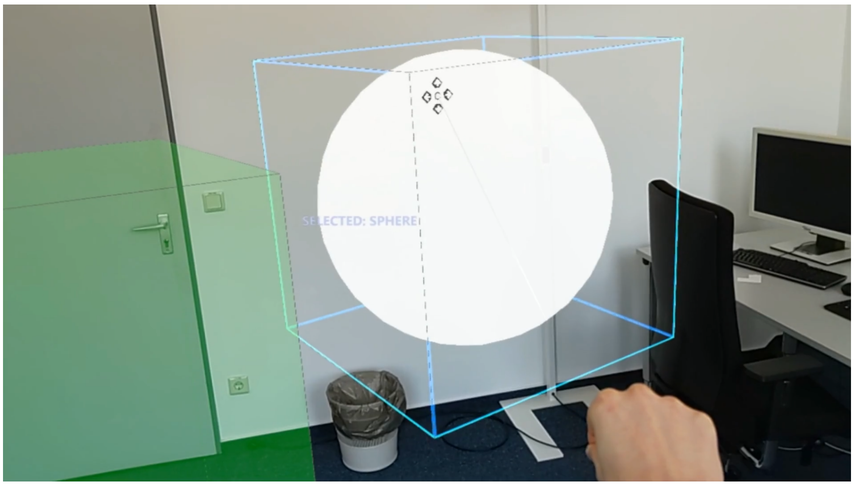


Figure 4.2.: Manipulating a sphere in the tutorial video from a first-person perspective.

4.1.2. Task

For the technical implementation, a router (TP-Link TL-WR940N N450) was used to create a connection between the devices. A laptop ran the VirSat server. By giving every device a static IP address, it was sufficient to create a total of three configuration files for all HoloLens 2 devices: One for the tutorial, one for the host device that fetches data from the laptop's VirSat server and another one for connecting to the host device.

Three main questions are being researched in the scope of this thesis:

1. How well does face-to-face collaboration with AR technology work in the CEF?
2. How high is the potential of resolving conflicts between disciplines in the CEF's spacecraft configuration process?
3. How high is the acceptance of AR technology in the CEF?

The designed task tried to address these questions by forcing the participants to communicate with each other. The task's environment is based on the planned process adaption.

First, the user study's supervisor took a HoloLens 2 and started the application. Then, he fetched the task's visualization from the VirSat server of the laptop. Thus, this device hosted the network session. The supervisor wore it during the whole task. Therefore, he could observe the assembly process.
The other devices were connected to the host. In addition, every device's visualization was calibrated to the QR marker for providing the same position, rotation and size of the visualization for everyone. This procedure was done by the supervisor. Since the application's configuration was only the responsibility of the supervisor, the *Send Changes* button as well as the *Calibrate* button were hidden during the task.

Due to the configuration mentioned before, all participants saw the visualization with the same position, rotation and size in the room. Everyone could manipulate

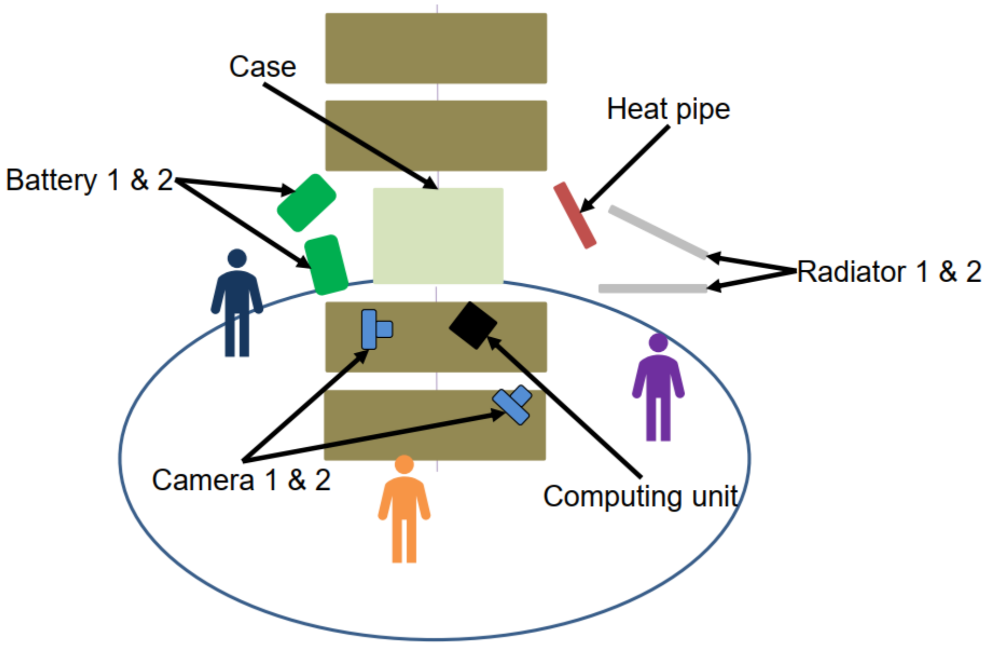


Figure 4.3.: Concept of the task given in the user study.

every part of the spacecraft. Furthermore, every participant could see the other's manipulations in real-time.

At the beginning a raw structure of a satellite was given (a case with solar panels attached, see figure 4.3). Internal parts were located around the satellite in the room with imaginary requirements for each part. In a group of three people, the satellite had to be assembled properly by placing the components according to their requirements. This was a reference to the planned process of designing a draft together which would be handed to the configuration engineer along with other data. For enforcing at least a minimum of communication, each participant was assigned to a unique role which was responsible for a group of components. Therefore, every user only knew about a unique subset of all constraints.

In detail, eight parts were placed around the satellite's case at start: two cameras, one computing unit, one heat pipe and two radiators. These components were distributed to three roles: *Expert for energy supply*, *Expert for thermal control* and *Expert for technical devices*. The roles along with their associated components and their requirements can be seen in table 4.1

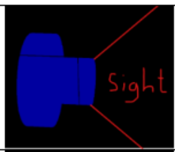
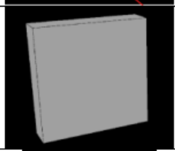
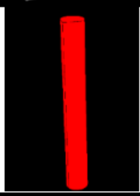
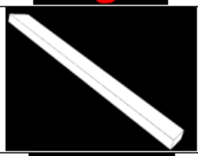
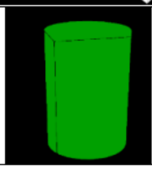
Component	Picture	Role	Requirements
Camera 1 & 2		Technical systems	•Has to be placed outside the case •Has to have free sight
Computing unit		Technical systems	•Has to be placed inside the case •Has to touch both radiators
Heat pipe		Thermal control	•Has to be placed inside the case •Has to be as far away as possible from the radiators and the computing unit
Radiator 1 & 2		Thermal control	•Has to be inside & outside the case
Battery 1 & 2		Energy supply	•Has to be placed inside the case •Has to be placed near a camera

Table 4.1.: User study task's components with their assigned roles and requirements.

As support, every participant received a DIN-A4 handout with information regarding their role and an overview of all available parts. Furthermore, the first third contained a short overview describing every button in the toolbar. All handouts can be found in appendix A. A photo of the executed user study is shown in figure 4.4.

During the task, the groups assembled the spacecraft together. By communicating with each other, all groups assembled it correctly.

The task was over, if the group assembled the satellite successfully or if the specified length of the user study was in danger of being exceeded significantly. The latter did not happen.

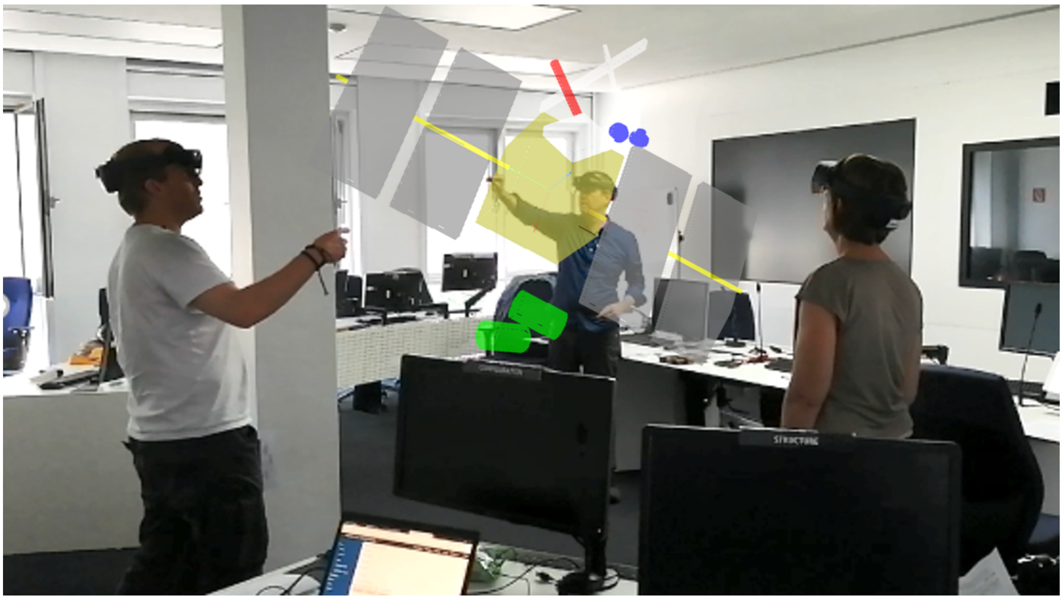


Figure 4.4.: Participants assembling a satellite during the user study at CEF in Bremen.

4.1.3. Questionnaire

To evaluate the application, a questionnaire (appendix B) was used to quantify user impressions. It contained *Likert scales* ranging from one to five for rating and a rating of 5 — 100 for *NASA TLX* questionnaire.

The questionnaire consists of several categories for different purposes. First, four questions target the user's experience about AR/VR, HoloLens 1/2, VirSat and its use frequency (Q1 - Q4). After that, the questionnaire *System Usability Scale* by John Brooks [35] is used to evaluate the perceived user experience. It allows calculating a score when following an algorithm, which is explained in section 4.2 (Q5 - Q14). For evaluating the collaboration and use for the CEF, eight additional questions are included asking about perceived communication, collaboration, desired use frequency in the CEF, the application's potential to solve interdisciplinary conflicts and two additional questions that target usability (Q15 - Q23) in the category *Collaboration*. The category *Software* contains a list of questions about the application functionalities. The results can potentially guide the further development of the prototype (Q24 - Q29).

Last, the widely approved NASA Task Load Index (TLX) questionnaire is used to collect information about the task's demands which can give evidence for the task's evaluation. Further, it gives information about mental impact of the application on its users. The TLX is rated with a *Likert* scale ranging from five to 100 in five-point steps.

4.2. Results

4.2.1. User demographics

To evaluate the application and collaboration both subjectively and objectively, we held a within-subjects user study with 11 participants, 1 female, and 10 males. The age range was 25 to 56 years, with the average age 40.7 years.

Figure 4.5 describes the experience regarding VirSat, HoloLens 1/2 and AR/VR as well as if VirSat was used frequently. Here 1 means they have no experience / never used and 5 means much experience / used daily. Participants were mostly unfamiliar with the Microsoft HoloLens 1 or 2 with a *Mean* (M) of 1.6 and a *Standard Derivation* (SD) of 1, but had a bit more experience with AR/VR in general ($M = 2.4, SD = 0.9$). Most of the people had experience in VirSat ($M = 3.2, SD = 1.1$), apart from one outlier, that did not have any knowledge about it. Only one participant never used VirSat ($M = 2.2, SD = 0.9$).

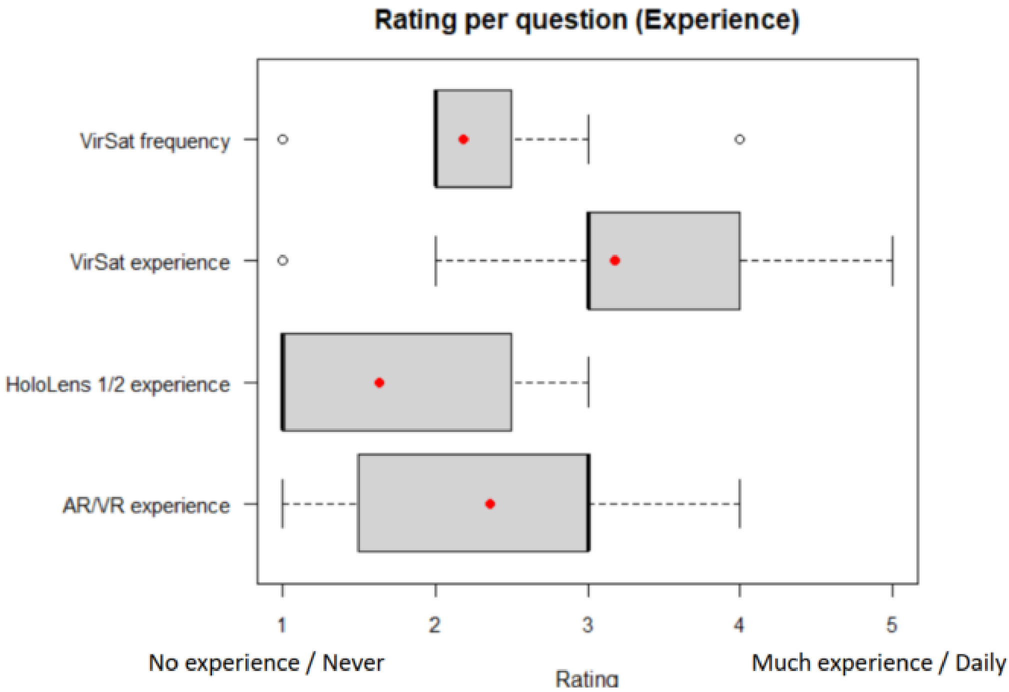


Figure 4.5.: Box plots showing the ratings of questions regarding *Experience*.

4.2.2. Usability

This subsection evaluates the questionnaire's System Usability Scale (SUS). It calculates every participant's score and tests the mean score's significance with a *one tailed one-sample t-test*.

To calculate a user's SUS score, we add $result - 1$ for every question with an odd position (1,3,5,7,9) (within the SUS questionnaire) but add $5 - result$ for every other question. The outcome multiplied with 2.5 is the SUS score.

A *Shapiro test* with following hypotheses allows a rejection of H0, which gives evidence that the SUS scores are normally distributed ($\alpha = 0.05, p = 0.14$):

1. H0: The scores are not normally distributed
2. HA: The scores are normally distributed

Now, a one tailed t-test is performed for checking if the calculated SUS score (77,3) is statistically significant above the average score of 68. The result is, that the score is statistically significant higher than the average SUS score ($\alpha = 0.1, t = 1.89, p = 0.044$). In addition, research [36] shows, that a score of 77.3 can be labelled as “Good”. As can be seen in figure 4.6, only one participant rated the system’s usability way under average.

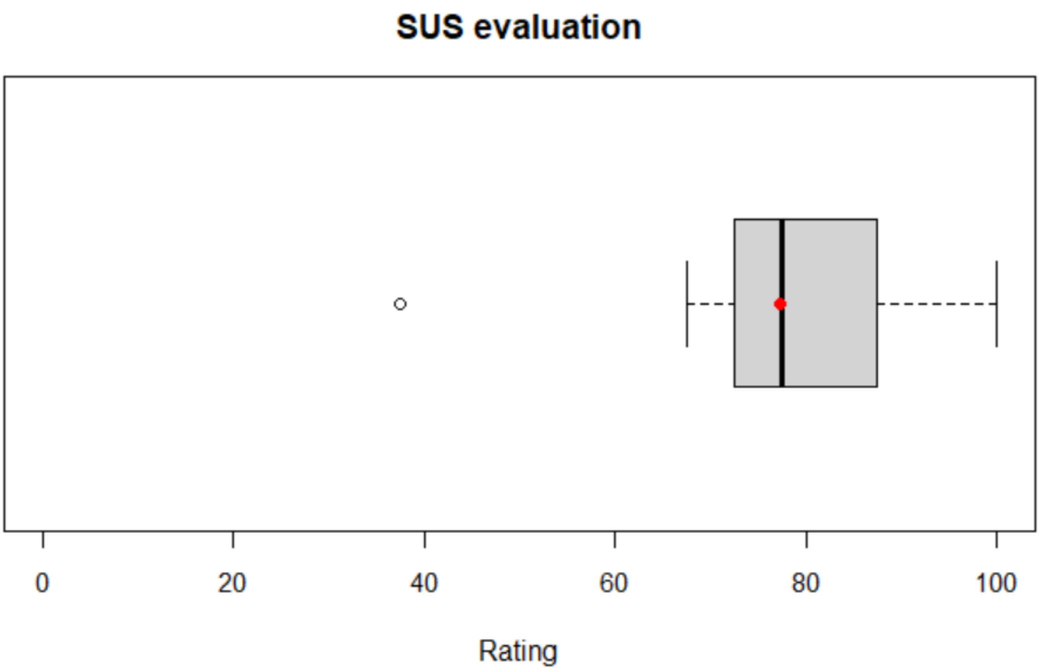


Figure 4.6.: Box plot showing the calculated *SUS* scores.

Learning needed (Q14) ($m = 1, M = 1.45, SD = 0.93, min = 1, max = 4$), *Quick learning* (Q11) ($m = 5, M = 4.45, SD = 0.93, min = 2, max = 5$) and *Easy use* (Q7) ($m = 4, M = 4.18, SD = 0.87, min = 2, max = 5$) (see figure 4.7) indicate that the entry hurdle is not perceived as particularly high in general. Regardless, the averagely only very slightly disagree on needing a technical person for using the system (Q8) ($m = 2, M = 2.18, SD = 1.17, min = 2, max = 5$). Also, the average opinion about a frequent use of the application was rather reserved ($m = 3.36, M = 4, SD = 1.03, min = 2, max = 5$).

Additional to that, few users reported that gestures, especially the *Grab* gesture, were not properly recognized by the HoloLens 2 which led to frustration, as could be observed during the study. These observations coincide with few participants having rated questions regarding the application’s use negatively (see 4.7).

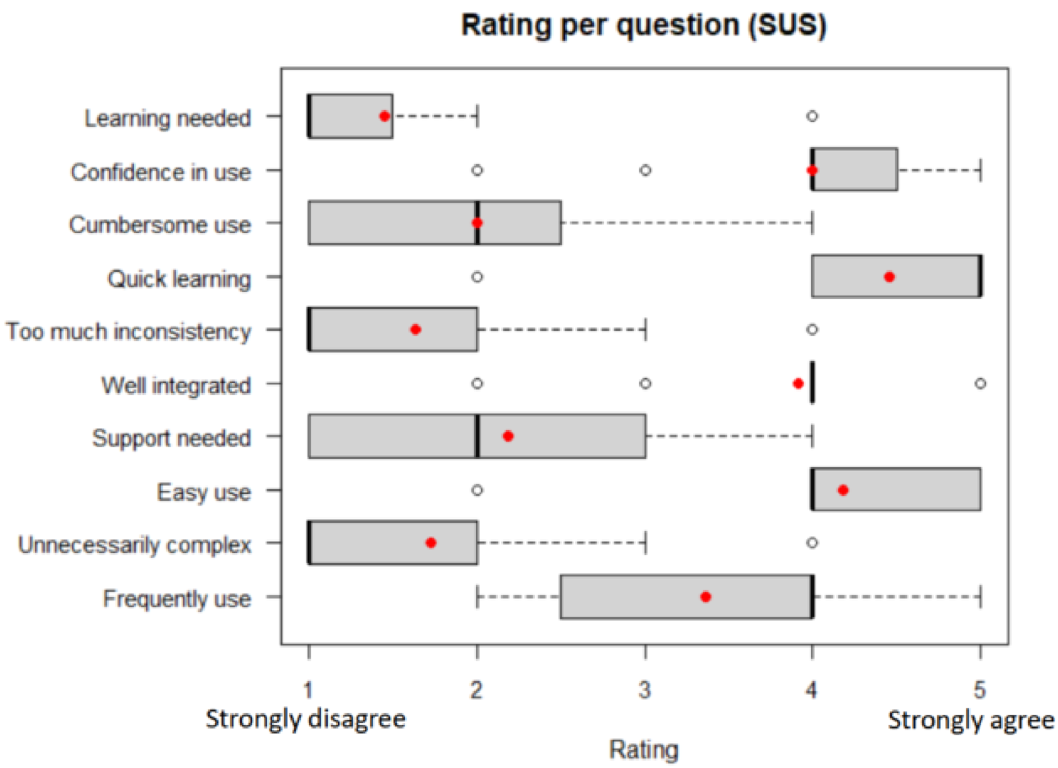


Figure 4.7.: Box plots showing the ratings of questions regarding *SUS*.

4.2.3. Mental demand

This subsection evaluates the questionnaire's *NASA TLX* part in order to observer the task's mental demand. Its questions have a scale ranging from five to a hundred points in five-point steps. 5 means *Very Low / Perfect* and 100 *Very High / Failure*.

Shown in figure 4.8, the results depict that the task was in general more mentally demanding ($M = 38.8, SD = 21.9$) then physically ($M = 17.1, SD = 15.5$). Despite that every group finished the task successfully, the own success was perceived various with a minimum of 5 and a maximum of 95 points ($M = 17.1, SD = 15.5$). 25% of the participants said that they rather failed the task (third quartile: 75 points). *Confidence in use* (Q13) correlates weakly with Q33 *Task success* conforming to the *Spearman test* ($\alpha = 0.1, p = 0.077$) using following hypotheses:

1. H0: Q13 does not correlate negatively with Q33
2. HA: Q13 does correlate negatively with Q33

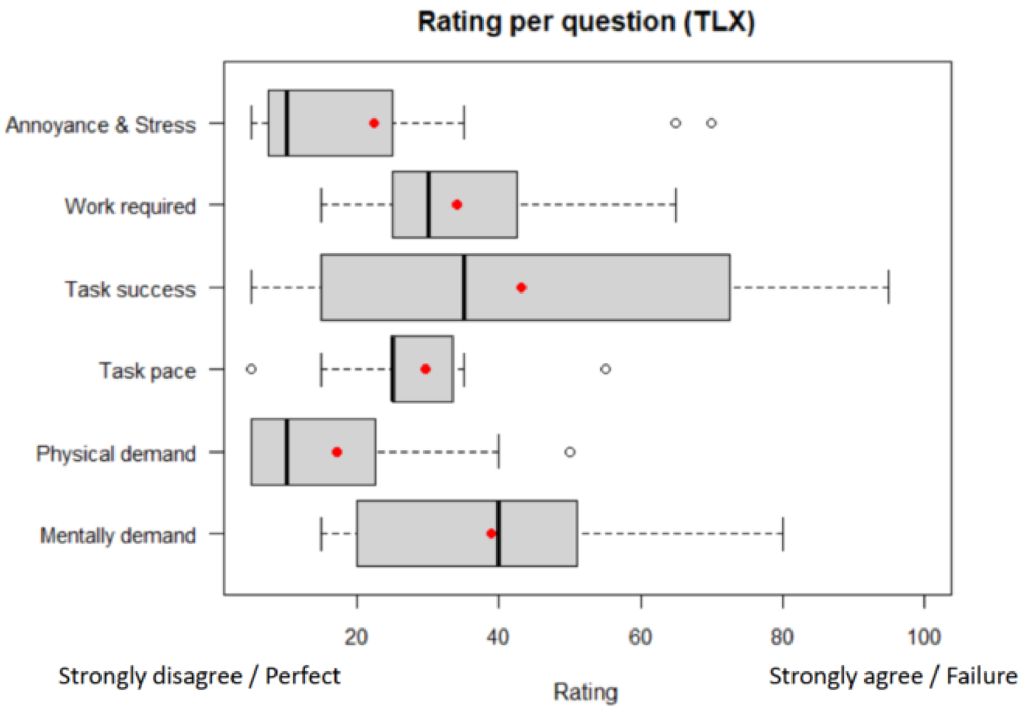


Figure 4.8.: Box plots showing the ratings of questions regarding *TLX*.

It is assumed that people have a higher chance of succeeding in task if they are confident in using it . Thus, we execute the *Spearman-test* for a negative correlation.

The level of annoyance as well as stress was not particularly high with a *median* of 10, but with two outliers describing the level with at least 60 points ($m = 10, M = 22.27, SD = 23.91, min = 5, max = 70$).

4.2.4. Software features

This subsection focuses on results regarding the questionnaire's *Software* part for supporting the application's development in the future.

With 3 as the median and 2.91 as the mean ($SD = 1.3, min = 1, max = 5$), the participants generally did not think that the precision in manipulation was enough. That coincides with verbal and written comments. They suggested adding features like an automatic snapping of spacecraft parts to other surfaces, or a system which enables a smoothing of the angles after a rotation.

In general, all three functions were found at least helpful when looking at the medians and means ($M_{\text{Mode}} = 3.27, m_{\text{Mode}} = 4, M_{\text{Undo}} = 4.00, m_{\text{Undo}} = 5, M_{\text{ModelManipulation}} = 4.36, m_{\text{ModelManipulation}} = 5$), while the functionality of manipulating the whole model being perceived as the most useful one.

The *mode* functionality's perceptive usefulness (Q26) correlates positively with its frequency of use (Q25), according to the *Spearman-test* ($\alpha = 0.1, p = 0.001$) with the following hypothesis:

1. H0: Q25 does not correlate positively with Q26
2. HA: Q25 does correlate positively with Q26

It is assumed that people want to use a system more often when they find it useful. Thus, we execute the *Spearman-test* for a positive correlation.

On the other hand, the perceptive usefulness of the *Undo* function (Q28) does not correlate with frequency of its use (Q27) according to the *Spearman-test* ($\alpha = 0.1, p = 0.218$) with following hypotheses:

1. H0: Q27 does not correlate positively with Q28
2. HA: Q27 does correlate positively with Q28

Because the questionnaire contained no questions regarding performance or stability, observations from the user study are described in the following. Once there was a malfunction, which led to a loss of connections for all participants. Just as the third group finished their satellite, the error occurred. The most likely reason for this is that a simultaneously started video recording on the host device destabilized the system for a short period of time, which probably led to a build-up of unanswered *Heartbeat* messages on the clients until the maximum level for disconnecting was reached.

Apart from this incident, nothing regarding performance of the system was reported by the participants. In addition, the *Calibration* system worked, but was perceived as tricky to the head of the study and therefore has to be improved further for increasing its usability.

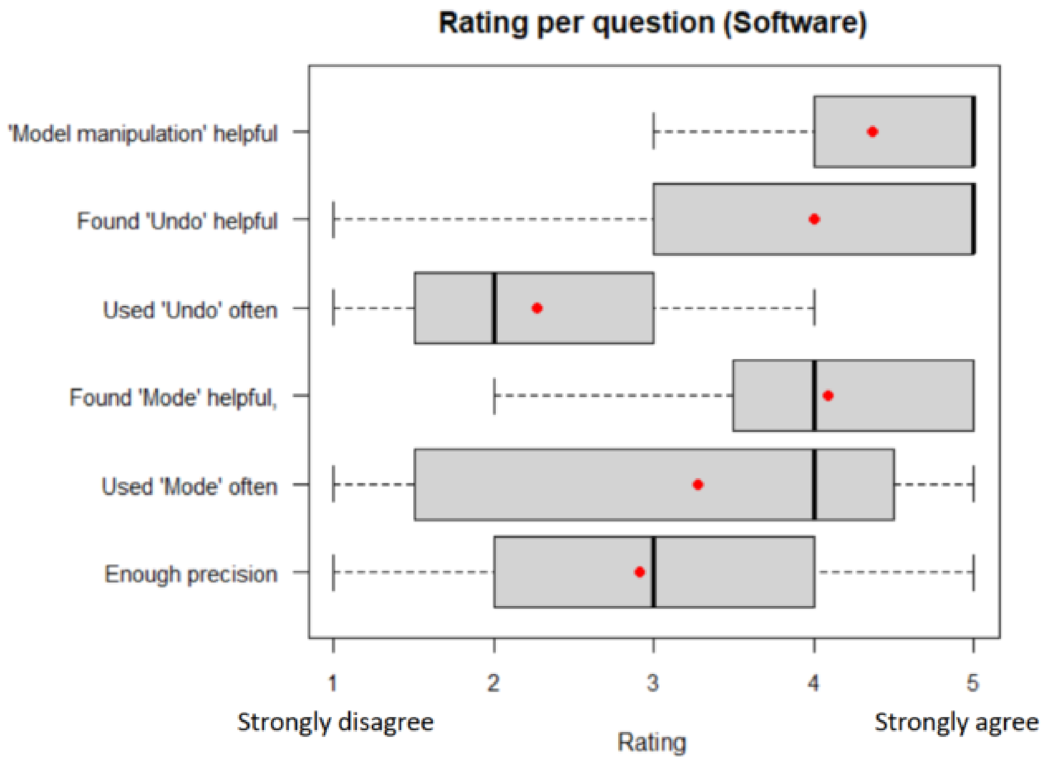


Figure 4.9.: Box plots showing the ratings of questions regarding *Software*.

4.2.5. Collaboration & CEF

This subsection observes the questionnaire's crucial part *Collaboration*, which contains questions about the application's possible use in the CEF and perception of collaboration. 1 equals *Strongly disagree* and 5 *Strongly agree*.

Q19 ("I found the system to be very frustrating") is omitted because frustration is addressed through the NASA TLX questions. Q21 ("I think the system was intuitive to use") is ignored as well because of its redundancy due to Q12 ("I found the system very cumbersome(awkward)".

First, the questions about the collaboration are being discussed.
Figure 4.10 shows that communication and collaboration during the task were generally good. Regardless of some participants mentioned not having paid attention to facial expressions or gestures of the others, the communication was widely perceived as natural (Q17) with a mean of 4.18 and median of 4 ($SD = 0.75, min = 3, max = 5$). Also, users found the communication to have worked out well during the task (Q18) ($m = 4, M = 4.27, SD = 0.65, min = 3, max = 5$). In addition,

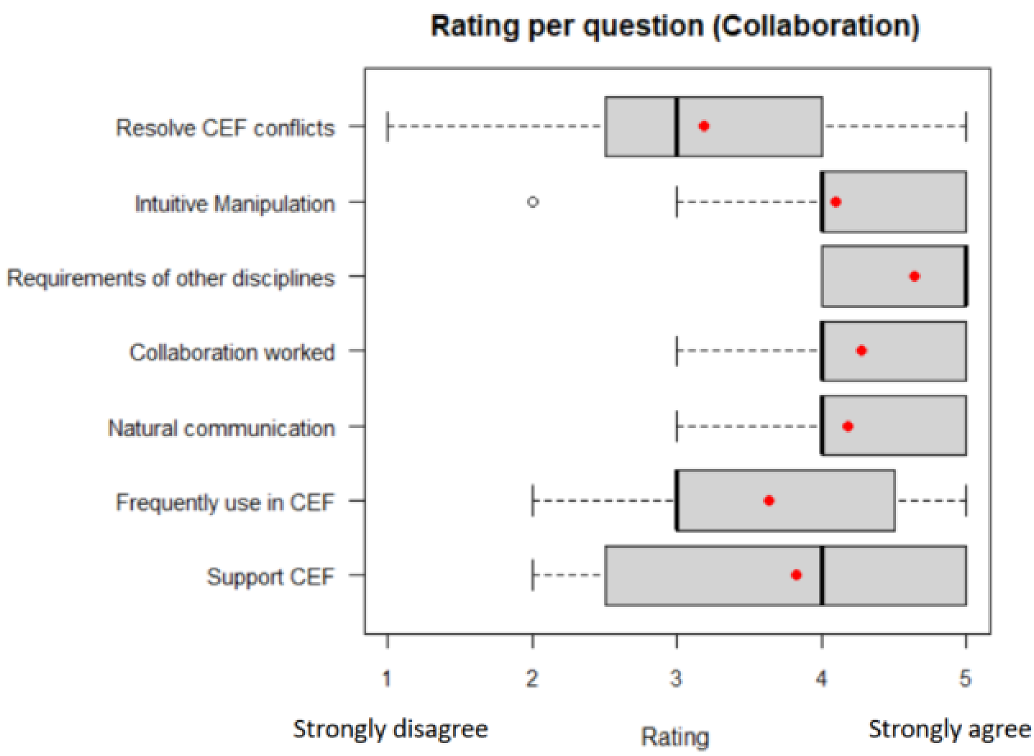


Figure 4.10.: Box plots showing the ratings of questions regarding *Collaboration*.

the requirements of other disciplines were well understood with a mean of 4.64 and median of 5 ($SD = 0.5, min = 4, max = 5$). Furthermore, the placement of the spacecraft parts felt intuitive for the most participants ($m = 4, M = 4.09, SD = 0.94, min = 2, max = 5$), besides one outlier that rated it with 2.

However, the questions regarding benefits of using the application in the CEF are more controversial. With a median of 4 and mean of 3.82 ($SD = 1.3, min = 2, max = 5$), 50% thought that the application could improve the process (rating >= 4) in the CEF. A bit more than a quarter (27%) gave a rating of two or lower, stating that they thought the system could rather not support it. Furthermore, a frequent use of such a system in the CEF was desired by a slim majority with a median of 3 and mean of 3.64 ($SD = 1.03, min = 2, max = 5$) which coincides with the results of frequent use in general (Q5). Last, the participants were rather uncertain about the application's ability to help resolving conflict with other disciplines in the CEF, with a light tendency to agreement ($m = 3, M = 3.18, SD = 1.17, min = 1, max = 5$).

Furthermore, some verbal and written comments stated that they have thought about using AR technology to improve the process before. On the other side a participant warned about developing new tools with such sophisticated technology due to its learning hurdle.

It is worth mentioning that all CEF related questions (Q15, Q16, Q23) strongly correlate with the given SUS score, according to following tests with the assumption that a higher SUS score leads to a higher acceptance for using it in the CEF:

1. *Spearman-test* with SUS score and Q15 ($\alpha = 0.1, p = 0.002$)
   a) $H0_{\text{Score Q15}}$: Score does not correlate positively with Q15 (*Support CEF*)
   b) $HA_{\text{Score Q15}}$: Score does correlate positively with Q15 (*Support CEF*)
2. *Spearman-test* with SUS score and Q16 ($\alpha = 0.1, p = 0.004$)
   a) $H0_{\text{Score Q16}}$: Score does not correlate positively with Q16 (*Frequently use in CEF*)
   b) $HA_{\text{Score Q16}}$: Score does correlate positively with Q16 (*Frequently use in CEF*)
3. *Spearman-test* with SUS score and Q23 ($\alpha = 0.1, p = 0.013$)
   a) $H0_{\text{Score Q23}}$: Score does not correlate positively with Q23 (*Resolve conflicts in CEF*)
   b) $HA_{\text{Score Q23}}$: Score does correlate positively with Q23 (*Resolve conflicts in CEF*)

It is assumed that people want to use a system more often when they like its usability. Thus, we executed the three *Spearman-tests* for a positive correlation.

4.3. Discussion

Most participants had experience with VirSat but very few with the HoloLens 1/2. Besides the wide age range, some people wore glasses and one participant was on crutches. Against this background it can be said that the participants as a whole covered a broad spectrum.

The application's usability was rated statistically significant above average and can be labelled as "Good" with a score of 77.3. Nevertheless, a lot of written and verbal comments focused on tools or features for improving manipulation of spacecraft parts. Furthermore, it should be noted that participants did not have any contact with the opening menu nor did they have to calibrate the visualization to with QR code. This could have impact on the resulting SUS score. Moreover, the learning barrier and ease of use was perceived as rather low, although the need for a technically versed person for using the application was controversial and not strongly opposed. Also, lacking depth and realism of the holograms were criticized.

Tools recommended by participants, regarding spacecraft part manipulation were an automatic smoothing of angles to other spacecraft parts or being able to set relationships between two parts (e.g. box A has always a fixed position/rotation to box B).

The *Mode* function was in general used more often than *Undo*, while both were seen as useful in average, as well as the *Model manipulation*. Therefore, they should not be discarded, but improved. Furthermore, the window for the mode *Structure View* was commonly criticized for its default behaviour of following the user's sight and its buttons being hard to touch. According to that, the window's default behaviour should be staying in the environment and only following the user's sight if desired (button push) and the entry buttons ability to be pushed should be improved (e.g. by enlarging their colliders).

Also, the gesture recognition was criticized. Especially the *Grab* gesture was reported as most problematic, although more frequently at the beginning than at the end of the study, which could be attributed to the learning effect and familiarization with the system. Additional to that, the HoloLens 2 sometimes recognized hand movements as gestures without the user's intention which led to frustration.

Although most participants stated that their stress level during the task was rather low, two participants perceived it as quite high instead, one even reported eye pain after the task. Therefore, the mental and physical impacts of the application and

HoloLens 2 have to be investigated further. Despite these rare cases, the task was not physically demanding for the most participants, but slightly mentally demanding. It also should be noted that the perceived success of the task was controversial, although every group finished their task successfully. The perceived success correlates weakly with a user's confidence in using the application. Furthermore, the required work and task pace were perceived rather low. From these results it is concluded that the task was appropriate. Nevertheless, the tutorial time or its quality might not have been sufficient.

The questionnaire's results show, that communication between the participants of the task was perceived as good. The users rated the collaboration very positive as well. Regardless, some mentioned that they focused more on the task, then on other people's gestures and mimics. Furthermore, the manipulation of spacecraft parts was perceived as intuitive.

Results regarding the questions about using an AR-system in the CEF are more restrained. Most participants found, that the application could support the process in the CEF. Nevertheless, it is suspected, that the word "process" in the question could mean the configuration of the spacecraft parts (intended) as well as the whole process happening in the CEF (not intended). The strongly varied ratings about if the application could resolve conflicts supports this assumption.
Most people thought that the application could be helpful in the CEF. A frequent use was less desired which could be explained by the fact that this software is only supposed to support a part of a larger process happening a few times per year (according to participants), so a frequent use could inhibit the whole spacecraft design process. The results combined with verbal comments by the participants lead to the conclusion that a cautious acceptance exists for such a technology.

Moreover, it should be noted that all questions regarding the application's use in the CEF strongly positively correlate with the SUS score, which could be a hint to the importance of a tool's usability for a wide acceptance in the CEF.

5. Conclusion

In this work, a prototypical application for the Microsoft HoloLens 2 was developed to prevent conflicts during the configuration process in the Concurrent Engineering Facility (CEF). To research its usability, usefulness and acceptance, the prototype was evaluated afterwards through a user study. The work is summarized in the following and an outlook is given afterwards.

5.1. Summary

First, the theoretical background was explained through describing the CEF and spacecraft development, as well as the state of the art of design conflict solving in engineering. It included the CEF's data model Virtual Satellite 4 and its stand-alone server. Furthermore, Augmented Reality (AR) technology and the state of the art collaboration in AR was presented. In addition, the AR headset Microsoft HoloLens 2 was depicted.

The prototype running on HoloLens 2 was built in Unity and allows concurrent manipulations on the same visualization over network. For communication, a client-server architecture was built with the open source library *NetMQ 4*. All manipulations can be observed by other participants in real-time and several tools were implemented to support the manipulation. Moreover, the visualization data comes from Virtual Satellite 4 which is used as the integrated data model during the spacecraft development process.

Afterwards, the prototype was evaluated through a user study in the CEF of the German Aerospace Center (DLR). Small groups of three people had to collaboratively solve a given task in the application. With the help of a questionnaire, we tried to find out about the application's acceptance in the CEF as well as its ability to support communication and resolve interdisciplinary conflicts during the spacecraft configuration process. In addition, its usability was investigated.

The evaluation showed, that the prototype's usability can be rated 'Good'. However, the participants did not have contact to every functionality of the software. Furthermore, the need for technical support when using the application was not strongly opposed, although the learning barrier was not perceived as high. Also, the visualization was criticized for lacking depth and realism.
The application's manipulation tools *Undo*, *Mode* and *Model Bounds* are useful, but were used differently often. The manipulation of virtual objects feels intuitive, but its precision was not sufficient. This applies in particular to the rotation of holograms.

The perceived success of the task was controversial, although every user study's tasks were finished successfully. The perceived success correlates weakly with the user's confidence in using the application which allows the conclusion that not enough confidence could be built up with the tutorial before the task. The user study task's physical demand was low and therefore slightly lower than the mental demand. However, the mental demand was rather low as well. A small part of participants stated a high mental demand, one participant reported eye pain.

The application's relevance for solving interdisciplinary conflicts during the spacecraft configuration process is controversial. Yet, the results lead to the conclusion that AR technology like the Microsoft HoloLens 2 enables effective and natural communication. Also, the application could be useful in the CEF and acceptance for such technology is given. It should be noted, that a strong correlation exists between the application's SUS score and the ratings regarding its usefulness in the CEF, which gives a hint about the importance of an application's usability for its acceptance.

5.2. Future Work

In order to decrease the need of technical support, a system could be integrated that informs the user about tools and can give tips for using the system's features. For example, this could be done with adding voice commands like "Help" or "Tips" that will give information of a function (e.g. a focused button) on demand.
Furthermore, lighting and more advanced shaders could be added to the visualization to create more depth while being cautious about performance.

Also, the manipulation's precision, most importantly the rotation's precision, has to be improved. For that, new tools could be implemented. For example, a configurable automatic angle smoothing could be implemented, that automatically rounds the current angle when rotating a hologram. On the other hand, emphasis should be placed on not bloating the system with new features. Due to partly changing participants in CEF studies, simplicity is important.

The application's tools *Undo*, *Mode* and *Model Bounds* should not be discarded but improved instead. Furthermore, the visualization of hand rays should be considered which would allow users to see what exact object another user is pointing at. This could be especially helpful if the application's extension to support remote collaboration would be desired.

Besides, a closer look must be taken on the system's mental demand and effects to humans. Also, there are hints that users may need a more effective tutorial in the future.

Due to the positive results regarding collaboration and communication, we recommend considering using AR technology like the HoloLens 2 for situations where natural collaboration is needed. Because of the strong correlation between usability and acceptance in the CEF, usability should be focused in further development. Also, research has to be conducted on finding other possible uses in the spacecraft design phase. For that, communication with the CEF's engineers is crucial.

Bibliography

[1] P. M. Fischer, R. Wolff, and A. Gerndt, "Collaborative satellite configuration supported by interactive visualization," in *2012 IEEE Aerospace Conference*, 2012, pp. 1–11.

[2] "Concurrent engineering facility (cef)," accessed 17.08.2021. [Online]. Available: https://www.dlr.de/content/de/grossforschungsanlagen/concurrent-engineering-facility.html

[3] C. J. Haddad, "Operationalizing the concept of concurrent engineering: a case study from the us auto industry," *IEEE Transactions on Engineering Management*, vol. 43, no. 2, pp. 124–132, 1996.

[4] W. Quan and H. Jianmin, "A study on collaborative mechanism for product design in distributed concurrent engineering," in *2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design*. IEEE, 2006.

[5] D. Knoll, C. Fortin, and A. Golkar, "Review of concurrent engineering design practice in the space sector: state of the art and future perspectives," in *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2018.

[6] DLR, "Virtual satellite 4 core: User manual," accessed 13.08.2021. [Online]. Available: https://github.com/virtualsatellite/VirtualSatellite4-Core/releases/download/Release_4.13.0/VirSat_Core_Release_4.13.0_08b129cb4888ebf263db82a60f3f06a7bab970b9_Win32.zip

[7] Roy Thomas Fielding, "Chapter 5: Representational state transfer (rest)," 2000, accessed 21.08.2021. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[8] H. Subramanian and P. Raj, *Hands-On RESTful API Design Patterns and Best Practices*. Birmingham, England: Packt Publishing, 2019.

[9] DLR, "Virtual satellite 4 core: Server manual," accessed 12.07.2021. [Online]. Available: https://github.com/virtualsatellite/VirtualSatellite4-Core/releases/download/Release_4.13.0/VirSat_Server_Release_4.13.0_08b129cb4888ebf263db82a60f3f06a7bab970b9_Win32.zip

[10] P. Milgram, D. Drascic, J.J. Grodski, A. Restogi, and C. Zhou, *Merging Real and Virtual Worlds*, 1995. [Online]. Available: https://www.researchgate.net/profile/david-drascic/publication/246807797_merging_real_and_virtual_worlds

[11] *VIRTUAL UND AUGMENTED REALITY (VR/AR): Grundlagen und methoden der virtuellen*. SPRINGER-VERLAG BERLIN AN, 2019.

[12] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[13] D. J. C. Matthies, M. Haescher, R. Alm, and B. Urban, "Properties of a peripheral head-mounted display (phmd)." Springer, Cham, 2015, pp. 208–213. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-21380-4_37

[14] Google, "Discover glass enterprise edition," accessed 05.10.2021. [Online]. Available: https://www.google.com/glass/start/

[15] Microsoft, "Join the mixed reality partner program," accessed 05.10.2021. [Online]. Available: https://www.microsoft.com/de-de/hololens/mrpp

[16] Sarah Cooley, Jitin Mathew, "JoyJaz", Evan Miller, ". v jodben", and Seth Paniagua, "About hololens 2," 2020, accessed 21.08.2021. [Online]. Available: https://docs.microsoft.com/en-us/hololens/hololens2-hardware

[17] Canggih Puspo Wibowo, *Evaluation of Protocol Buffers as Data Serialization Format for Microblogging Communication*, 2011, accessed 21.09.2021. [Online]. Available: https://www.academia.edu/download/37191710/Evaluation_of_Protocol_Buffers_as_Data_Serialization_Format_for_Microblogging_Communication.pdf

[18] P. Montiel-Overall, "Toward a theory of collaboration for teachers and librarians," *School Library Media Research*, vol. 8, 2005. [Online]. Available: https://eric.ed.gov/?id=ej965627

[19] D. Phutela, *IUP Journal of Soft Skills: The Importance of Non-Verbal Communication*, 9th ed., 2015, accessed 29.09.2021. [Online]. Available: http://library.tuit.uz/knigiPDF/Ebsco/8-1106.pdf

[20] K. Kiyokawa, M. Billinghurst, S. Hayes, A. Gupta, Y. Sannohe, and H. Kato, "Communication behaviors of co-located users in collaborative ar interfaces," in *Proceedings. International Symposium on Mixed and Augmented Reality*, 2002, pp. 139–148.

[21] Z. Szalavri, D. Schmalstieg, A. Fuhrmann, and M. Gervautz, "Studierstube: An environment for collaboration in augmented reality," *Virtual Reality*, vol. 3, no. 1, pp. 37–48, 1998.

[22] X. Wang and P. S. Dunston, "Comparative effectiveness of mixed reality based virtual environments in collaborative design," in *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009.

[23] M. Deshmukh, R. Wolff, P. M. Fischer, M. Flatken, and A. Gerndt, "Interactive 3d visualization to support concurrent engineering in the early

space mission design phase," in *Challenges in European Aerospace, CEAS Air & Space Conference*, 2015. [Online]. Available: https://elib.dlr.de/101924/

[24] Artur Baranowski, "3d-bauteilmanipulation zur satellitenkonfiguration unter verwendung von mixed-reality-technik," 2018, accessed 12.07.2021. [Online]. Available: https://elib.dlr.de/121073/

[25] N. A. Borromeo, *Hands-on Unity 2020 game development: Build, customize, and optimize professional games using Unity 2020 and C#*. Birmingham: Packt Publishing, Ltd, 2020.

[26] Unity, "Unity real-time development platform," accessed 11.08.2021. [Online]. Available: https://unity.com/

[27] Unity, "Unity: Scriptingreference," accessed 10.08.2021. [Online]. Available: https://docs.unity3d.com/2019.4/Documentation/ScriptReference/index.html

[28] Yoon Park, "Vinayak0706 ", David Coulter,Harrison Ferrone, " Kurtis", "Introducing mrtk for mixed reality," accessed 11.08.2021. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/mrtk-getting-started

[29] "What is the mixed reality toolkit," 03.03.2021, accessed 25.08.2021. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/

[30] A. Butz and A. Krüger, *Mensch-MaschineInteraktion*. DE GRUYTER OLDENBOURG, 2014.

[31] Microsoft, "Spatial anchors," accessed 11.10.2021. [Online]. Available: https://azure.microsoft.com/en-us/services/spatial-anchors/

[32] PTC, "Vuforia," accessed 09.10.2021. [Online]. Available: https://www.ptc.com/en/products/vuforia

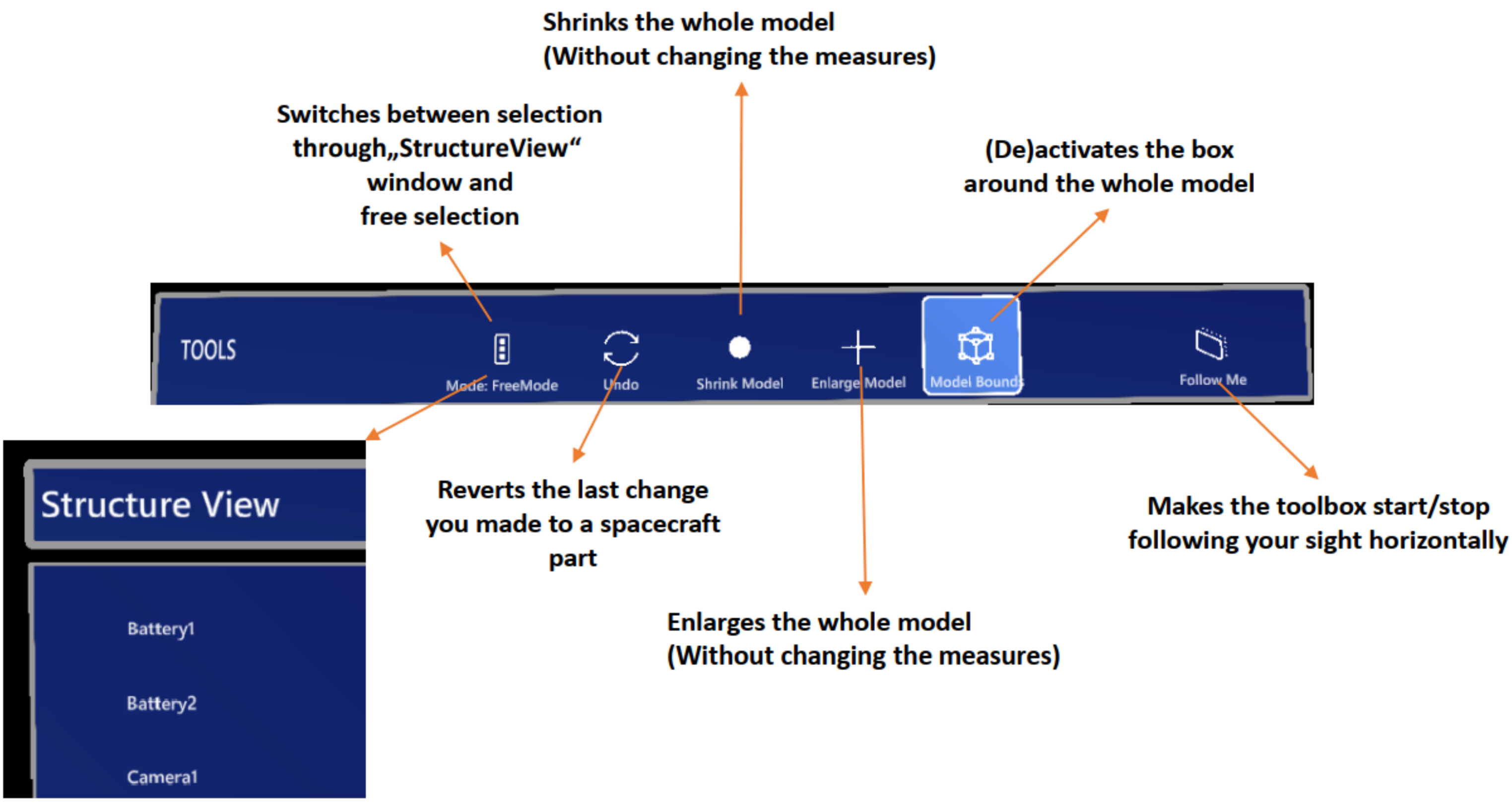
[33] PTC, "Image targets," accessed 11.10.2021. [Online]. Available: https://library.vuforia.com/features/images/image-targets.html

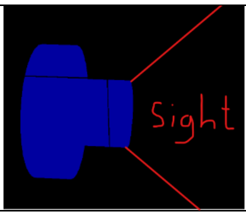
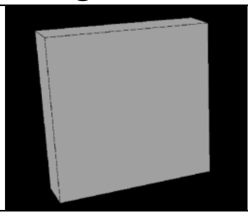
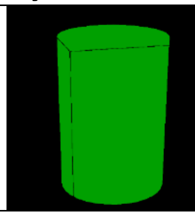
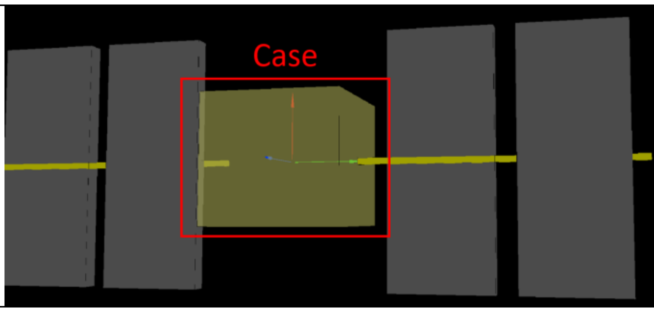
[34] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: A comprehensive survey," in *2017 IEEE international conference on industrial technology (ICIT)*. IEEE, 2017, pp. 1059–1063.

[35] John Brooke, "Sus: A 'quick and dirty' usability scale," in *Usability Evaluation In Industry*. CRC Press, 1996, pp. 207–212. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.1201/9781498710411-35/sus-quick-dirty-usability-scale-john-brooke

[36] A. Bangor, P. Kortum, and J. Miller, *Determining what individual SUS scores mean: Adding an adjective rating scale*, 2009. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.177.1240&rep=rep1&type=pdf

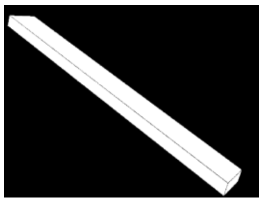
A. Handouts



Camera 1 & 2	Computing unit	Battery 1 & 2
		
Satellite		
		

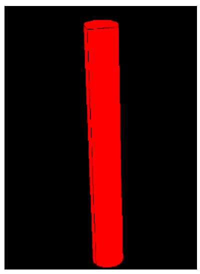
Your role:

Expert for thermal control



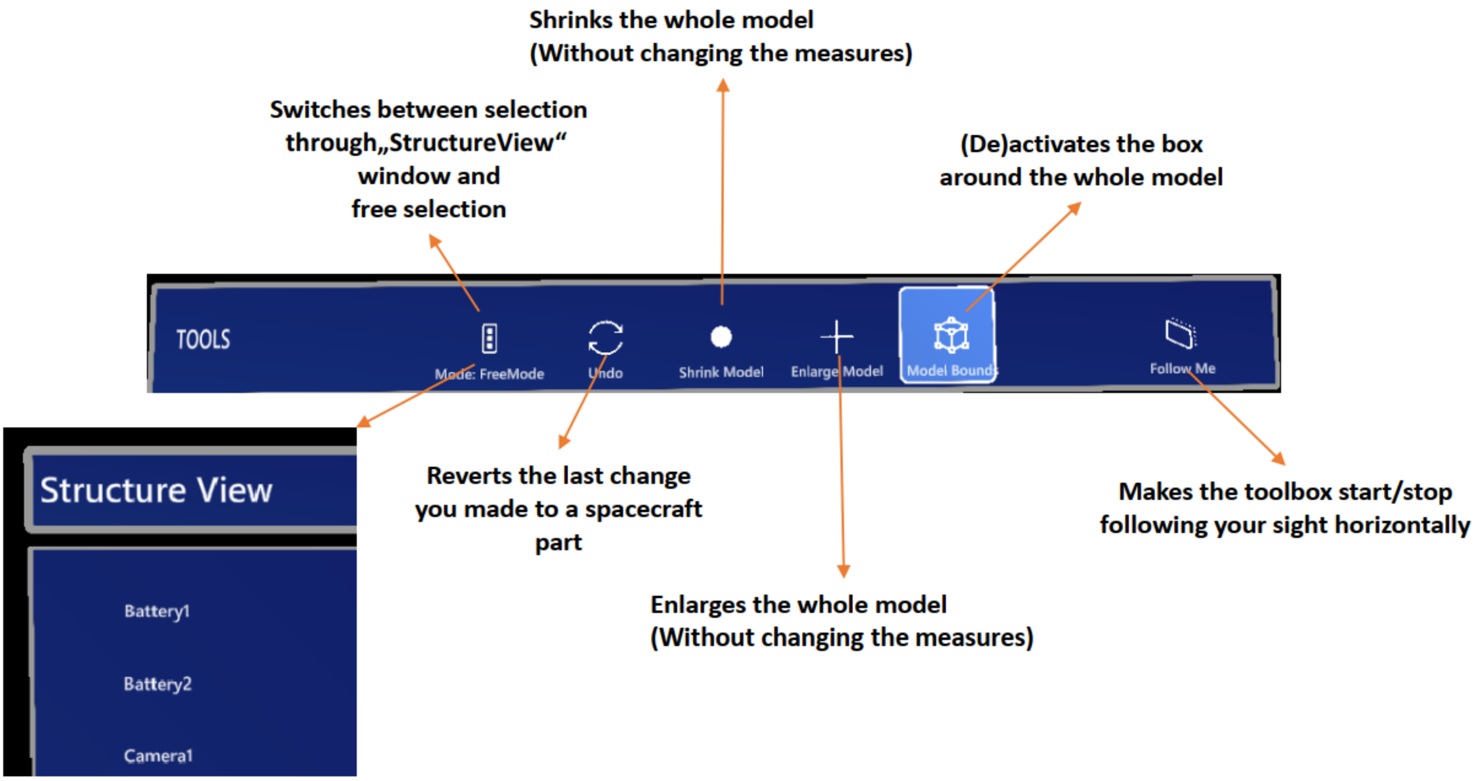
Radiator 1 & 2

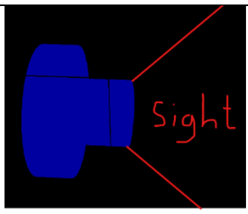
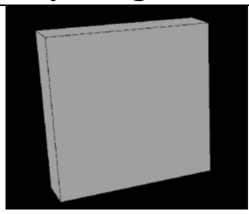
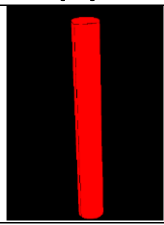
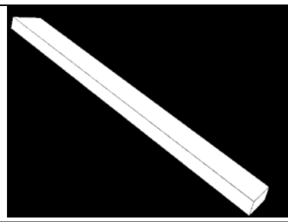
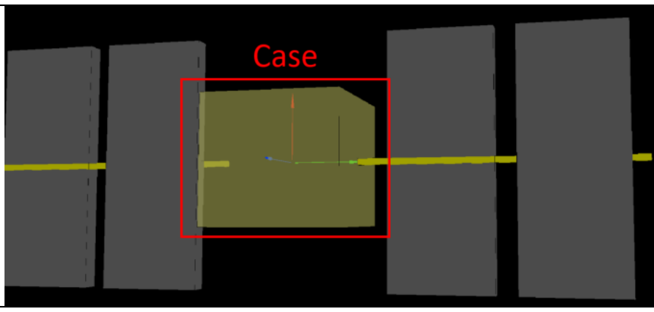
- Has to be inside & outside the *case*



Heat pipe

- Has to be placed inside the case
- Has to be as far away as possible from the *radiators* and the *computing unit*

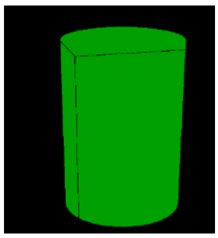


Camera 1 & 2	Computing unit	Heat pipe	Radiator 1 & 2
			
Satellite			
			

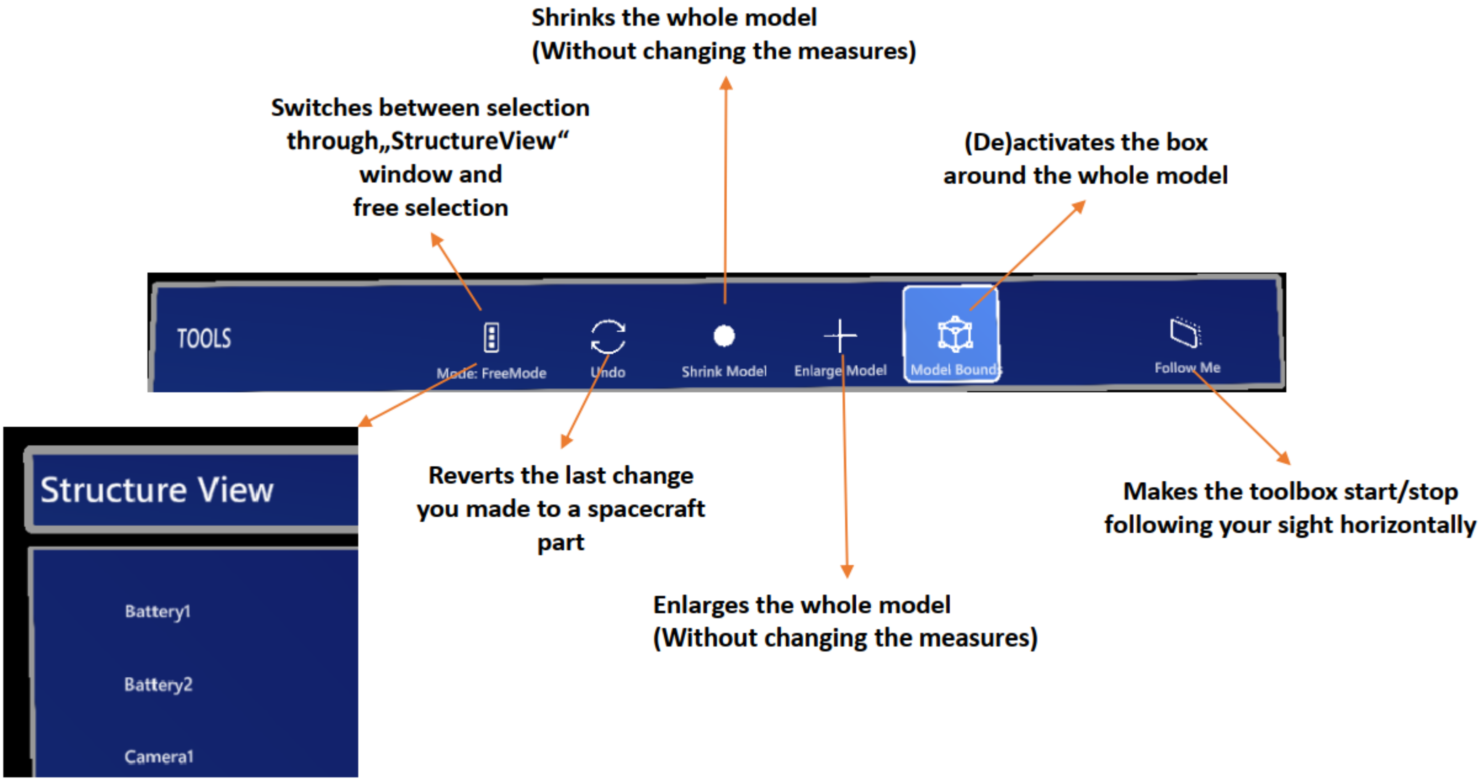
Your role:

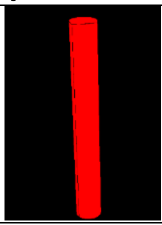
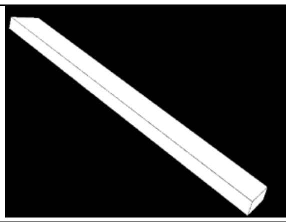
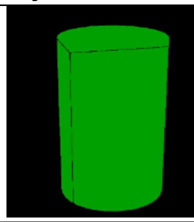
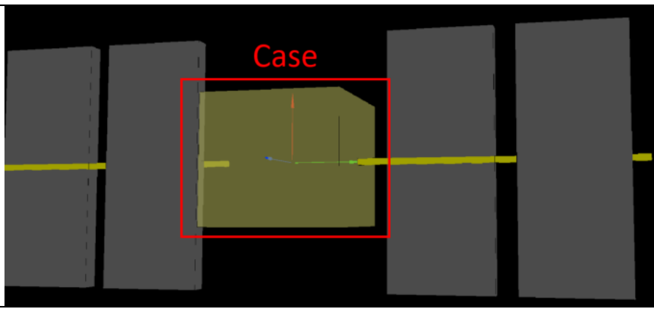
Expert for energy supply

Battery 1 & 2



- Has to be placed inside the *case*
- Has to be placed near a *camera*
- Must touch a *radiator*

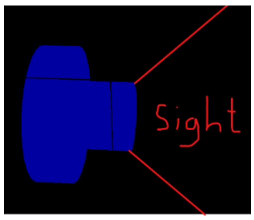


<table>
<tr><th>Heat pipe</th><th>Radiator 1 & 2</th><th>Battery 1 & 2</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><th colspan="3">Satellite</th></tr>
<tr><td colspan="3"></td></tr>
</table>

Your role:

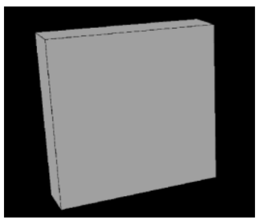
Expert for technical devices

Camera 1 & 2



- Has to be placed **outside** the *case*
- Has to have free sight

Computing Unit



- Has to be placed **inside** the *case*
- Has to touch both *radiators*

B. Questionnaire

Questionnaire

User Number	Age	Gender	System: **Collaborative Manipulation in an AR environment**

Experience

			1 2 3 4 5	
1	Do you have prior experience with *AR* or *VR*?	No experience	☐☐☐☐☐	Much experience
2	Do you have prior experience with *Microsoft HoloLens 1/2*?	No experience	☐☐☐☐☐	Much experience
3	Do you have prior experience with *Virtual Satellite*?	No experience	☐☐☐☐☐	Much experience
4	How often do you work with *Virtual Satellite*?	Never	☐☐☐☐☐	Daily

Task

			1 2 3 4 5	
5	I think that I would like to use this system frequently.	Strongly disagree	☐☐☐☐☐	Strongly agree
6	I found the *system* unnecessarily complex.	Strongly disagree	☐☐☐☐☐	Strongly agree
7	I thought the *system* was easy to use.	Strongly disagree	☐☐☐☐☐	Strongly agree
8	I think that I would need the support of a technical person to be able to use this *system*.	Strongly disagree	☐☐☐☐☐	Strongly agree
9	I found the various functions in this system were well integrated.	Strongly disagree	☐☐☐☐☐	Strongly agree
10	I thought there was too much inconsistency in this system.	Strongly disagree	☐☐☐☐☐	Strongly agree
11	I would imagine that most people would learn to use this *system* very quickly.	Strongly disagree	☐☐☐☐☐	Strongly agree
12	I found the *system* very cumbersome (awkward) to use.	Strongly disagree	☐☐☐☐☐	Strongly agree
13	I felt very confident using the system.	Strongly disagree	☐☐☐☐☐	Strongly agree
14	I needed to learn a lot of things before I could get going with this system.	Strongly disagree	☐☐☐☐☐	Strongly agree

Collaboration

			1 2 3 4 5	
15	I would imagine that the *system* can support the process in the CEF.	Strongly disagree	☐☐☐☐☐	Strongly agree
16	I would like to use such a *system* frequently in the CEF.	Strongly disagree	☐☐☐☐☐	Strongly agree
17	I found that communicating with other participants felt very natural during the task.	Strongly disagree	☐☐☐☐☐	Strongly agree

18	I found that collaborating with other participants using the *system* worked out very well during the task.	Strongly disagree	☐☐☐☐☐	Strongly agree
19	I found the *system* to be very frustrating	Strongly disagree	☐☐☐☐☐	Strongly agree
20	I think I understood the requirements of other disciplines.	Strongly disagree	☐☐☐☐☐	Strongly agree
21	I think that the system was intuitive to use.	Strongly disagree	☐☐☐☐☐	Strongly agree
22	I think that it was intuitive to place the parts during the task.	Strongly disagree	☐☐☐☐☐	Strongly agree
23	I think that the system can help in resolving conflicts with other disciplines in a CEF.	Strongly disagree	☐☐☐☐☐	Strongly agree

	Software		1 2 3 4 5	
24	I found the manipulation precise enough for the given task.	Strongly disagree	☐☐☐☐☐	Strongly agree
25	I used the "Mode" function often during the task *(showing structure-view/ free-view)*	Strongly disagree	☐☐☐☐☐	Strongly agree
26	I found the "Mode" function to be very helpful *(showing structure-view/ free-view)*	Strongly disagree	☐☐☐☐☐	Strongly agree
27	I used the "Undo" function often during the task (*Reverse your last object placement*)	Strongly disagree	☐☐☐☐☐	Strongly agree
28	I found the "Undo" function to be very helpful (*Reverse your last object placement*)	Strongly disagree	☐☐☐☐☐	Strongly agree
29	I found the function to move, rotate and resize the whole satellite to be very helpful	Strongly disagree	☐☐☐☐☐	Strongly agree

			5 25 50 75 100	
30	How mentally demanding was it to perform the task?	Very Low	|__________|	Very High
31	How physically demanding was it to perform the task?	Very Low	|__________|	Very High
32	How hurried or rushed was the pace of the task?	Very Low	|__________|	Very High
33	How successful were you in accomplishing what you were asked to do?	Perfect	|__________|	Failure
34	How hard did you have to work to accomplish your level of performance?	Very Low	|__________|	Very High
35	How insecure, discouraged, irritated, stressed and annoyed were you?	Very Low	|__________|	Very High

Additional comments: