D5.10
Visualisation Design Solutions

Editors:
Jazz Rasool
Mark Ransley
Carl Smith
Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Contributor(s)</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>04.04.2018</td>
<td>Mark Ransley (Ravensbourne)</td>
<td>Table of contents (document outline)</td>
</tr>
<tr>
<td>0.2</td>
<td>14.08.2018</td>
<td>Kaj Helin (VTT)</td>
<td>Added Real Time Sensor Panel Visualisation elements</td>
</tr>
<tr>
<td>0.3</td>
<td>17.08.2018</td>
<td>Jazz Rasool (Ravensbourne)</td>
<td>Added Recorder, Player and Post Analysis Sensor Visuals, Captioned Figures and Tables, Updated Styles and Table of Contents, Produced List of Figures and List of Tables.</td>
</tr>
<tr>
<td>0.4</td>
<td>18.08.2018</td>
<td>Alla Vovk (OBU)</td>
<td>Ergonomics of visualisation designs</td>
</tr>
<tr>
<td>0.5</td>
<td>20.08.2018</td>
<td>Carl Smith (Ravensbourne)</td>
<td>Internal review</td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>Paul Lefrere (CCA)</td>
<td>Peer review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roland Klemke (OUNL)</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>28.08.2018</td>
<td>Jazz Rasool, Carl Smith (Ravensbourne)</td>
<td>Edits after receiving Peer review. Deliverable requirements added to beginning of Executive summary. Minor grammatical edits. Design decisions to be expanded and edited.</td>
</tr>
<tr>
<td>0.8</td>
<td>29.08.2018</td>
<td>Jazz Rasool, Carl Smith (Ravensbourne)</td>
<td>Edits after receiving Peer review. Reconfigured WEKIT Recorder and Player Descriptions to cover design decisions for icons and panels.</td>
</tr>
<tr>
<td>0.9</td>
<td>31.08.2018</td>
<td>Mikhail Fominykh (EP)</td>
<td>Quality review, style and formatting</td>
</tr>
</tbody>
</table>

Disclaimer: All information included in this document is subject to change without notice. The Members of the WEKIT Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the WEKIT Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.
Visualisation Design Solutions

WP 5 | D5.10

Editors:
Jazz Rasool (Ravensbourne)
Mark Ransley (Ravensbourne)
Carl Smith (Ravensbourne)

Authors:
Jazz Rasool (Ravensbourne)
Mark Ransley (Ravensbourne)
Carl Smith (Ravensbourne)
Kaj Helin (VTT)
Alla Vovk (Oxford Brookes University)

Reviewers:
Paul Lefrere (CCA)
Roland Klemke (OUNL)
Mikhail Fominykh (EP)

Deliverable number: D5.10
Dissemination level: Public
Version: 0.9
Status: Final draft
Date: 31.08.2018
Due date: M33
Table of Contents

REVISION HISTORY.................................................................................................................. 2
LIST OF FIGURES .................................................................................................................... 5
LIST OF TABLES ....................................................................................................................... 5
EXECUTIVE SUMMARY ............................................................................................................ 6
1. INTRODUCTION ..................................................................................................................... 7
2. WEKIT FRAMEWORK ........................................................................................................... 7
    2.1 WEKIT RECORDER, INTERFACE VISUALS ....................................................................... 7
        2.1.1 Recorder Control Panel ............................................................................................. 7
        2.1.2 Model View Controller Design ................................................................................... 8
        2.1.3 Task Station and Annotations Menu: Design Overview .............................................. 8
        2.1.4 User Interface Design Principles ................................................................................ 10
    2.2 WEKIT PLAYER, DESIGN OF INTERFACE VISUALS ..................................................... 14
        2.2.1 WEKIT Player, Interface Visual Design ....................................................................... 15
    2.3 SENSORS, DATA AND STATISTICAL METHODS RECAP ........................................... 20
        2.3.1 Real-time component .................................................................................................. 20
        2.3.2 Post-task component .................................................................................................. 23
3. USER INTERFACE DESIGN METHODOLOGY ................................................................. 24
    3.1 REAL-TIME ASSISTANCE ................................................................................................ 24
4. CONCLUSION .......................................................................................................................... 25
    4.1 SEQUENTIAL NAVIGATION ............................................................................................. 25
    4.2 POSITIONING .................................................................................................................... 25
    4.3 INTERACTION FEEDBACK .................................................................................................. 26
    4.4 SPATIAL REPRESENTATION ............................................................................................... 26
REFERENCES .............................................................................................................................. 27
List of Figures

FIGURE 1: RECORDER CONTROL PANEL  
FIGURE 2: TASK STATION, A YELLOW 3D SPHERE ELEMENT  
FIGURE 3: ANNOTATION MENU ACTIVATED BY TAPPING TASK STATION SPHERE.  
FIGURE 4: ANCHOR CALIBRATION TOOL MARKER  
FIGURE 5: VUFORIA AUGMENTABLE TRIGGER QUALITY RATING FOR WEKIT TRIGGER IMAGE  
FIGURE 6: PLAYER STARTING SCREEN  
FIGURE 7: ACTIVITY OVERVIEW  
FIGURE 8: ACTIVITY CARD VIEW  
FIGURE 9: CONTENT VIEW  
FIGURE 10 CONTENT ACTIVATED  
FIGURE 11 Toggling from the Card view into the Task list view to Mark a step as complete  
FIGURE 12 SENSOR PANEL  
FIGURE 13 LAST STEP VIEW  
FIGURE 14: GROVE HEART SENSOR: ARDUINO SERIAL PLOTTER OUTPUT  
FIGURE 15: LIVE VISUALISATION OF SENSORS VIA A LOCAL LAPTOP MQTT SERVER/WEBPAGE FOR ENSURING SENSOR FUNCTIONALITY  
FIGURE 16: REAL-TIME ANALYSIS UI IN WEKIT PLAYER  
FIGURE 17: ONE OF THE HUMAN SENSOR VARIABLES (TEMPERATURE) WITHIN THE RED LIMITS  
FIGURE 18: ONE OF THE ENVIRONMENT SENSOR VARIABLES (HUMIDITY) WITHIN THE YELLOW LIMITS  
FIGURE 19: USE OF SUPPORT VECTOR MACHINE (SVM) ALGORITHMS LEARNING TO ESTABLISH IF A CANDIDATE’S SENSOR READINGS (FOR EXAMPLE, X1=PULSE AND X2=GSR) ARE WITHIN OPTIMAL RANGES  
FIGURE 20: AFFECTIVE STATE AS A CORRELATION BETWEEN PULSE AND TEMPERATURE

List of Tables

TABLE 3: ANNOTATION ICONS  
TABLE 4: ICONS FOR ADDING CONTENT  
TABLE 5: IMAGE ANNOTATION ICONS  
TABLE 6: TEXT ANNOTATION ICONS  
TABLE 7: GLYPH ANNOTATION ICONS
Executive summary

Relation to WEKIT Objectives

This deliverable focused on creating Visualization design solutions, including a visualization method to measure the performance and well-being of the user as well as design solutions for creating visual elements for interaction such as menus, icons and models. These are the outcomes for the ergonomics methodology established in WP4. Challenges of effective workflow including information overload will be addressed by establishing optimum attention engagement of design solutions. Most effective visual techniques are implemented based on perceptual visual methods and tests. Well-being and ergonomics factors will be tested through user test cycles. Final design solutions will be released for implementation. Conclusive Wearability and Ergonomics toolkit will be developed with easy access interactive publication. Findings will be transferred and organised into group of tools and methods to be downloaded (lead participant RAV, contributors: VTT).

After producing the design outcomes illustrated in WEKIT D5.4, progress was made to generate three types of visualisation materials:

1. Interface elements that include application icons, menus and dashboards
2. Task components such as lists of active task steps, models, images, animations and text
3. Data components such as data charts for data from sensors and data guidance from analysis of sensor signals related to affective state.

Methods

Interface and task components were mostly designed and implemented by using 3D illustration software by Oxford Brookes University whereas Data components were designed and implemented by VTT and Ravensbourne. VTT designed the Sensor Panel on the Hololens display dashboard showing live metrics for Pulse, Heart Rate Variability, Temperature and Humidity. Ravensbourne constructed routines to extract required metrics from the raw signals from sensors that could be fed to the Hololens Sensor Panel for live display and recorded and archived to a Cloud based repository for Post Analysis. Post Analysis visualisation included creating graphics showing if a user's clusters of metrics were falling within optimal performance ranges as well as what affective state could be derived from the clusters of metrics using Machine Learning algorithms.

Results

All Interface and task related visual elements and components were imported into Unity 3D and incorporated into Scenario task scenes through the WEKIT Recorder. The items were also able to be integrated in an alternative process where they were added as part of a data file and in folders for the WEKIT Player and then integrated into a playable task and sequence of steps using JSON formatted structure.
1. Introduction

In designing visualisation elements for the Hololens application and post analysis visuals, consideration was given to participatory design discussions held at a workshop at Ravensbourne and outlined in WP4 for outcomes. Many of the schematic ideas for icons, menus and task workflows developed were used as a basis for the design elements finally chosen. Factors of human perception, visual ergonomics and existing design standards for typical icons were considered and implemented. The design choices for elements took into consideration that they would be used in a 3D workspace. As a result, many elements were made as 3D models while 2D interfaces were used for tracking progress either of task steps or data streams from sensors or analysis.

2. WEKIT Framework

The Recorder interfaces had to have options for creating task stations, entering text, selecting and placing models, recording voice instructions, taking photographs of environments and placing them in a scene as well as initiating tracking of movement through Ghost recordings of the user’s actions. The Player needed the user to follow the pre-recorded task list and interact with it to activate the next segment of training as it had fewer specific requirements for interaction and interface design.

2.1 WEKIT Recorder, Interface Visuals

2.1.1 Recorder Control Panel

On starting the WEKIT application on Hololens, the Recorder Control Panel is displayed. All control panels that access the workflow of steps in a task sequence for a given scenario were designed in a flat panel, rectangular format similar to those likely to be found on mobile applications and tiling designs seen in current versions of Microsoft Windows associated with Hololens applications. A 2D geometry was selected because at this stage no 3D interactions such as placing a scene model or annotation are required. This panel must include design elements to provide functions for when the session recording of an expert’s action is complete (Save Recording) as well as provide an entry point into the main Menu for the WEKIT Player (Main Menu). Initialization checks for sensors such as the Myo are placed at the bottom of the panel for validating that sensors are ready to gather data for the session (Fig. 1). Green is a colour often used, as in traffic lights, to indicate that something is ready to be proceeded with so the when each of the Myo armbands is connected, the respective squares (Fig 1. bottom left and bottom right) will turn green.

Figure 1. Recorder Control Panel

WEKIT consortium Dissemination: Public Page 7/28
In addition to designing the visual elements and data visualisation objects, there need to be clear workflow steps that follow naturally from use of the icons and interfaces. The designers used two strategies to link visual elements to workflow models and underlying controls, specifically the 4 Components of Instructional Design approach (4C/ID) [1] and the Model-View Controller (MVC) approach.

**Instructional Design approach (4C/ID)**

The basic assumption of the 4C/ID model is that all complex learning can be represented in a combination of four components described by the model, namely:

1. **Learning task**: Learning tasks are experiences that are provided to promote help candidates build a context for the task, through observing or imitating the expert.
2. **Supportive information**: As candidates develop their mental models for a task, Supportive information cultivates deeper processing of the new information.
3. **Just in time information**: By providing the candidate information just in time for an action the candidate requires less preparation for the subsequent task which facilitates automation of the task.
4. **Part-task practice**: Some parts of the task are automatic, and they recur. Candidates must practice often to carry out steps automatically.

**2.1.2. Model View Controller Design**

In the Model-View-Controller approach [2] the Model defines format of the data that will be used in carrying out a function of an element. For WEKIT, we are using the ARLEM data format standard. Examples include data format linked to ‘Take a Photo’ or ‘Start Recording’. The View component decides how the function will be visualised, displayed and formatted as a Menu panel, Icon or visual interactive scene elements will be present. Examples include the Icons to ‘Take a Photo’ or to ‘Start recording’ a session’s steps into a session’s data archive on the Hololens. The Controller component focuses on the algorithm or computer programming code that will actually carry out the interaction or function. This includes things as commands to access the camera for taking a photo or connect to the recorder code routine infrastructure to start logging recorded steps.

**2.1.3. Task Station and Annotations Menu: Design Overview**

An expert records a session by adding a number of Task Stations, each with its own interactions or Annotations.

**Task Stations**

A general location fixed in the Environment space for a step in a task procedure which is being recorded by the Expert is known as a Task Station (TS). Through the Hololens, TSs are created using a double air tap while the gaze pointer is hitting the spatial map of the local environment. The TS is represented as a simple point in the form of an interactive yellow sphere. In constructing graphic elements in illustration or paint software, ‘anchors’ or ‘handles’ are used in the form of 2D circles or squares. As we are in a 3D space, the circle was replaced with its 3D equivalent, a sphere (Fig. 2). Although fixed in the Environment space within the Player, in the recorder the TS sphere can be moved or deleted. The TS is a general location close to where the step will take place. During each step, the expert adds annotation to their environment, which will later be played back (in the order they were created) to a trainee. A TS can contain any number of annotations. To go back and edit a TS, you must first tap on it to activate it (and you will see all of your previous annotations then...
displayed) and then you can tap on it again to open its menu. Deleting a TS will delete all the annotations associated with it.

![Task Station, A yellow 3D sphere element](image)

**Figure 2.** Task Station, A yellow 3D sphere element

Once created, air-tapping a TS will open the menu for adding annotations. Unlike one-dimensional linear dashboards or rows of icons, a 3D working space had activity or annotation icons laid out on a circle around the TS space to maintain focus toward the TS (Fig. 3).

![Annotation Menu activated by tapping Task Station sphere.](image)

**Figure 3.** Annotation Menu activated by tapping Task Station sphere.

Tapping a menu item will create a copy of the object, though distinguishable by a ring surrounding it (menu item = no ring). Each annotation has its own menu that will manage the content stored in each. TSs and annotations also have configuration menus (containing object-level functionality, such as delete and copy), which will have their functionality set in the top-level manager objects (in Unity). When a TS is created, it is the active TS. To help the user, only annotations for the active TS are visually displayed. To go back and edit a TS, you must first tap on it to activate it (and you will see all of your previous annotations then displayed) and then you can tap on it again to open its menu.

**Annotations**

A number of different types of annotation exist (see next section for details on each one). The user should first create an annotation of a certain type and then add content to it. The annotations can be dragged around (see section 2.1) and are used to precisely locate parts or position relevant to a specific action.

To record an expert’s instructions certain visualisation elements were designed for use in Unity to provide session references:
- **World Origin Marker Prefab** - the Unity visual marker for where in the 3D space of the Environment the person started the app
- **Task Station Prefab** - the Unity yellow sphere visual element to use for all task stations
- **Debug Text** - this is a text object found in the Hololens display. For development, it was used to display any information that should be user-locked.

With the Unity design environment, The Task Station element has code for:

- **Gaze focus as well as magnification** - sets the visual colour change and size when the gaze pointer is hovering over the task station.
- **Menu Animation Settings** - this sets the menu visual opening speed and the visual distance of the menu items (radially) from the centre object.

### 2.1.4. User Interface Design Principles

#### Gestures

Creating an annotation requires the gaze pointer to be on top of the object the person wishes to interact with, the gesture can be actioned to access menus and icons (Air-Tap gesture), create content (Double-Tap) as well as move it to its ideal locations (Hand drag).

#### Annotation Icon Design considerations

Designs were created for several types of annotation in the WEKIT Recorder and Player applications. To prevent confusion between Menu items for annotations from actual annotations placed in a scene the Menu Item for creating a given annotation has the annotation placed in a 3D ring or ‘Halo’ (Table 1).

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Text Annotation" /></td>
<td>Text Annotation</td>
<td>This annotation is modelled on the common icon for a document with lines on it representing writing. As text is flat relative to the 3D space, it is represented by a flat, 2D icon. For example, the icon for a Word Document: <img src="image" alt="Word Document Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Image Annotation" /></td>
<td>Image Annotation</td>
<td>The stereotypical icon used for adding images is a schematic of a two-mountain scene. As the content is 2D the Icon is made to look that way too. <img src="image" alt="Image Icon" /></td>
</tr>
<tr>
<td><img src="image" alt="Audio Annotation" /></td>
<td>Audio Annotation</td>
<td>Audio icons are usually represented by a speaker. Here a 3D model of a speaker was designed and used. <img src="image" alt="Audio Icon" /></td>
</tr>
</tbody>
</table>
Video Annotation

Video symbology usually focuses on the triangle representing the 'Play' function of media devices, sometimes combined with a Film Directors 'Clapperboard' for marking timing of scenes. As Videos tend to be 2 dimensional, a 2D icon is used.

Model Annotation

A Model annotation for placing a 3D model is best represented by a selection of 3D primitives such as low polygon representations of Cube, Sphere and Rhomboid shapes.

Glyph Annotation

3D models of X,Y,Z Axes and Circles are used to infer locate and rotate instructions to find a particular object or move it accordingly in 3D environment space.

Hands Annotation

To ask the user to move their hands in a particular way, e.g. to lift something or twist it a pair of low polygon 3D hands was designed. This is based on common icons for manual tasks being illustrated with hand glyphs.

Ghost-track Annotation

To create a trace of a user 3D motions through an environment, a low polygon 3D model of a human body is duplicated with a white ghost clone just behind it implying ghosting of the person's movements, including their head, body and hands.

**Icons for Adding Content**

Design of icons for **audio, video, hand- and ghost-track annotations**, was based on standard icons for such functions (Table 2).

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Start Recording" /></td>
<td>Start Recording</td>
<td>The standard symbol for recording usually involves a circle filled with red, sometimes with a ring around it.</td>
</tr>
</tbody>
</table>
Stop Recording

Stopping a recording process is indicated by use of a square surrounded by a ring.

Play

Playing recorded content is represented by annotations similar to the Video Annotation but with a green ‘proceed’ colour.

Pause

An icon to represent a Pause in the playback of stored content is a Ring with two vertical lines like a sideways ‘=’.

When placing images in a scene the icons image annotation, were designed so the user can either take a new photo or choose one from the gallery (Table 3).

**Table 3.** Image Annotation Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Take a Photo" /></td>
<td>Take a Photo</td>
<td>Taking photo of a scene requires use of the Hololens camera so a symbol of a Single Lens Reflex Camera model was used to represent this function.</td>
</tr>
<tr>
<td><img src="image" alt="Choose Image" /></td>
<td>Choose Image</td>
<td>To select an image, from a selection a picture of several image icons placed in a gallery or grid format were designed as if accessing an archive, picture wall or a repository.</td>
</tr>
</tbody>
</table>
Creating and deleting text in a scene can be done with the text annotation, the user may do the following (Table 4).

### Table 4. Text Annotation Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Enter Text" /></td>
<td>Enter Text</td>
<td>Normally typing text into a device involves either using a physical or touchscreen keyboard. Here we use a 3d model of a keyboard to act as the interaction element for placing text.</td>
</tr>
<tr>
<td><img src="image" alt="Erase Text" /></td>
<td>Erase Text</td>
<td>In the real world when writing with a pencil, text can be erased – usually with an eraser that is often coloured white or grey that is rectangular or trapezoid block.</td>
</tr>
</tbody>
</table>

To get a user to locate or move objects in an environment the glyph annotation, standard models are used for typical interaction methods (Table 5).

### Table 5. Glyph Annotation Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Locate (target)" /></td>
<td>Locate (target)</td>
<td>Concentric circles apparently radiating out from a central point are the basis of animations for location awareness (for example to see the pointer on a computer display) as well as in Internet map displays to show the users location.</td>
</tr>
<tr>
<td><img src="image" alt="Locate (arrow)" /></td>
<td>Locate (arrow)</td>
<td>Another location marker is some kind of pointer like the Google Maps pointer or arrows used to point out features in an illustration. As this is likely to point out features in a 3D space it is made of a cone and cylinder to infer a 3D location.</td>
</tr>
<tr>
<td><img src="image" alt="Rotate (clockwise or anticlockwise)" /></td>
<td>Rotate (clockwise or anticlockwise)</td>
<td>Rotation arrows normally used to rotate objects in a document can made from a ring segment and a cone again to infer movement along a circular path in 3D space.</td>
</tr>
</tbody>
</table>
2.2. WEKIT Player, Design of Interface Visuals

The Player task sequence begins with the Hololens camera recognising an AR target identified by the Vuforia software integrated into the Unity Development Platform. When choosing such visual targets, it is important that they have a unique number of identifiable points. This is best done if the image used is not symmetrical along as many axes as possible and uses lines which cross one another differently on different parts of the image. Vuforia website allows images to be uploaded and categorized for their AR suitability, which will be used to reference all further task components displayed. The calibration routine requires the printed calibration target found on the manual folder. The printed marker frame was sized at 20 cm x 20 cm.

![Anchor calibration tool marker](image)

**Figure 4.** Anchor calibration tool marker

When such a visual is a suitable AR Target, Vuforia gives it a high star rating and many unique points marked with a yellow ‘+’ symbol are seen. If there are only a few of these then the rating will be less and the image unlikely to be a good AR trigger. The image above was analysed and the result shown in Figure 5 was seen.
2.2.1 WEKIT Player, Interface Visual Design

A scenario task needs to be initiated through the Player Starting screen (Fig. 6). This was designed to allow a particular training sequence recorded by an expert to be run through the Player. For this an option was needed to access the file from the Hololens or Cloud Server. A field was provided to open the file through a neighboring Load button, with the option to overwrite any current activity by check marking a tick box to the left of the field. An icon to return to the main app screen was needed and one was designed with a picture of a House to infer the ‘Home’ screen of the app. Loaded activities needed to be placed in a list panel from which a user can select the activity they are interested in. Once an activity is loaded it can be activated from the list of loaded activities at the bottom of the panel. Icons were added to play the activity in the selected row or replay it through another icon.
An activity’s sequence of task steps needs to be shown (Fig. 7). A list panel was designed to load the steps into. The user can then select each in turn to go through the training required. To show that a step was complete a tick box was added to the row of each step and checked when the step was completed. The activity list panel was designed to show the Activities’ Title, the list of steps as well as icons to go back to the app’s ‘Home’ screen, return to previous screen to select an activity, Start the current list of steps in the activity and lastly an icon to keep the activity list menu locked relatively to the user’s eyes. At any time, the user may be required to check the sensors monitoring his performance. An icon with a human figure, for accessing the current state was designed and added to the centre of the Activity List panel.

Figure 6. Player starting screen

Figure 7. Activity overview
When the user accessed a particular step, it could be given as part of ‘Carousel’ type rolling stream of panels for the activity steps (Fig. 8). The design in such a configuration allowed previous and next steps to be accessed by tapping the panels to the left or right of the current foreground step panel. The main Activity card panel was designed to hold the step’s or current Action’s instruction.

![Figure 8. Activity card view](image)

Sometimes additional content is needed to expand the instructions so a ‘+’ icon was designed and added to the top left of the Activity card panel to access it and ensure the users was better informed. To get more information about a content element in the scene, the content panel can be activated (Fig. 9).

![Figure 9. Content view](image)

The content within the content element, in this case a picture of a Mars Rover, is then displayed to inform the user (Fig. 10).
It was also requested in the original prototyping of visual elements that a timeline be made available, so the user could see how far into the activity they had progressed. This was felt to be best placed below the main content requiring attention. Also requested was the possibility of controlling progress either through Air-Tapping or voice based triggers. Icons for selecting the appropriate trigger were added to the bottom right of the panel for ergonomic affordance. Some comments mentioned during prototyping requested to have a choice between moving through the Activity steps in a linear vertical list, much like a to-do list and a carousel mechanism similar to accessing icons on an Apple MAC computer’s dashboard. An icon was designed to be button that would allow the user to toggle between the task list and card /carousel views. An icon to close the menu at any time was also added to the bottom right of the card panel, again as it was a low priority/attention item. Typically, the user would have to switch views between card and task list views when a task was complete on the Activity Card and it need to be marked as complete on the Task list view (Fig. 11).

The sensor panel icon on being activated brings up a panel showing current state of the body sensors worn by the user (Fig. 12). A rainbow/heat map bar chart showing optimal and outlier ranges of sensor data were provided for each sensor and a triangle added to represent the current
reading. Two temperature readings, one for body and one for environmental temperature were displayed. The Human figures body colour was chosen to reflect the body temperature and was selected to be green if in range, red if outside optimal range and yellow if a transition between the two was in the process of happening. The space around the body would be similarly colored for the environmental temperature readings.

**Figure 12. Sensor panel**

On completing all steps, a flag is displayed indicating the end of the training scenario (Fig. 13).

**Figure 13. Last step view**

Visual elements can be activated by Voice commands. More details of these can be found in D2.5.
2.3 Sensors, data and statistical methods recap

Visualisation of Real Time Sensor information can be found in the Sensor Panel icon, whereas visualisation of data based on Statistical analysis of data to determine affective state and provide performance guidance is done in Post Analysis elements.

2.3.1 Real-time component

Original visualisation came from connecting to an Arduino to establish the nature of the data. For example, from the raw heart sensor signal the Beats Per Minute (BPM) was calculated as well as Heart Rate Variability (time between beats, Inter-Beat Interval, IBI) illustrated in Figure 14.

![Grove Heart Sensor: Arduino Serial Plotter Output](image)

Figure 14. Grove Heart Sensor: Arduino Serial Plotter Output

Real time sensor data was also streamed to a web page on a local laptop via MQTT protocol to ensure sensors were functioning prior to a user beginning their training session. These visualisations are seen in Figure 15.
Real Time visualisation, found on the Hololens WEKIT application Sensor panel, includes elements for Heart rate, Heart Rate Variability, Temperature, Humidity and Galvanic Skin Response.

The real-time component visualisation is displayed in Figure 16. The icon in the bottom bar of the User Interface (UI) window is shown whenever the sensor connection is active. The human icon in the symbol is representing the human sensor overall state and the icon background is representing the environment sensors overall state. Both the human and environment state can have three possible colours: green, yellow and red. If everything is good, both the human icon and the background should be green.

The small icon acts also as a clickable button to bring out the detailed sensor view. The sensor view can be also activated with a voice command "show sensors". The detailed sensor view also contains the same kind of icon representation of the overall sensor states. In addition to this, all the individual sensor values are displayed. The sensors values and their ranges, limits and units are defined in the ARLEM\(^1\) workplace JSON file.

\(^1\) https://standards.ieee.org/develop/project/1589.html
Figure 16. Real-time analysis UI in WEKIT player

Real-time sensors are categorized to three different types. The types are human, environment and device sensors. Supported protocols are MQTT and UDP. Sensors are linked to ARLEM workplace JSON thing, person or place categories. Device sensors can be linked with things, human sensors with persons and environment sensors with places. Multiple users can connect to device and environment sensors, but all the users have their own set of human sensors. This enables personalized configurations of the human sensors so that e.g. the heart rate limits can be matched to individual values. More detailed technical description could be found in WEKIT deliverable D4.4.

Method and System for Post Analysis of AR-based Learning and Biofeedback Data.

The real-time analysis relies on the green, yellow and red limits defined in the configuration file (workplace JSON). For the environment and human sensors, the analysis results can be seen either on the small human icon visible all times in the lower part of the main UI window or more detailed in the sensor view UI panel. Figure 16 shows a state where everything is ok. The analysis logic is that if any of e.g. the human sensors is within the red limits, the human icon is turned red (figure 17). Figure 18 shows a state where one of the environment variables is within the yellow limit. The problematic variable can be seen from the detailed view that shows graphs for all individual values. The graph limits are configured by the configuration file.
2.3.2 Post-task component

Current Post task visualisations require extensive analysis and processing, especially via Machine Learning with each data set typically taking several minutes to produce data. Specific data linking pairs of data variables such as Heart Rate and Temperature, Heart Rate and GSR or Temperature and GSR are being analysed. The final aim is to visualise beyond 2D variable spaces and visualise all three variables in a 3D volume, identifying regions within which ideal performance is located and...
raising flags where performance fell short of what was required. Ideal state is best derived from sensor data using charts generated from Support Vector Machine algorithms for machine learning, producing visual charts such as the SVM chart in Figure 19.

![Support Vector Machine (SVM) chart](image)

**Figure 19.** Use of Support Vector Machine (SVM) algorithms learning to establish if a candidate’s sensor readings (for example, x1=Pulse and x2=GSR) are within optimal ranges.

Affective state will be visualized on a Radar type chart that maps pairs of Sensor Data coordinates, such as Pulse and temperature as seen in Figure 20.

![Radar chart](image)

**Figure 20.** Affective state as a correlation between Pulse and Temperature

### 3. User interface design methodology

Four design sessions relating to different types of visual elements as well as visualisations were carried out. Each session informed future iterations and included Participatory Design sessions as well as illustration and design workshops with key partners.

#### 3.1 Real-time assistance

Real-time assistance component was developed by following the main principles of Human Centred Design (HCD) [3]. The main objective of a small-scale laboratory test is to get feedback from users of new 3D UI concepts together with real-time sensor panel in it. Same time were tested the concept of finding features of various annotation in 3D space. These elements were selected based on trial
feedback (D6.4, D6.5, D6.6). There were four human centred design cycles where users were able to give feedback of the new design and give and/or update new requirements for UI. HCD design cycles were in VTT Tampere Mixed reality lab and in VTT Otaniemi:

1. 5th-6th of October 2017, Main focus for the updated 3D UI concept
2. 30th-31st of October 2017, Main focus for new concept for finding objects in 3D space (and first ideas of sensor panel)
3. 11th-12th of January 2018, Main focus for evaluate 2D/3D symbols, 3D animation, text and sensor panel layout.
4. 9th-10th of April 2018 together with WEKIT technical meeting to collect feedback of pre-final version of the real-time assistance panel

Altogether 22 persons (12 males, 10 females) participated in the tests. Between the tests the system was updated based on user feedback and requirements.

More detailed description could be found in D4.2 - Guidelines for Information Adaptation to Multi-Device Presentation, and D4.5 - AR-visualization System and AR-based Experience Re-enactment and Learning System

4. Conclusion

Of the visual elements tested and for which user feedback was given, it was found that four key areas were noted for their affordances, advantages and criticisms. These related to Sequential navigation of Hololens application menus, Positioning of menu and scenario elements, feedback about how users interacted with elements as well as the Spatial Representation of elements.

4.1. Sequential navigation

Related to the visual nature of the curved UI panel, it was suggested that more information could be visible at a glance, not only the content of the active (middle) step, such as the additional functionality buttons. This can be seen in Activity view in Figure 5. More similarity with existing and well-known UIs and devices was suggested as well, for instance reference to the Cover Flow file arrangement on Mac OS. The Horizontal orientation of the active step shown in Figure 5 is more natural for available field of view and head position, than the vertical. Users also mentioned the effect of depth perception with the horizontal menu, and that animated cards behind the active card are good for understanding where you are in the training.

4.2. Positioning

The progress bar (below the menu) is helpful for progress tracking, but was not noticed by any participants, compared to the green step indicator field (1/8 steps) in the first version of the menu. It was also described that the user should be able to position the panel along the direction of the gaze. It should also be possible to activate this option with a voice command (the test system position commands included 'left', 'right' and 'low'); and suggested that the panel would follow the head movement, with an option to lock/unlock function, where you can attach it to the point in the environment.
4.3. Interaction Feedback

There were three ways of interacting with the system: using hand gestures, voice commands or clicker for air tapping. The positive expressions related to voice command were ‘functional’, ‘good enough’, ‘easy’ and ‘straightforward’. The negative descriptions included ‘puzzling’ and ‘slow’. For selecting and activating items on the UI panel, using gesture (air tapping) was preferred to the voice command by all participants. The participants described that the tap gesture felt ‘nice’, ‘pleasant’ and ‘rewarding’. On the other hand, voice commands were perceived as unnatural in this context and described by one participant as ‘stupid’. However, it was also seen as a good solution to have both possibilities (gesture and voice).

4.4. Spatial Representation

Comparing 2D and 3D icon representations, all the participants agreed that tools and content of the application should have different spatial representation, describing it as “there is no need for the main menu to be completely 3D, because it will be too distracting from the actual task”, or “menu should be familiar to computer user”. In relation of the types of icons that were more understandable and helpful to complete the task, one participant commented that “2D icons are good when we need to place something on a surface, and 3D — when manipulating the physical object”.
References


