BIM prove

D2.7 – User interface software and interface description

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement nº 958450. This document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

Project Title	Improving Building Information Modelling by Realtime Tracing of Construction Processes
Project Acronym	BIMprove
Grant Agreement No	958450
Instrument	Research & Innovation Action
Topic	Industrial Sustainability
Start Date of Project	1st September 2020
Duration of Project	36 Months

Name and Number of the deliverable	2.7 - User interface software and interface description
Related WP number and name	WP 2 - Components, technologies & functionalities
Deliverable dissemination level	Public
Deliverable due date	31 December 2022
Deliverable submission date	31 December 2022
Task leader/Main author	Jörg Frohnmayer (USTUTT)
Contributing partners	Matthias Aust (FhG-IAO), Alexandros Giannakidis (FhG-IAO), Dag Fjeld Edvardsen (CATENDA), Ruprecht Altenburger (ZHAW), Kaj Helin (VTT)
Reviewer(s)	Gabor Sziebig (SINTEF)

Abstract

This deliverable document the development of the user interfaces (UIs) to the BIMprove System. It briefly recapitulates the UIs' conceptualisation and their role in the BIMprove System. The implemented UIs are described and links to the respective software are provided.

Keywords

Software development, Virtual Reality, User Interfaces, UxV, Augmented Reality, IFC

Revisions

Version	Submission date	Comments	Author
v0.1	18.11.2022	Initial version	Jörg Frohnmayer (USTUTT), Matthias Aust (FhG-IAO)
v0.2	17.12.2022	Corrections	Alexandros Giannakidis (FhG-IAO), Dag Fjeld Edvardsen (CATENDA), Ruprecht Altenburger (ZHAW), Kaj Helin (VTT)
V1.0	30.12.2022	Approved, final version	Gabor Sziebig (SINTEF)

Disclaimer

Acronyms and definitions

Acronym	Meaning
AI	Artificial Intelligence
API	Application Protocol Interface
AR	Augmented Reality
BCF	BIM collaboration format, an open standard task-format with strong industrys supportindustry support
BIM	Building Information Model(ling)
CDE	Common data environment, a centralized digital platform for data storage and access in BIM projects.
DT	Digital Twin
FOV	field of view
GUI	Graphical user interface
HMD	Head mounted display: "Goggles" for VR/AR/XR
IFC	Industry Foundation Classes, the file format for open building information models
MUVR	Multi-User-VR
MUXR	Multi-User-XR
OS	Operating system
PC	Personal computer (Computer with a Microsoft Windows OS)
POV	Point of view
SDK	Software development kit
UAV	Unmanned aerial vehicle, e.g. drone
UI	User interface
UxV	Unmanned vehicle, e.g. drone or gound robot.
VR	Virtual Reality
XR	Extended Reality or Mixed Reality – either VR or AR or something in between

BIMprove project

In the past 20 years, productivity in the European construction industry has increased by 1% annually only, which is at the lower end compared to other industrial sectors. Consequently, the sector has to step up its digitization efforts significantly, on the one hand to increase its competitiveness and on the other hand to get rid of its image as dirty, dangerous and physical demanding working environment. Construction industry clearly needs to progress beyond Building Information Modelling when it comes to digitizing their processes in such a way that all stakeholders involved in the construction process can be involved.

The true potential of comprehensive digitization in construction can only be exploited if the current status of the construction work is digitally integrated in a common workflow. A Digital Twin provides construction companies with real-time data on the development of their assets, devices and products during creation and also enables predictions on workforce, material and costs.

BIMprove facilitates such a comprehensive end-to-end digital thread using autonomous tracking systems to continuously identify deviations and update the Digital Twin accordingly. In addition, locations of construction site personnel are tracked anonymously, so that **BIMprove** system services are able to optimize the allocation of resources, the flow of people and the safety of the employees. Information will be easily accessible for all user groups by providing personalized interfaces, such as wearable devices for alerts or VR visualizations for site managers. **BIMprove** is a cloud-based service-oriented system that has a multi-layered structure and enables extensions to be added at any time.

The main goals of **BIMprove** are a significant reduction in costs, better use of resources and fewer accidents on construction sites. By providing a complete digital workflow, BIMprove will help to sustainably improve the productivity and image of the European construction industry.

Contents

Index of Figures

1. Overview of BIMprove UIs and UI-concepts

1.1. The BIMprove input-processing-output-cycle

The cycle of data input, data processing, data output, user request for the next round of data acquisition an input, processing of that new data, and so on, has been described in the proposal of the BIMprove project and has since then been refined in Deliverable 1.1 Overall system design of BIMprove and Deliverable 2.8 Layer Services description (M18 update). The Figure 1. shows an abstraction of the BIMprove System which illustrates this cycle. The white boxes represent system modules (and their submodules), and the coloured boxes with the rounded corners represent functions.

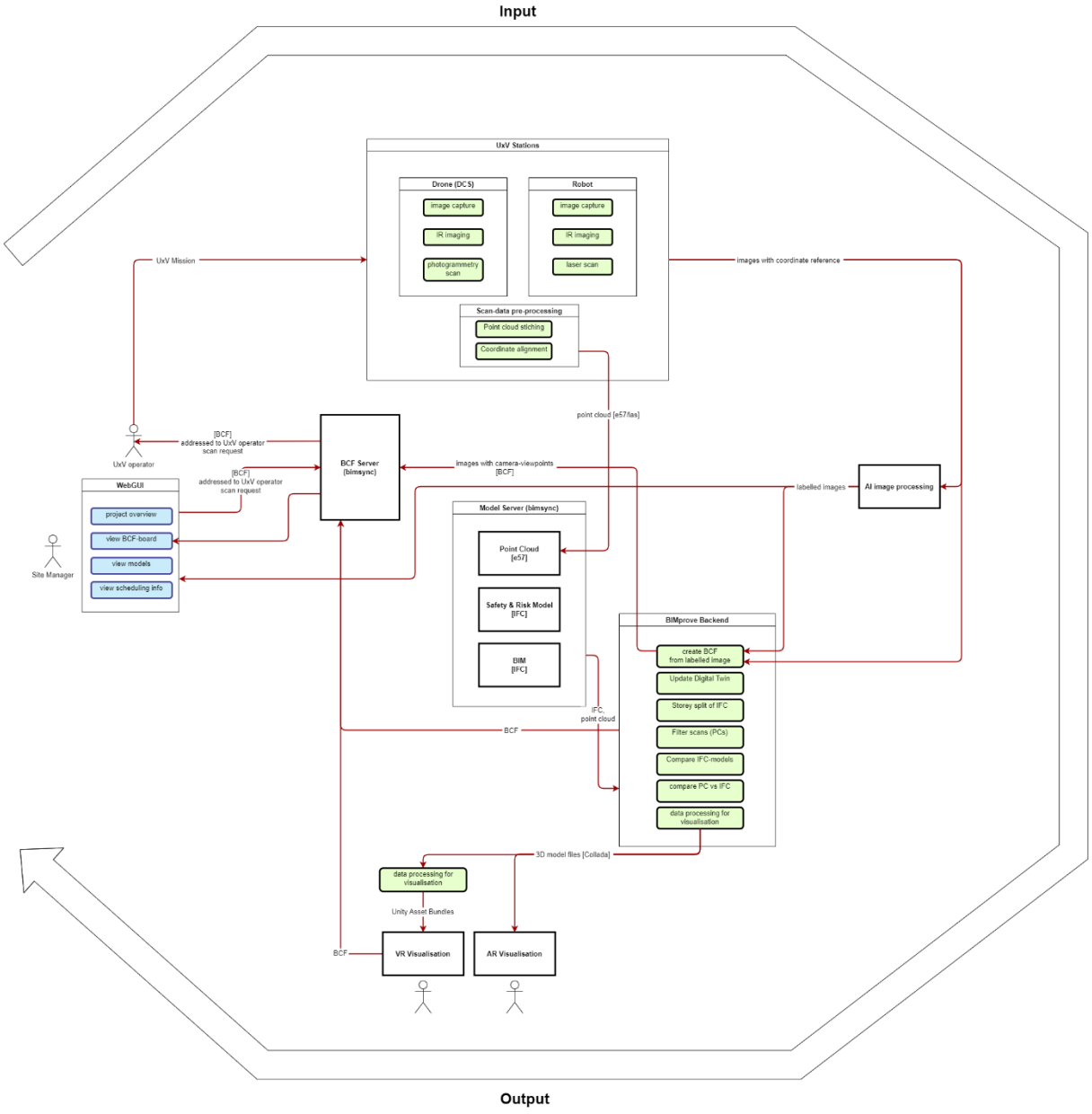

Figure 1 The cycle of BIMprove system

Input to the system is mainly provided through data acquisition by the UxVs, e.g. drones and ground robots and their sensors. (3D IFC BIM models and other information provided, for example, by planner engineers, can also be considered input to the system, of course.) The UxVs are represented in the top part of the above illustration. Their input, which consists of point clouds and images, is then processed. This processing is represented in the centre part of the illustration. Some pre-processing is needed before the data can be stored in the Model Server (which, for the BIMprove project itself, is Catenda's Bimsync). From the Model Server the data can be loaded to the backend to make use of its further processing functionality.

Output to the user is provided mainly by three types of user interfaces: the WebGUI and the VR and AR visualisation modules. These are represented on the left and bottom part of the illustration. These are also the UIs that are described and documented in more detail in the following sections of this deliverable.

What the figure also shows is the role the BIM Collaboration Format (BCF) plays in the BIMprove System - many of the red arrows are tagged with "BCF". This is described in further detail in the next section.

1.2. The role of the BCF standard for BIMprove user interaction

The BIM Collaboration Format (BCF) facilitates communication between stakeholders (e.g. planner engineers of different disciplines) in IFC-based BIM projects. It is a format developed by buildingSMART (buildingSMART International) and is also described in Deliverable 1.1 (Overall system design of BIMprove). Whenever details, issues, questions, or other information regarding IFC models is to be exchanged, this can be done via BCF. BCF is generally supported by BIM-CAD-software, so the 3D models and the communication about them can be interlinked. Affected building components, descriptions, responsible personnel, status, and due dates can all be specified within a BCF topic. This does not only ease communication between stakeholders in the planning phase of a building project. (See documentation of the BCF standard here: BCF-XML/Documentation) BCF topics can also be created automatically by software and are in turn also machine readable. One strength of the format is that it is designed to be easy to implement and also flexible. These factors make the BCF standard a valuable means for communication and user interfacing for the BIMprove system, and in turn a lot of the communication via the BIMprove system and from the BIMprove system is envisioned to work via BCFs.

The kinds of BCF topics envisioned for use within the BIMprove system include the following examples:

- "Normal" BCFs with construction issues, created by human stakeholders either via their BIM-software of choice, Bimsync, which serves a model- and BCF-server in BIMprove, or one of the BIMprove XR tools, namely the XR-Viewer or the BIM@Construction AR tool.
- Data request BCFs: These would hold information regarding what data should be obtained by an UxV. So, this could either specify a certain area of the construction site, maybe even a single BIM object, to be scanned either by a drone or by a ground robot, to obtain a point cloud of this specific object or number of objects. Or it could specify a path/trajectory for the UxV to take and scan along the way.
- BCF is used similarly for “high level mission requests”: When a set of tasks is scheduled to be completed on a certain day, a BCF task can be created based on this. Links in the system between tasks in the schedule and objects in the BIM models are used to create the BCF. It then contains references to these BIM objects (a core feature of BCF) and is assigned to the UxV operator to consider.
- BCF is being used to contain the results of mission planning. This is done by attaching a JSON file containing flight waypoints / scan points and such information. This is outside of the traditional use of BCF, but a big advantage is that this makes sure that this data is easily related to the high level mission requests (see above).
- A special case of a data request BCF could be to obtain IR thermal images of a certain area or object(s). For example, when welding work has occurred in a certain area of the construction site, it might make sense for the fire manager to request (via BCF) for this area to be monitored by a UxV with an IR camera (or for a high level mission request (see above) to be created from the scheduled welding task). These IR images would be uploaded to the BIMprove backend, processed, and if the detected temperature exceeds a certain threshold, another BCF, containing the respective IR images, would be addressed to the fire manager.
- BCF is being used to report the results of AI based detection of safety related features (or indicating their absence where they are expected).

So, even though the development of a special UI to work with the BCF format has not been a focus in the BIMprove project (although the WebGUI (see below) does include its own BCF board), the format is an important way for the user to interact with the system and for communication within the system itself.

1.3. User Interface Concept

1.3.1. Categorisation by mode of use

The construction Digital Twin Model has various layers and users with different needs and requests for information. There is no 'one-fit-for-all' GUI for all user roles and needs. BIMprove's general GUI concept can be seen in Figure 2. The GUI concept has four modes:

1. Digital Twin mode
2. Immersive Digital Twin mode
3. Mobile mode
4. Notification and Warnings mode

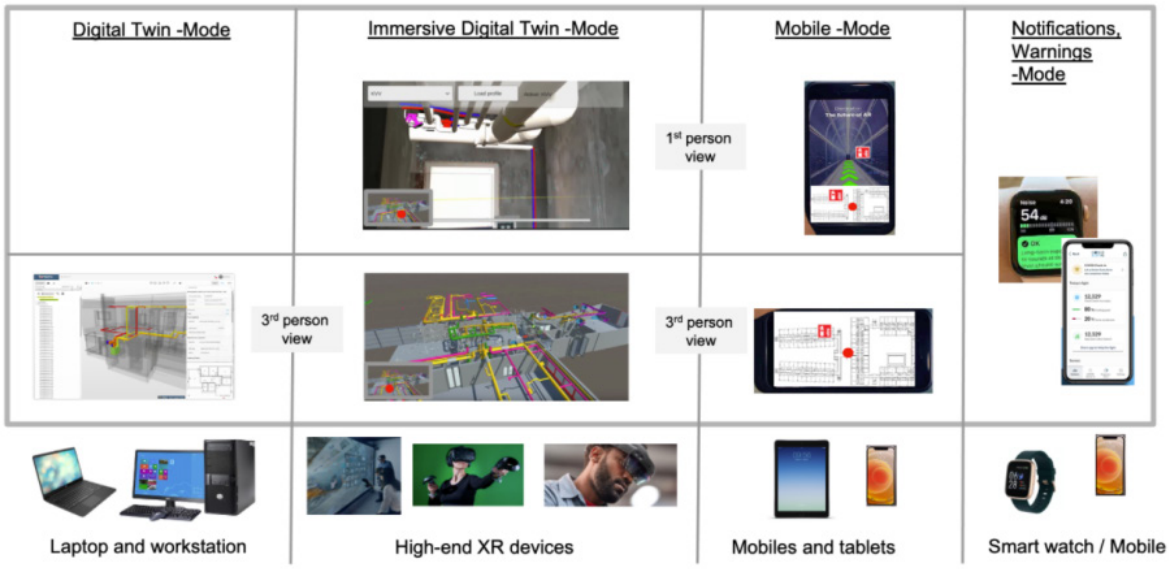

Figure 2 BIMprove's general-level UI concept, to be used in various devices.

The user is able to change modes based on device features and user-profile-based access rights. They can have the 1st person or 3rd person view in Immersive Digital Twin and Mobile modes. The Digital Twin mode allows the user to exploit only the 3rd person view. In the 1st person view, they will see information from the digital twin in real scale, i.e., can feel as if being located in a real room in Virtual Reality. In the 3rd person view, the user can see information from the digital twin in free distance, e.g., zoom in and out in a 2D map. There are mini maps on the 1st person or 3rd person views, which support the user's orientation in the BIMprove Digital Twin model and elicit better situation awareness. The most urgent and relevant information will be given users via Notification and Warning mode.

More information could be found in D1.3 "Concept descriptions of user interfaces and human-robot interaction" (Concept descriptions of user interfaces and human-robot interaction).

1.3.2. Categorisation by context of use

In the section above, the BIMprove user interfaces are categorised by ‘mode of use’ which considers hardware limitations, possible levels of immersion, and the type of viewing experience when inspecting 3D models (e.g. first vs. third person view). The BIM@ categories, which have been described in detail in Deliverable 1.3 (Concept descriptions of user interfaces and human-robot interaction) and in other BIMprove deliverables, categorise by ‘context of use’ to convey a more abstract but broader idea of the BIMprove user interface concept. The context of use, as defined in DIN EN ISO 9241-110 (https://www.iso.org/obp/ui/#iso:std:iso:9241:-110:ed-2:v1:en), is the combination of users (target audience), goals and tasks, resources, and environment; where “environment” includes the technical, physical, social, cultural and organizational environments. These different contexts are important to consider because in the construction industry the differences between them are quite substantial – whether you use a software tool in an office or on a construction site makes a difference.

The Figure 3 shows the six BIM@ categories and how these contexts of use are covered by the user interfaces which are described in the next section of this deliverable, e.g. the WebGUI, AR Visualisation and the XR Viewer.

The WebGUI has been developed as a responsive website. This means it works well on almost any device with a modern web browser, which makes it usable in almost any context. On the other hand, this also means it is not a specialised tool – it will work best on an office PC. Notice that there are no specialised implementations for BIM@Vehicle or BIM@Emergency. These are concepts for future considerations.

The XR Viewer, as described in section D2.7 User interface software and interface description | The XR Viewer: Multi user VR below, exists as a desktop version for use with a PC monitor, mouse, and keyboard, and as a VR version for use with head mounted displays (HMDs). So it can be used in any context where one of these hardware set-ups is available. This makes most sense in an office environment. In BIMprove we have suggested that a construction site office should be equipped with VR hardware – the BIM@SiteOffice concept. But, of course, the desktop version can be used in a site office, just like the VR version can be used in an “off-site-office”, e.g. a planner's office, if there is VR equipment available.

The augmentation of the real environment with virtual 3D models makes most sense on the construction site itself. Merging the 3D plan and the reality of the site is the augmented reality concept of BIM@Construction.

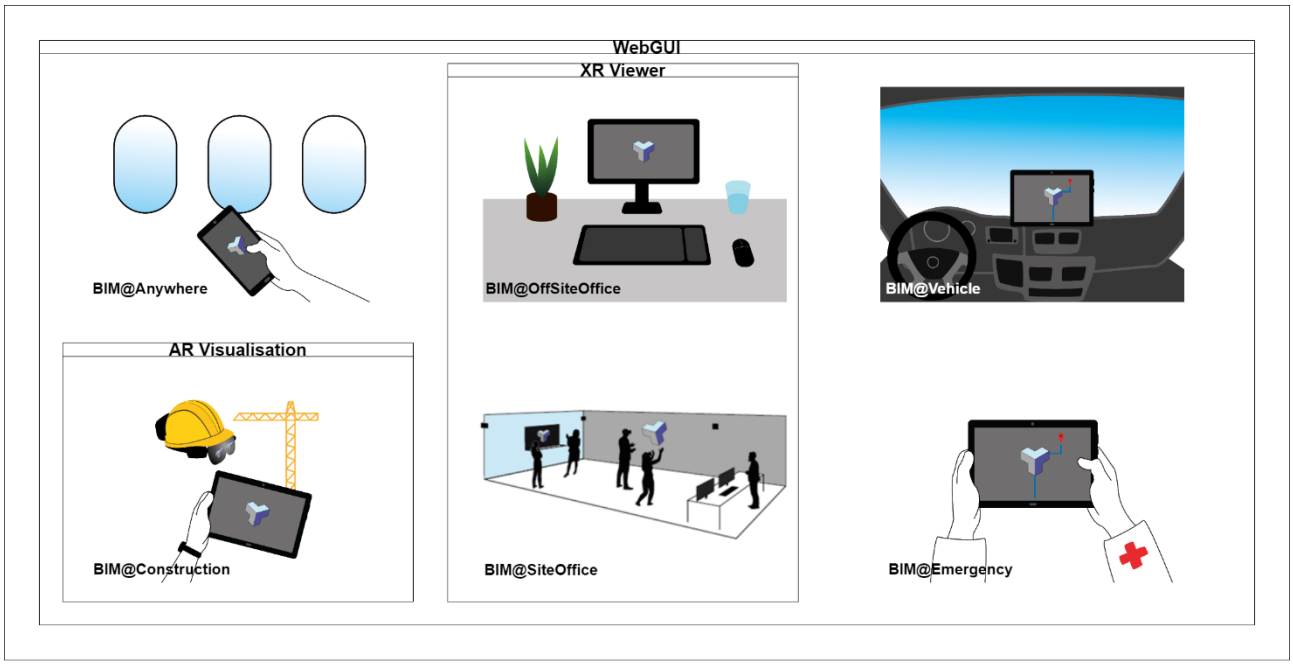

Figure 3 The BIM@ categories and how they are covered by the actual user interfaces

2. User Interfaces

Described below are the three UIs developed specifically for BIMprove: The WebGUI, the XR Viewer, and the BIM@Construction AR Tool. Links to actual software is provided in the respective subsections.

2.1. The WebGUI

As indicated above, the WebGUI offers an access point to a lot of the functionalities developed in the project. It consists of a website with a main project overview page and subpages for each project. It is developed using HTML, CSS and JavaScript utilizing the Bootstrap framework.

End user target audience are all authenticated users of the BIMprove System.

2.1.1. Inputs

The WebGUI acts as the frontend of the Backend Server similar to a MVC architecture and uses the FastAPI (FastAPI) Framework. Data is pulled and displayed from the Backend using Jinja2 Templates (Jinja2). This data includes the following, each of which is presented to the user on a separate page. While “Available Projects” are presented on a single overview page, there is one instance of each of the other pages per project.

- Available Projects
- Project Details
- Project Schedule
- BCF Issue Board
- Single Issues
- 3D Models of Buildings

While most of this is “read only” for the end user, there is the possibility to create, edit (assign to users, update description, attach screenshots) and remove BCF Issues (see also section The role of the BCF standard for BIMprove user interaction and section User Interaction by BCF).

2.1.2. File structure of the WebGUI

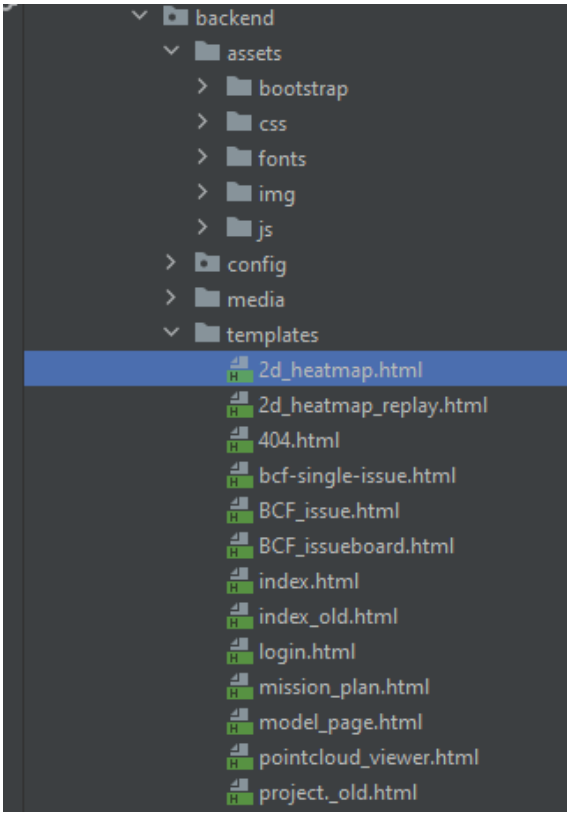

Figure 4 The file structure of html, css, javascript and some image files

```
<div class="d-flex flex-column" id="content-wrapper">
    <div id="content">
        <nav class="navbar navbar-light navbar-expand-lg navigation-clean">
            <div class="container"><img src="/assets/img/logobim.png">
                <div class="collapse navbar-collapse" id="navcol-1">
                    <ul class="navbar-nav ms-auto">
                        <li class="nav-item"><img src={{ gravatar }}></li>
                        <li class="nav-item dropdown"><a class="dropdown-toggle nav-link" aria-expanded="false" data-bs-toggle="dropdown" href="#">{{ username }}</a>
                            <div class="dropdown-menu"></div>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
        <div class="container-fluid">
            <div class="d-sm-flex justify-content-between align-items-center mb-4">
                <h3 class="text-dark mb-0">Overview of Project: {{bimsync_project_info.name}}</h3>
                <picture><img></picture>
            </div>
```

Figure 5 The HTML layer uses Jinja2 tags to include dynamic content (served mainly by the backend)

2.1.3. Software

The HTML, CSS and JavaScript templates for the main WebGUI is available from this git repository: https://bitbucket.org/dag_fjeld_edvardsen/web_gui/src/master/.

2.1.4. Screenshot examples

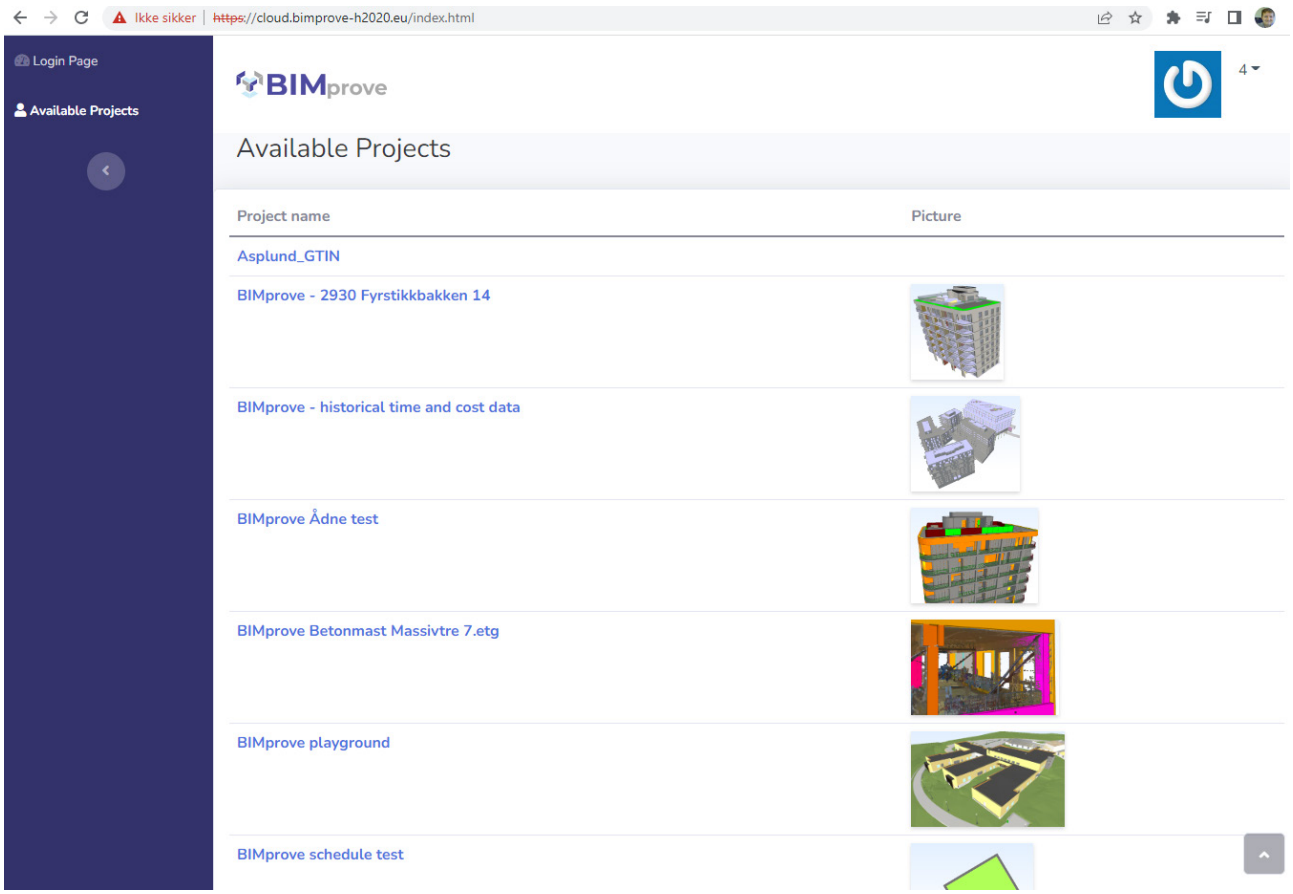

Figure 6 Main projects page of the WebGUI

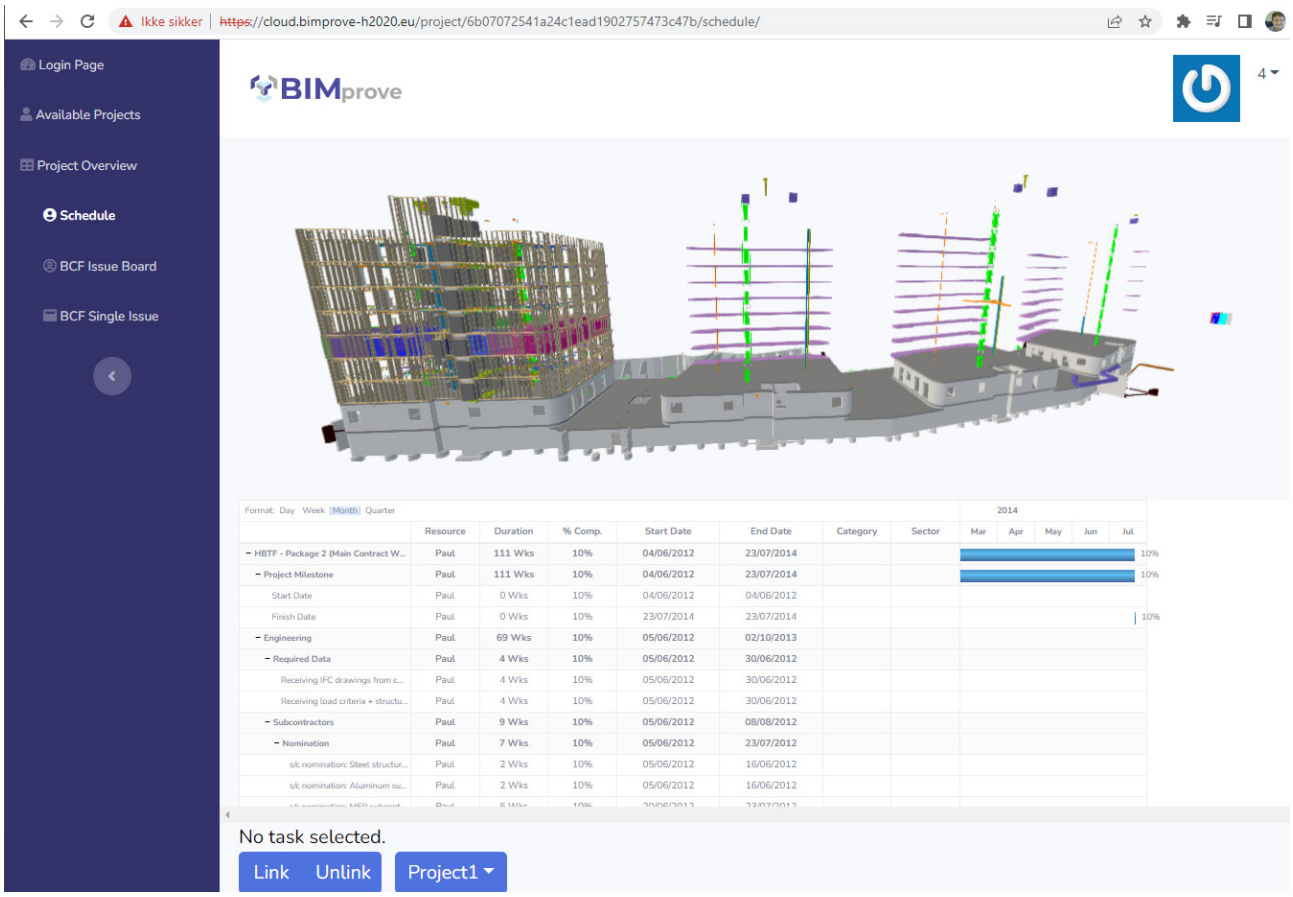

Figure 7 One of the project-specific subpages: The scheduling view

2.2. User Interaction by BCF

As described above (see section The role of the BCF standard for BIMprove user interaction) the BCF standard plays an important role in the user interaction of the BIMprove system. Below are some example screenshots of BCF topics. These screenshots are taken from Bimsync (https://bimsync.com/), which serves as the example CDE (Common Data Environment) in BIMprove. Any other BCF-capable BIM CDE could work in a similar way.

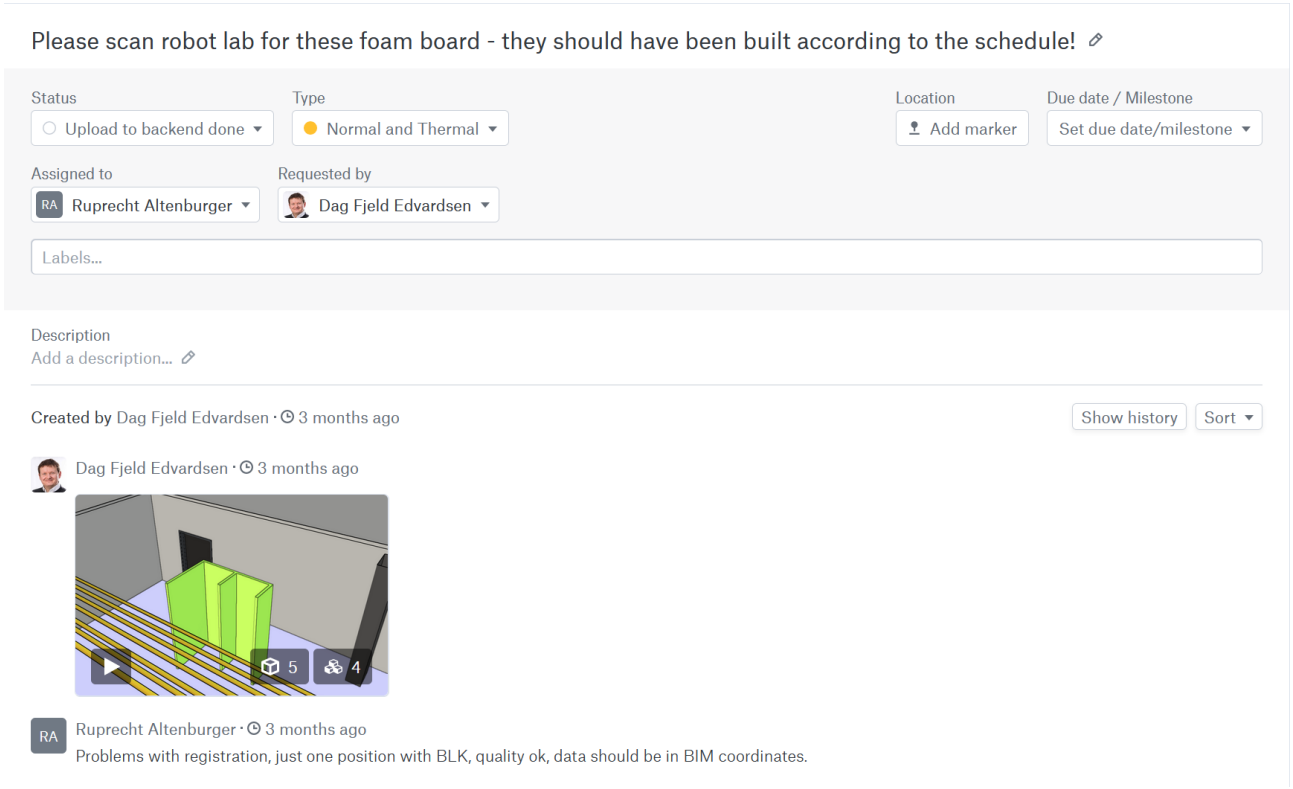

Figure 8 A high level mission request

The above BCF is forwarded to the UAV operator and can be further processed there. The planning of a drone mission is done graphically (clicking on waypoints) in a similar environment as before. Besides the pure position data, additional information can be given for orientation and data acquisition (for example: take an IR photo).

The corresponding BCF is still valid. With the mission planning a .json-file is generated and associated to it. This file contains the corresponding path data and additional information (see Figure 9).

```
[
  {
    "name": "waypoint",
    "x": 0.84,
    "y": 3.25,
    "z": 1.5,
    "angle": 0,
    "action": START,
    "velocity": 0.25
  },
  {
    "name": "waypoint",
    "x": 1.94,
    "y": 2.62,
    "z": 1.5,
    "angle": 0,
    "action": HOOVER,
    "velocity": 0.25
  },
  {
    "name": "waypoint",
    "x": 2.32,
    "y": 4.19,
    "z": 1.5,
    "angle": 0,
    "action": IMAGE_1HZ,
    "velocity": 0.25
  },
  {
    "name": "waypoint",
    "x": 3.13,
    "y": 0.9,
    "z": 1.5,
    "angle": 0,
    "action": FLIGHT,
    "velocity": 0.25
  },
  {
    "name": "waypoint",
    "x": 1.69,
    "y": -0.96,
    "z": 0.0,
    "angle": 0,
    "action": LAND,
    "velocity": 0.25
  }
]
```


Figure 9 .json file created and associated to BCF issue

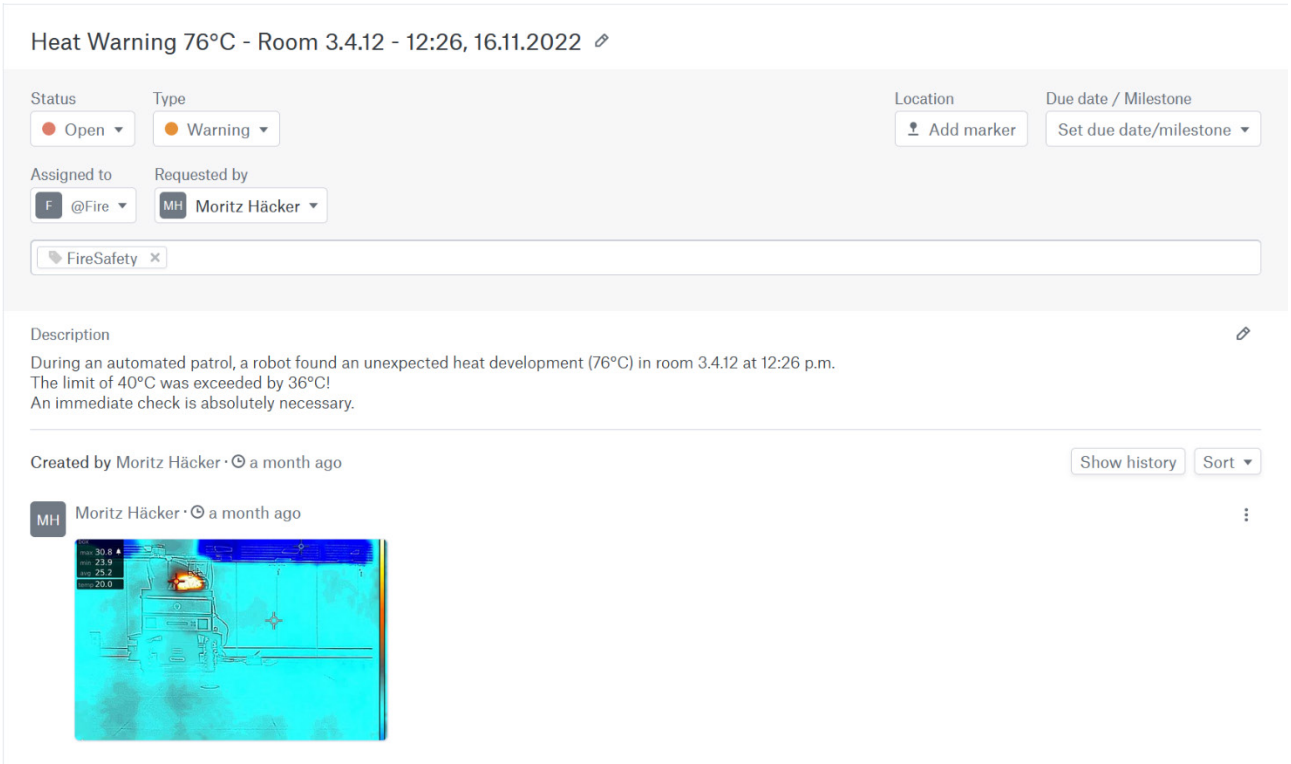

Figure 10 A BCF reporting about thermal image analysis

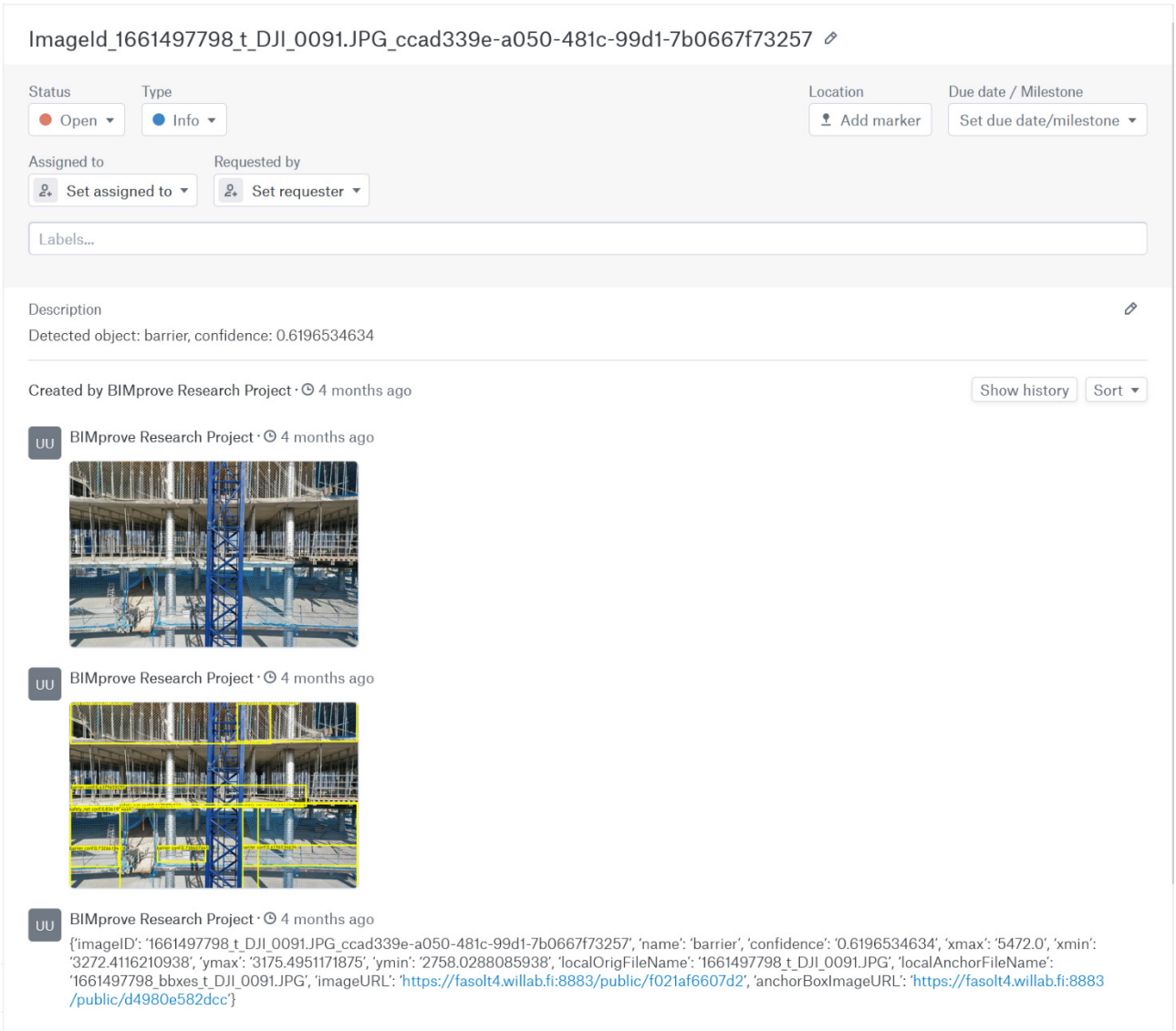

Figure 11 A preliminary version of a BCF topic, automatically created by the BIMprove system following automatic detection of safety measures

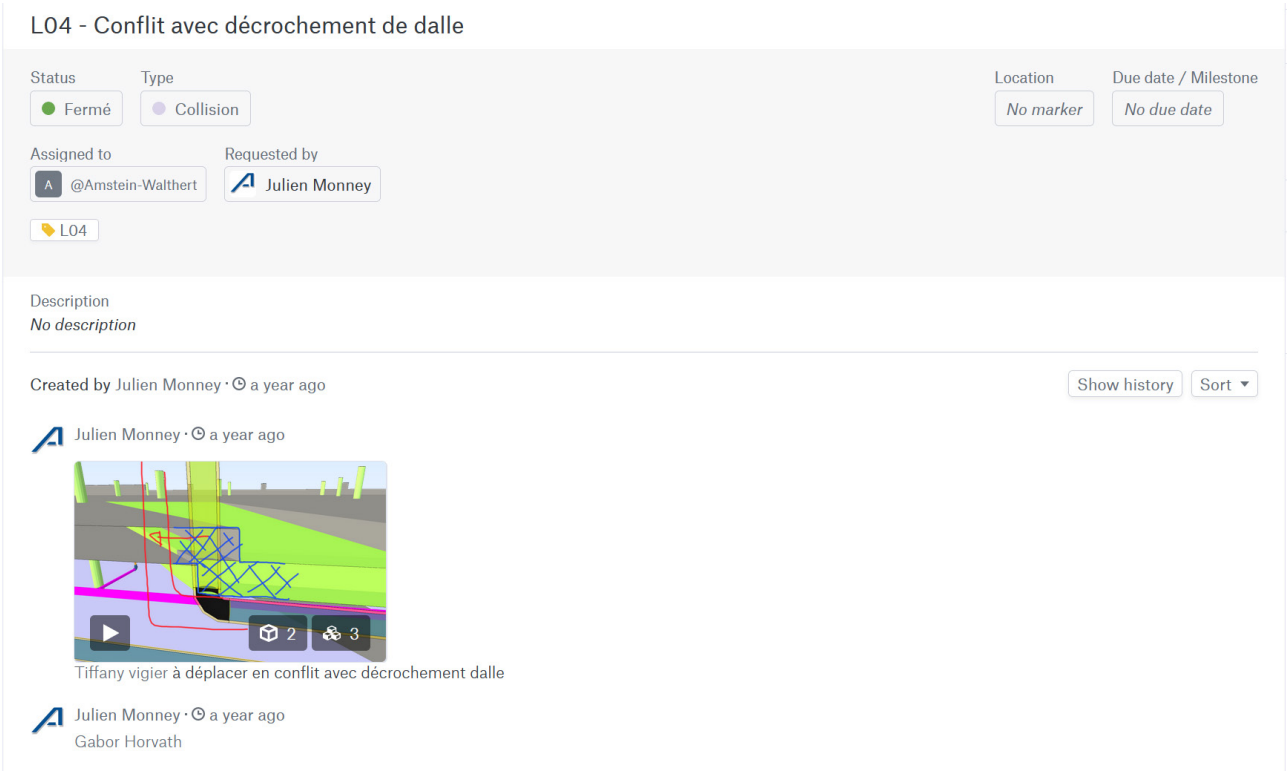

Figure 12 A "normal" BCF, reporting a construction problem, that consequently has been solved

2.3. The XR Viewer: Multi-user-VR

The XR Viewer a user interface software described in Deliverable 1.3. It is developed using the game development engine Unity (http://unity.com/). This makes it easy to deploy it to different hardware configurations. So far, two versions of the XR Viewer exist: One for multi-user-Virtual-Reality, and one for use on a Desktop PC. The latter of the two is described in the next section of this Deliverable, whereas this section is about the VR version. Both versions can be combined in the same multi-user-XR-session, with a number of users using VR head mounted displays and others using PCs. The multi-user functionality is facilitated using Normcore (Normcore. Seamless multiplayer for Unity, Normcore: High Quality Multiplayer Networking for Unity), a multiplayer networking plugin for Unity.

2.3.1. Inputs

The main input for the XR Viewer are the BIM models (IFC) of the respective construction project. For real-time visualisation in unity this data is converted to Collada (.dae). For this conversion the open-source-software IFCconvert is used. IFCconvert can be called from the command line and is therefore automatable – the conversion can also be provided by the Backend, so Collada files can be loaded directly from there.

Using IFCconvert the following command line will convert the FB_KJ_RIB.ifc file in to a Collada file

ifcConvert.exe --generate-uvs --orient-shells FB_KJ_RIB.ifc FB_KJ_RIB.dae

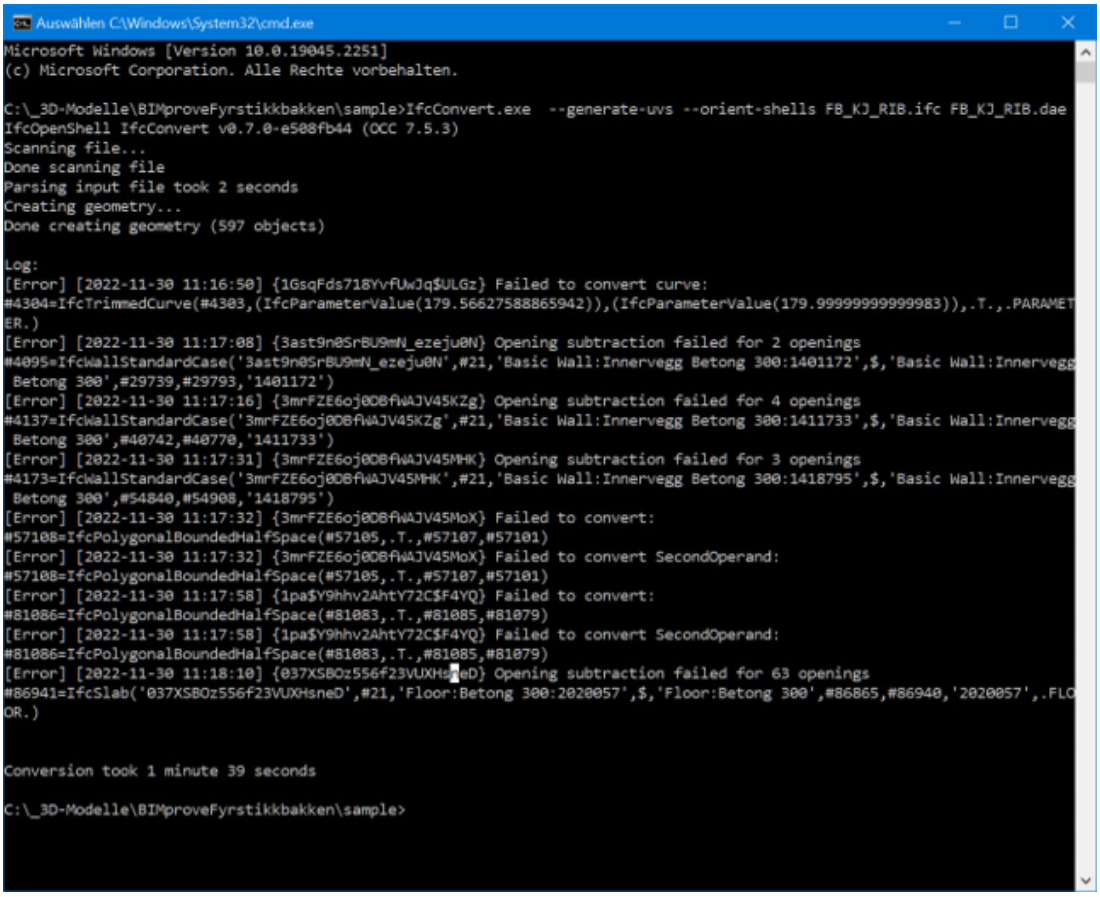

Figure 13 An example of applying IFCconvert

These Collada files can be loaded directly by the XR Viewer. For quicker loading (amongst other advantages) they can also be converted to Unity AssetBundles. AssetBundle is a binary format unique to the Unity game engine.

2.3.2. Software structure

The Figure 14 shown below illustrates the object structure of the XR-Viewer. Please note that the diagram is simplified, in that it does not show the differences between the VR and PC versions of the software. This is notable, for example, on the bottom left where the "User/Avatar"-object is illustrated - user interactions and differences in Avatar representation are simplified.

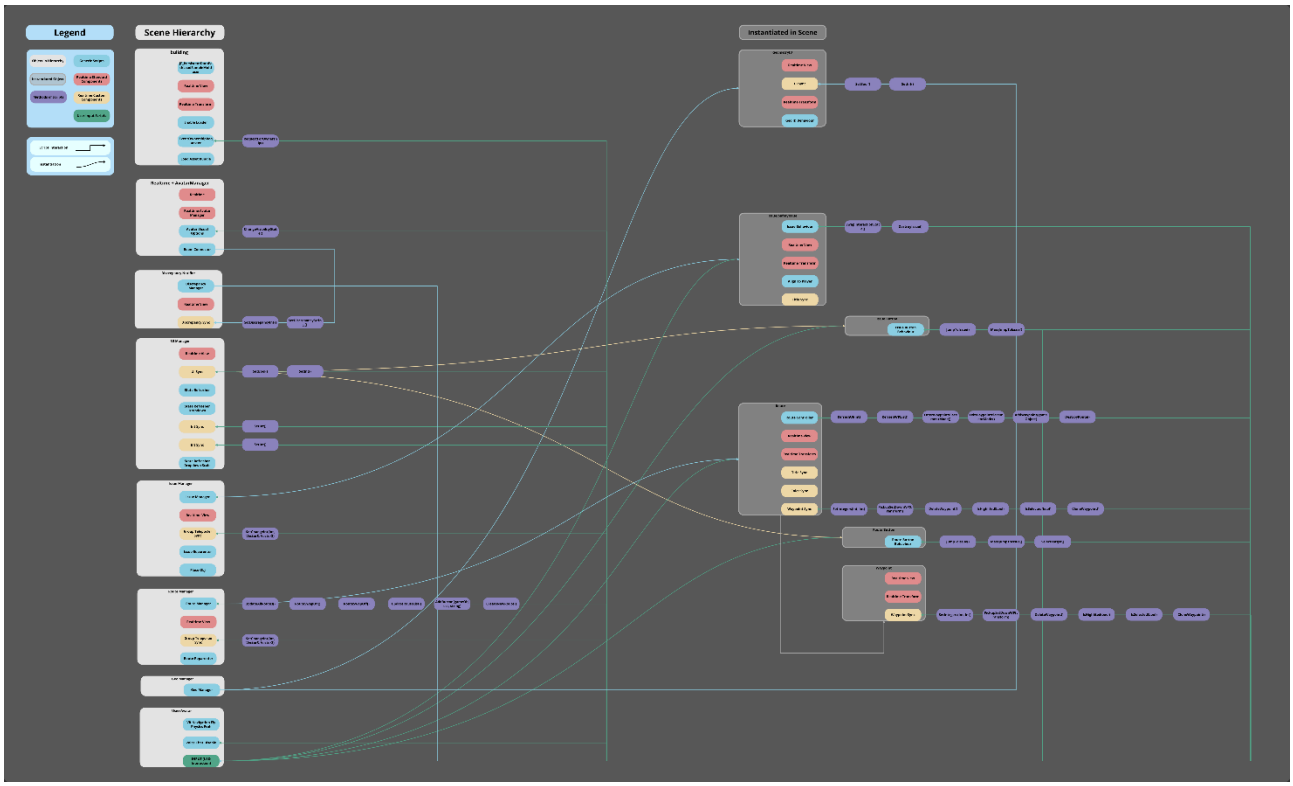

Figure 14 Unity-object structure

As shown in the diagram's legend on the top left of the illustration, all Unity-objects are either represented by white or grey boxes. White objects are already there when the software is started, whereas grey objects are instantiated at runtime. Each Unity-object can have several components, e.g. scripts (shown in blue or green) and Normcore-scripts that are synced for all users (red and yellow). Each script can communicate through methods (purple) with other components or other methods. Script interactions are represented by squared lines whereas instantiations of objects are represented by curved lines.

This is shown by the following use case example: A group of 10 people are investigating last days scan of the construction site. Lisa is checking the airflow inside the building and detects a difference between the as-planned BIM model and the as-built point cloud. She creates places a BCF-issue. For placing the issue, the object User/Avatar (white, bottom left) instantiates two objects from its INPUT (UI & interaction) (green) component. First, in the 3D world an Issue/SafetyIssue (grey, top centre) is created. It will be synced for all users by its real-time components (red and yellow). Second, inside the user menu a related Issue Button (grey) is created. These initializations are shown by the curved green lines from INPUT (UI & interaction) to Issue/SafetyIssue and to Issue Button. This newly created button will be synced for all users by the real-time components (red and yellow) of the UI Manager (white, centre left). To discuss the issue with her colleagues Lisa uses the just now created an "Issue Button" instance inside her user menu to group-summon all people around the BCF Issue. From the Ui input of Lisa the User/Avatar's Input (UI & interaction) component triggers

the Issue Button's Issue Button Behaviour (blue) component which executes the MassJumpToIssue() method (purple). This method communicates through the squared blue line with the Issue Manager's (white, centre left) Group Teleporter Sync (yellow) component and triggers its SetChangeInt() method. This method is distributed for all users since it's inside a Normcore custom component. Now there are 10 Avatars standing inside close quarters and the Issue is not really visible anymore. Lisa decides to turn all Avatars transparent. To do this, the object User/Avatar executes from its INPUT (UI & interaction) component through the squared green line the Realtime + AvatarManager's (white, top left), Avatar Visual Options (blue), ChangeVisiblityState() method. Therefore all Avatars for Lisa are set transparent and she can investigate the issue with everyone else close by.

2.3.3. Used SDKs and packages

The following SDKs, software packages etc. are used in the development of the XR Viewer:

- TextMeshPro 2.1.4
  (Included in Unity Editor but requires importing "TMP Essential Ressources", TextMesh Pro User Guide | TextMesh Pro | 1.4.1)
- Post Processing Stack 3.0.3
  Post Processing Stack v2 overview | Post Processing | 3.2.2
- Normcore 2.1.8
  Documentation | Normcore
- TriLib 1.9
  TriLib 1.9.0 – Download
- Steam VR Plugin 2.4.5
  SteamVR Plugin
  Quickstart | SteamVR Unity Plugin
- HTC Vive Input Utility 1.10.6
  Releases · ViveSoftware/ViveInputUtility-Unity
  VIVE Input Utility

2.3.4. XRvisualizer Asset and Release

Find the original XR Viewer (aka XRvisualizer) code here:

GitHub - bci-fhg-iao/bimprove-h2020-xrvisualizer: Unity 3D feature

Please note the ReadMe-file, there.

2.3.4.1 Outputs

How the XR Viewer can be applied has been documented well. The information can be obtained here:

- Deliverable 3.1 “Technology Demonstrations” D3.1 Technology Demonstrations (version 3, M25)
- YouTube: BIMprove H2020 Project
- BIMprove Pilot Use Case Lausanne November 2022 summary video
- BIMprove XR-Viewer: MulitiXR (Multi-User - Multi-Location - Multi-Device)
- Project partners gather in Madrid to visit the Spanish Pilot Use Case - BIMprove project
- Demo 12 Multi User Virtual Reality BIMprove XR Viewer
- Demo 2 Multi User Virtual Reality

2.3.4.2 User-inputs

As is documented in the sources listed above, the user can create “Issues” and “Routes” using the XR Viewer. These will be saved as BCF topics and sent back to the Backend. This is work in progress and not implemented in the current version, yet.

2.4. The XR Viewer: Desktop version

The XR Viewer is designed to be a multi-device software. Its multi-device purpose can be seen in the user interface, which is designed so that one can interact with 2D touch, mouse, as well as with 3D interactions. Therefore, the user interface is not as highly specialized on one specific device as it could be, but all functions are represented with very similar interactions throughout all devices. This will help users to access the XR Viewer from different devices and find user interactions they already know, as well as users helping each other inside a multi-user session even if they are on different devices.

The Desktop version of the XR Viewer is meant for a context of use (e.g. BIM@OffSiteOffice) where an HMD is either not available or not practical to use for other reasons. The software is the same as the VR version, except that the desktop navigation uses keyboard and mouse instead of an HMD and controllers.

2.5. BIM@Construction - AR Visualisation

The main idea behind the BIM@Construction concept application is to use BIM as digital twin data in immersive XR devices on the construction site. While AR devices like Microsoft HoloLens 2 are already built for interaction with 3D models (as shown in the figures below), the scale of those models is usually dictated by the FOV (Field of View) of the headset. In order to overlay digital twin information on top of the real building, additional pivot objects should be created. Naturally, pivot

object placement should match the placement of the fiducial markers in the real world. That way the digital twin and its real-world counterpart will be located in the same coordinate system. BIM@Construction includes several tools, which users can use in 1st person mode, like notes, measure tools, warning signs etc.

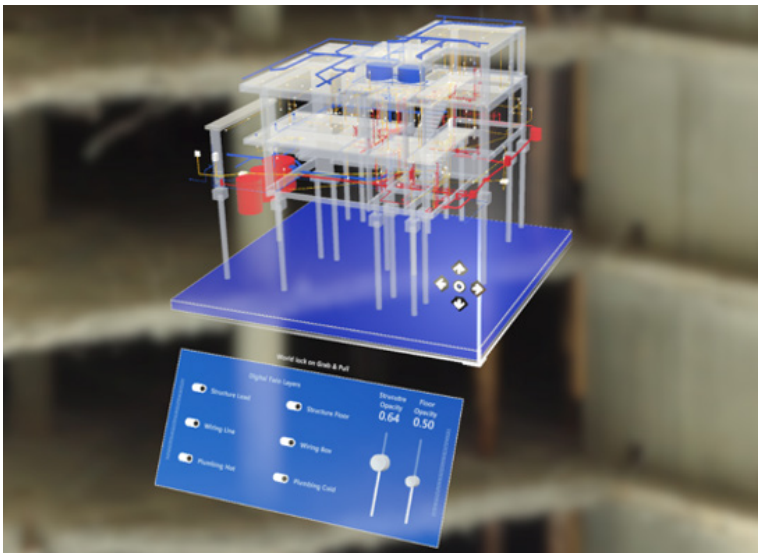

Figure 15 Immersive AR version of BIM@Construction: 3rd person view

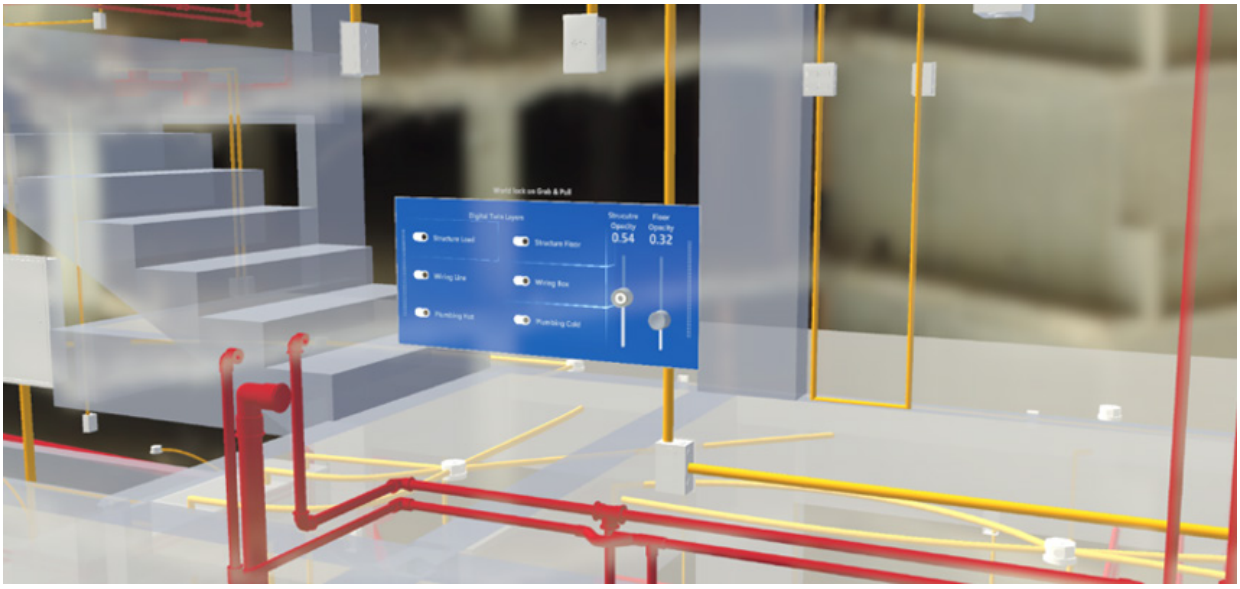

Figure 16 Immersive AR version of BIM@Construction: 1st person view

The mobile mode version of BIM@Construction concept application (see figures below) is using computer vision for feature and plane detection in order to place AR-based objects in the scene. It doesn't necessarily need fiducial markers but works well with them since they can be used for on-site tracking and contain additional image features (texture and contrast variation). In the same way as a HoloLens 2 application, the tablet version of the mobile mode provides both the 1st and the 3rd person views.

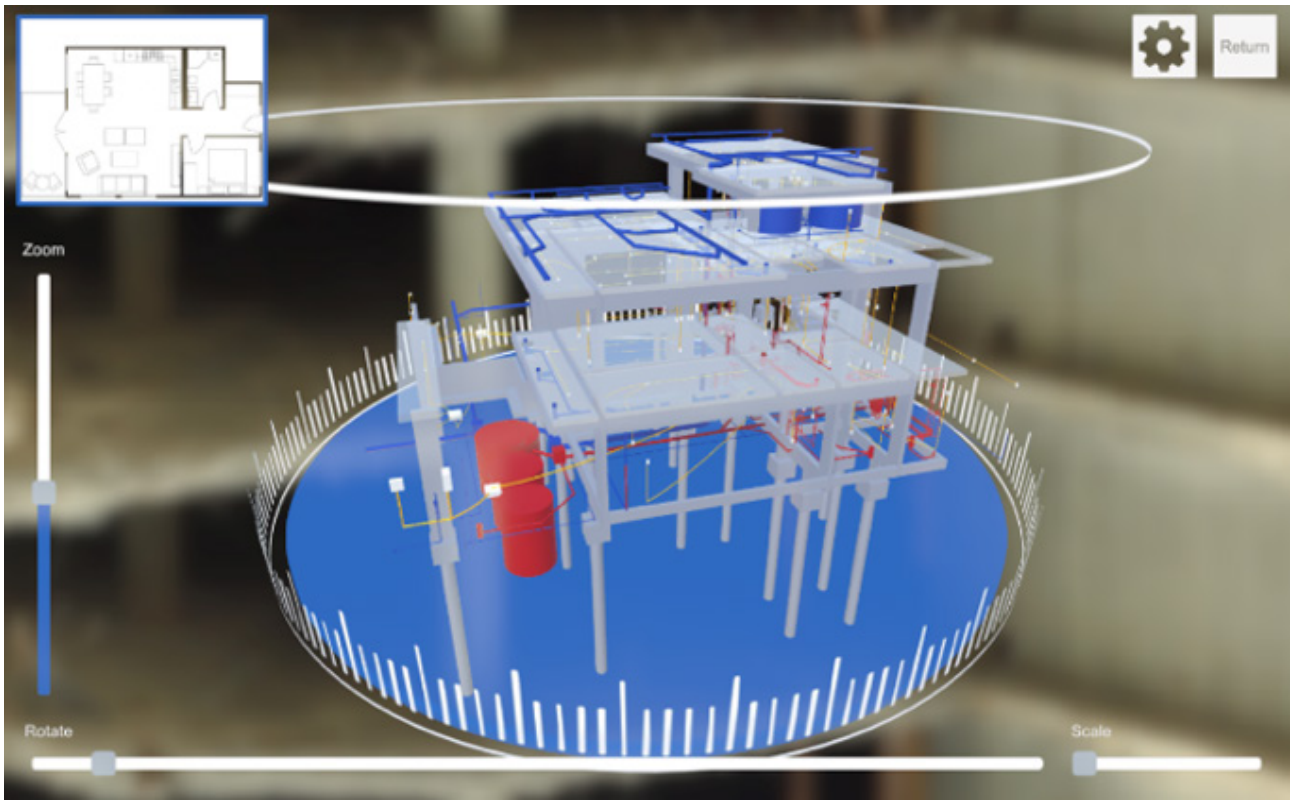

Figure 17 Mobile mode version of BIM@Construction: 3rd person view

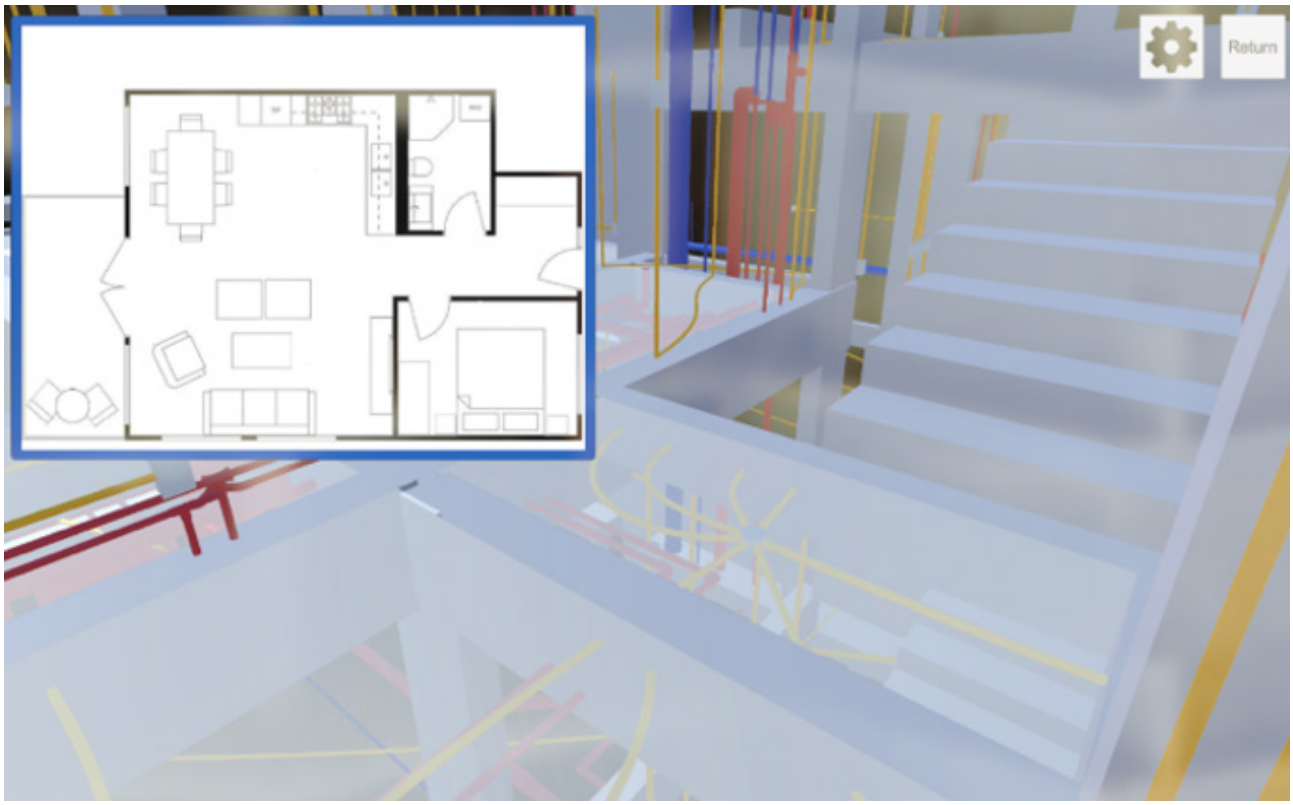

Figure 18 Mobile mode version of BIM@Construction: 1st person view

2.5.1. Inputs

The BIM@Construction concept application takes the following data as input:

- Digital Twin layers constructed from the BIM data.
- Picture of a fiducial marker used for tracking initialization.

For this application prototype, the Digital Twin layers are constructed manually for each new build and saved as Unity engine prefabs. To do so, first the BIM data is exported to one of the supported formats (.fbx, .dae, .3ds, .dxf, or .obj), and then grouped into suitable layers. Each layer is converted to a prefab and assigned to its own scriptable object, which in turn is referenced in the Digital Twin main control script. This process can be automated in the future.

Performance plays a huge role on AR devices. Use of BIM data can easily exceed available the polygon limit for scene geometry. On HoloLens 2 this limit is set to ~100 000 triangles, after which performance is going to be significantly reduced (see figure below). The prototype application testing shows that AR devices are suitable for displaying BIM data on room-by-room basis rather than loading entire building floors at once.

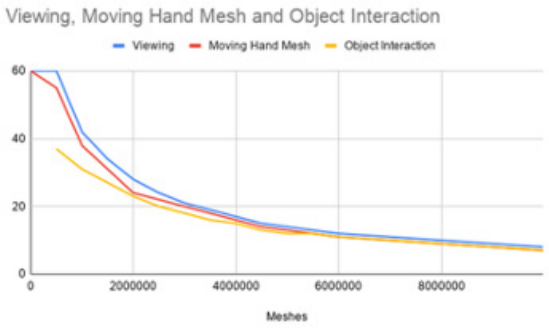

Figure 19 HoloLens 2 performance to polygon count

Images of fiducial markers can be in .jpg or .png formats with a maximum size of 2 megabytes. It is important to specify the real-world size of the marker in the Vuforia target manager and keep it the same in the Unity scene.

The following code shows the application's internal format for Digital Twin layer (see figure below):

```
public class DTLayerScriptableObject : ScriptableObject
{
    [Header("Digital Twin Layer parameters")]
    [SerializeField]
    public DTLayerNames layerName;

    [SerializeField]
    public Material layerMaterial;

    [SerializeField]
    public GameObject layerObject;
}
```

Figure 20 Digital Twin layer scriptable object structure

Each Digital Twin layer has its own Unity prefab and material (applied to every mesh during runtime) assigned and is referenced by its unique name.

2.5.2. Software structure

The BIM@Construction concept application is built by utilizing the Mixed Reality Toolkit (MRTK), Vuforia AR platform, and AR Foundation compatibility layer. The UI is created as a combination of MRTK, Unity UI, and TextMeshPro objects. In the figure below a simplified application architecture is presented:

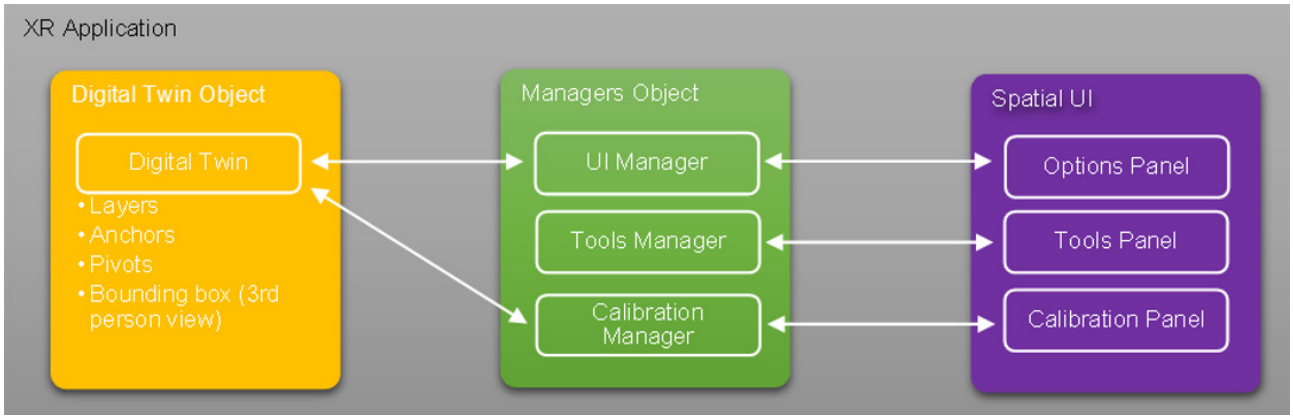

Figure 21 BIM@Construction concept application architecture

The Digital Twin object is used to store and control Digital Twin layers, their placement relative to the world origin, as well as transforms between immersive mode and third-person view. The position of the Digital Twin after tracking initialization with the marker can be adjusted with the help of a Calibration Manager. This manager controls the Calibration Panel UI and lets the user change position and rotation of the main pivoting point by selectable increment. The offset generated during this process is saved and will be applied automatically at the next application launch.

User input from the Options Panel is used by the UI Manager to select Digital Twin mode, layers visibility and their transparency level. The Tools Panel UI provides access to different tools (notes,

measuring ruler, etc.). Unlike the calibration offset, tools placement and Digital Twin configuration are not saved for the next session.

2.5.3. Outputs

The BIM@Construction application is used to display the BIM data on top of the real-world building on the construction site in immersive and third-person modes, and in its current form does not provide any outputs to the BIMprove Backend. However, it is possible to use built-in HoloLens 2 photo- and video-capture capabilities to record the application during runtime. For example, this will allow to identify issues, highlight them with the application's tools, and review them from the user's POV later.

2.5.4. User guide - HoloLens 2 application setup

2.5.4.1 App installation

First unzip the provided installation package (BIMprove_AR_1.1.0.0_ARM64_Test.rar). The root of the unzipped folder should contain a "BIMprove_AR_1.1.0.0_ARM64.appx" bundle file and there should be a subfolder "Dependencies/ARM64/" which contains an .appx file. These are the files used for installing the app in the Device Portal.

2.5.4.2 Installing the app

Start by clicking "Views" option from the Device Portal side bar. Next click on the "Apps" suboption in the side. Click on the "Choose file" button in the Deploy apps section and select the .appx file from the root of the unzipped app bundle folder. Now select the option "Allow me to select optional packages" and then click "Next. Click on the "Choose file" button and select the dependency .appx file from the "Dependencies/ARM64" folder. After the file is selected, click on the "Install" button to start the installation process. Once the installation is completed, simply click "Done" to finish the installation process. Leave the device portal open since it is required also in the following step.

2.5.5. User guide - Application Start-up sequence

2.5.5.1 Printing the marker

The application uses spatial anchors to keep the AR content in place. To create an anchor, an image target is used. First, open and print the "**Marker_PUC-Oslo_000_cropped.pdf**" file. The result should be a combination of a fiducial marker and a QR code on a white background. It should be in landscape format, filling the entire A4 page.

2.5.5.2 Starting the application

Launch the application from the HoloLens 2 "All apps" menu and allow the application to use the camera and the internet connection in the pop-up dialogue window (firs start only). After that, attach the printed marker to the building wall according to the instructions and look at it - after the marker

is recognized, the application will create the spatial anchor, and present the user with an anchor visualization placed directly on top of the marker. Press the “Apply” button to start the application.

2.5.6. User guide - Tablet application setup

2.5.6.1 Installing the app

From the “This PC” section, under “Devices and drives”, select and open you connected. Navigate to the desired folder on the tablet’s SD card and copy/paste the .apk file there. After that, open the tablet’s “My files” application and navigate to the folder containing the copied .apk file. Select the .apk file and press the “Install” button.

2.5.6.2 Starting the application

Launch the application, place the printed marker on the table and point the tablet camera on it. The application will ask the user to move the tablet slowly around the marker, so the camera can recognize its features. Recognized features are presented as small crosses. After that, tap the screen near one of those crosses to create an anchor, and press “Start” button.

2.5.6.3 Download app

https://vtt.sharefile.eu/d-sc13f18935bd542aa938c3aed114b8056

3. Summary

An extensive UI concept has been developed for the BIMprove System. Of this concept a web application and XR applications for four different types of end user devices have been implemented. These applications and their documentation have been provided within this deliverable.