D4.5 AR-visualization System and AR-based Experience Re-enactment and Learning System
(combined deliverable from D4.1&3)
## 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>23.3.2018</td>
<td>Kaj Helin (VTT)</td>
<td>Template</td>
</tr>
<tr>
<td>0.2</td>
<td>20.5.2018</td>
<td>Kaj Helin (VTT)</td>
<td>Summary and section 1</td>
</tr>
<tr>
<td>0.3</td>
<td>23.5.2018</td>
<td>Jaakko Karjalainen, Kaj Helin (VTT)</td>
<td>Sections 2, 3, and conclusion</td>
</tr>
<tr>
<td>0.4</td>
<td>24.5.2018</td>
<td>Alla Vovk (OBU), Kaj Helin (VTT)</td>
<td>Main findings from combined WEKIT Player and Recorder HCD cycle</td>
</tr>
<tr>
<td>0.5</td>
<td>25.5.2018</td>
<td>Jaakko Karjalainen (VTT)</td>
<td>Sections 4</td>
</tr>
<tr>
<td>0.6</td>
<td>25.5.2018</td>
<td>Kaj Helin (VTT)</td>
<td>Ready for internal review</td>
</tr>
<tr>
<td>1.0</td>
<td>30.5.2018</td>
<td>Kaj Helin (VTT)</td>
<td>Updated based on internal review. Ready for submission.</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.
AR-visualization System and AR-based Experience Re-enactment and Learning System

WP 4 | D4.5

Editors:
Kaj Helin (VTT)

Authors:
Kaj Helin (VTT), Jaakko Karjalainen (VTT), Alla Vovk (OBU)

Reviewers:
Mark Ransley (RAV), Fridolin Wild (OBU)

Deliverable number D4.5
Dissemination level Public
Version 1.0
Status Final
Date 30.05.2018
Due date 31.05.2018
Contents

Executive summary 5
WEKIT Abbreviations 6
List of Figures 7
1. WEKIT Player (Re-enactment) Architecture 9
2. Presentation layer 10
2.1. WEKIT.One version 10
2.2. Human Centred Design based laboratory test for Presentation layers UI development 11
2.3. Final and implemented results of Presentation layer 13
3. Application layer 23
3.1. Workplace manager 23
3.2. Activity manager 28
3.3. Sensor interface 31
3.4. Processing 33
4. Data layer 36
5. Conclusions 39
References 40
Executive summary

The WEKIT Deliverable 4.5 “AR-visualization System and AR-based Experience Re-enactment and Learning System” is report of WEKIT Re-enactment (AR-Player) demonstrator/prototype. This deliverable is combined version of deliverables D4.1 “AR-visualization System” and D4.3 “AR-based Experience Re-enactment and Learning System” as these two components cannot separate in demonstrator. Deliverables merging has been agreed with Project Officer.

Deliverable 4.5 “AR-visualization System and AR-based Experience Re-enactment and Learning System” describes main features of WEKIT Player which is following the main principles of the IEEE Draft Standard for an Augmented Reality Learning Experience Mode - ARLEM (IEEE: ARLEM, 2018).

Section 1 describes overall architecture of the WEKIT Re-enactment prototype. The architecture includes Microsoft HoloLens side the sensor and tracking environment, GUI and back-end. Also, in the architecture can be found connection to local and cloud-based data repository.

The next Section 2, provides overview to process for developing Presentation layer and detailed description of implemented solution. Presentation layer was updated based on feedback from trials. Updated were done in small steps within three laboratory tests. Laboratory tests were following the main principles of Human Centred Design. Altogether 16 persons (7 male, 9 female) participated in the tests. Between the tests the WEKIT Player was updated based on user feedback and requirements before the final implementations.

Section 3 describes the application layer, which has been described in sub-sections. Workplace and activity manager are parsing information from ARLEM based Activity and Workplace JSONs. Sensor interface, in our case MQTT allows WEKIT Player to get information from user and environment. And processing part, where WEKIT Players system logic exits.

The following section 4, gives summary of the Data layer. From WEKIT player point of view data layer includes all ARLEM based information for work support and training. More detailed information of the data layer could be found in deliverable D2.3 ‘Final Architecture and Learning Experience Content Model’.

Last sections 5 provides an overall conclusion of AR-visualization System and AR-based Experience Re-enactment and Learning System.
### WEKIT Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Augmented Reality</td>
</tr>
<tr>
<td>ARLEM</td>
<td>Augmented Reality Learning Experience Mode</td>
</tr>
<tr>
<td>HCD</td>
<td>Human centred design</td>
</tr>
<tr>
<td>MQTT</td>
<td>Message Queuing Telemetry Transport</td>
</tr>
<tr>
<td>POI</td>
<td>Point of Interest</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UI</td>
<td>User interface</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>WT</td>
<td>Wearable Technology</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1. WEKIT.one Re-enactments system components ................................................................. 9
Figure 2. WEKIT.One tasklist based user interface, which is located in 3D space including general warning symbols .......................................................................................................................... 10
Figure 3. Example from the second HCD based lab test for finding objects in 3D space .......... 11
Figure 4. Player activity selection screen .............................................................................................. 13
Figure 5. Player activity overview ........................................................................................................ 14
Figure 6. Player activity overview with sensors online ......................................................................... 14
Figure 7. Activity cards view .................................................................................................................. 15
Figure 8. Tasklist view ........................................................................................................................... 16
Figure 9. Next step in tasklist view ........................................................................................................ 16
Figure 10. Next step in activity cards view ............................................................................................. 17
Figure 11. Additional content view with audio content with content description ............................ 17
Figure 12. Additional content view with video content without content description ..................... 18
Figure 13. Video playback inside the additional content panel ............................................................. 18
Figure 14. UI window in locked state ..................................................................................................... 19
Figure 15. Detailed sensor view with all sensors on green .................................................................... 19
Figure 16. Detailed sensor view with one of the human sensors (temperature) on red ............... 20
Figure 17. Detailed sensor view with one of the environment sensors (humidity) on yellow ........ 20
Figure 18. Guiding feature for the content placed in 3D space ........................................................... 21
Figure 19. Example of warning signs and activity symbols in 3D space .......................................... 21
Figure 20. Action symbols ................................................................................................................... 22
Figure 21. ISO 7010 symbols .............................................................................................................. 22
Figure 22. Player scene starting state ................................................................................................... 23
Figure 23. Example thing configuration ................................................................................................. 24
Figure 24. Thing configuration result in player Unity scene ................................................................. 25
Figure 25. Detectable configuration for Vuforia image target ............................................................. 26
Figure 26. Vuforia image target for a device and the device itself ....................................................... 26
Figure 27. Detectable configuration for spatial anchor ....................................................................... 26
Figure 28. Detectable transform .......................................................................................................... 27
Figure 29. Linked thing object transform ............................................................................................. 27
Figure 30. Thing child poi object transform and configuration ............................................................. 28
Figure 31. Configuration for a single action step .................................................................................. 29
Figure 32. Tangible configuration and result in player scene ................................................................. 30
Figure 33. Tangible configuration result in player HoloLens view ....................................................... 30
Figure 34. Device sensor configuration with variable types, ranges and limits .................................. 32
Figure 35. Device sensor display showing the result of the configuration ........................................... 33
Figure 36. Predefined anchor calibration target .................................................................................... 34
Figure 37. Main principles for data content in WEKIT Player ............................................................... 36
Figure 38. ARLEM activity file showing the reference to the workplace file stored in a Dropbox folder ..........................................................37
Figure 39. Content flow from different repositories to the WEKIT Player runtime ...........37
Figure 39. Activation of a video file in activity file activity step configuration .................38
1. WEKIT Player (Re-enactment) Architecture

This section describes the final version of WEKIT Player (Re-enactment) architecture, which is following the main principles of the ARLEM standard. The final version will be available for trials on WP6.

Figure 1 illustrates the main system components of the WEKIT Player system that has been developed in Unity3D (Unity3D, 2018). The whole system is configured around the Activity JSON and the Workplace JSON files. The workplace JSON describes workplace-related information such as point of interest, sensors, etc. It is parsed with the Workplace manager and information is transferred to the data layer. Activity JSON describes all action steps and what content should be active in each of these steps. It is parsed with the Activity manager and information is transfer to AR layer via local storage and/or cloud back-end.
2. Presentation layer

2.1. WEKIT.One version

WEKIT.One Re-enactment (Player) was exploited in the first trials in all of the three cases. WEKIT.One Player was reported in D2.4 ‘WEKIT First Prototype’, where detailed description could be found. WEKIT.One player 3D User Interface (UI) concepts was tasklist based (see Figure 2) and it was located in 3D space. User was able to use gestures and voice commands to go for next step of the task.

![WEKIT.One user interface](image)

Figure 2. WEKIT.One tasklist based user interface, which is located in 3D space including general warning symbols

Based on the observation of the trials and trial evaluation reports (D6.4, D6.5 and D6.6) following elements were selected to future development:
- 3D UI concepts
- Function to support finding annotation in 3D space
- Replay function for video and/or audio playback
- Back function for tasklist
2.2. Human Centred Design based laboratory test for Presentation layers UI development

Laboratory test follows the main principles of Human Centred Design (ISO 9241-210:2010). The main objective of a small-scale laboratory test is to get feedback from users of new 3D User Interface (UI) concepts, finding features of various annotation in 3D space and form of annotations (2D/3D symbols, 3D animation, and text). There were three human centred design cycles where users were able to give feedback of the new design and give and/or update new requirements for UI components. These HCD cycles was held in VTT Tampere Mixed reality lab in (see Figure 3);

1. 5th-6th of October 2017, Main focus for the updated 3D UI concept
2. 30th-31th of October 2017, Main focus for new concept for finding objects in 3D space (see Figure 3)
3. 11th-12th of January 2018, Main focus for evaluate 2D/3D symbols, 3D animation, and text. This test was done cooperation with WEKIT Recorder UI test.

Figure 3. Example from the second HCD based lab test for finding objects in 3D space

Altogether 16 persons (7 male, 9 female) participated in the tests. Between the tests the WEKIT Player was updated based on user feedback and requirements before the final implementations.

Detailed results from two HCD cycles could be found in deliverable ‘D4.2. Guidelines for Information Adaptation to Multi-device Presentation’
Main findings from combined WEKIT Player and Recorder HCD cycle

The task of navigation is the most common in 3D environments. In case of augmented reality, it is a real physical movement in space that has its challenges such as supporting spatial awareness and making navigation lightweight. In this case navigation was tested in the first and second trials. On average among the participants the time on the first condition (no support) took 33 seconds to compete, second condition (semi-transparent background) took 28 seconds, and with third condition took 24 seconds to find all boxes. The line hints (Figure 3) were clearly preferred to other concepts in both object finding cases. In the first object finding case, ‘line hint1 with sound’ was considered to be the best by 4 participants, and one person preferred the line hint only (without the sound). All participants considered line hint and line hint with sound better than the sound hint alone. In the second object finding case, “sophisticated line hint2” was considered as best by four participants. One participant preferred the “finding without any support” to other concepts.

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).

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”.

Spatial interface recommendations
- 2D UI elements are better for position representation, 3D UI elements — for action representation (selection/manipulation)
- A combination of gesture and audio commands can be used with possibility to choose and use them in parallel
- Gestures: The objects (symbols) should be large enough for easy “clicking”
- Voice: Emphasis for system understanding of different accents / pronunciations
- 360° guidance for finding the object location in the 3D space; a circle visualizing 360° view in the gaze area for pointing out the objects in the work space with Gabor patch-type lines for

---

1 A direct line drawn from the centre of the user viewport to the object, Figure 3
2 A line drawn from the users feet following the floor until beneath the object and then raising up to the object, Figure 18
better direction orientation (the circle can be presented on the floor level additionally, but for ergonomic reasons the gaze area dominates)

2.3. Final and implemented results of Presentation layer

Feedback gathered from the user tests were used for finalizing the player UI. It became quite clear during the user tests that there doesn’t seem to be any UI concept that would please all the users. Hence the aim was to implement a flexible UI system that would support multiple ways for content presentation and interaction.

The following illustrates the main functionalities of the presentation layer together with screen shots from the actual system. The first picture (Figure 4) shows the activity selection screen that is shown first when the player is started. All the available activities are listed and the user can load an activity by simply clicking the button. In the picture there is only one activity (ALTEC demo activity). The user can also load additional activities to the player by entering an activity url pointing to the activity JSON file. The loading process loads all the related content (JSON files, images, videos etc.) and stores them on the local file system to enable offline usage. Once the activity is loaded, it is added to the list of available actions.

![Figure 4. Player activity selection screen](image)

The next picture (Figure 5) shows the overview of the loaded activity. The overview shows how many steps there are in the activity and the titles of all the steps. There is a play button in the lower right section of the menu and the user can start the activity by clicking that or by using voice command “start”. Whenever a button is pressed, the user will hear a clicking sound. Also whenever a voice command is received, the user will hear a confirmation from the system. This way the user will always know if the input has been received. This has proven to be vital since
the clicking gesture on the HoloLens doesn’t always get through and the functionality of the voice commands depend on user accent etc.

Figure 5. Player activity overview

When the activity is loaded, player starts the sensor connection routine in the background. Once the sensors are connected, the sensor icon appears in the menu as seen in Figure 6 sensor connection is also indicated by an audio feedback.

Figure 6. Player activity overview with sensors online
When the action has been started either by clicking the button or using the voice command “start”, the player shows the activity cards. This is the default behaviour, but the user can also use just the tasklist view (Figure 8). The user can switch between the two presentation modes by either clicking button in the lower left part of the menu or by using voice commands “show actions” to show the activity cards view or “show tasklist” to show the tasklist view. The activity card view shows 5 steps at a time and the active step is displayed in the middle. In addition to the step title, there is also a place for text instructions on the cards. The lower right section of the cards shows all the active triggers in the step. So if the action step has an active touch trigger, the user can move to the next step by simply clicking the next card of the stack. If there is also an active voice trigger, the user can use voice command “next” to move to the next step. The third possible trigger type is the IoT trigger which can be used to move to the next step when e.g. a certain button is pressed on an IoT enabled device. The user can also move back by clicking the previous card in the stack or by using the voice command “back”. If the previous step contains an IoT trigger, the user can’t reverse to that step just to prevent any issues of e.g. IoT trigger launching automatically since the desired values was already set.

Figure 7. Activity cards view
When a step is completed, a checkmark is enabled to indicate that this step was completed. This can be seen either after the title on the tasklist view (Figure 9) or at the top left corner of the previous card on the activity card view (Figure 10). The active step is once again the middle card of the stack and the user can move back or forward based on the active triggers. Figure 10 shows also the + indicator for available additional content on the upper left corner of the active card. The user can show this either by clicking the active card or by using the voice command “show content”. If the user tries to use the voice command when there isn’t any content available, the system gives feedback “there isn’t any content to show”.

Figure 9. Next step in tasklist view
Additional content view shows the content type and an optional short description of the content. The user can open the content by clicking the content type symbol. Possible content types are audio (Figure 11), video (Figure 12) and images. Content is added to this view whenever the action step contains any of these file types. The content can be shown localized in the 3D space e.g. close to the working area but it is also reachable from the additional content panel. This way the user can easily replay the content if needed. Images and videos are played back in window inside the content panel (Figure 13). The content view can be closed by either clicking the card icon in the lower middle section of the content view or by using the voice command “hide content”.

Figure 10. Next step in activity cards view

Figure 11. Additional content view with audio content with content description
By default, the UI window follows the head movement of the user. It is still designed to stay out of the way by allowing small head movements before the follow feature kicks in. The user can also lock the menu in place by either clicking the small lock symbol on the bottom right section of the panel or by using the voice command "lock menu". The button icon indicates if the menu is locked or not (Figure 14). When the panel is locked, it stays in the locked position. This means that if the user locks the window in place and moves e.g. 2 meters to the left, the UI window is 2 meters away. This enables the user to have the UI window in a specific location for the whole activity which can be helpful in some tasks e.g. in confined spaces. The UI window can be unlocked by either clicking the lock button again or by using the voice command "release menu".
The small human icon in the lower middle section of the UI window shows the real time state of the WEKIT sensors. The human symbol colour represents the state of the human sensors and the icon background colour represents the state of the environment sensors. So, if everything is fine, both the human and the background are green. The user can get more detailed information by either clicking the icon or by using the voice command “show sensors”. Figure 15, Figure 16 and Figure 17 shows the detailed sensor view in various conditions. The examples were using an eMotion Faros (eMotion Faros, 2018) sensor for the human readings and a DHT11 (DHT, 2018) sensor for the environment readings. The details are described in the deliverables D3.2 and D3.3.
Since the activities can have content placed anywhere in the 3D space, the user cannot just rely on the main UI window. A guide functionality was implemented to help users locate the content in 3D space. The guides can be activated by the action step or the guiding feature can be enabled globally by either pressing the magnifying glass icon on the lower left portion of the UI window or by using voice command “show guides”. The principle behind the guides is that an orange glowing ring is placed at the feet of the user. The edge of this ring contains arrows pointing to the direction of a content item in 3D space. The content and the arrow are connected by an orange glowing line that follows the ground plane right beneath the content item and then rises up to the item itself (Figure 18). The guiding feature can be turned off globally by clicking the magnifying glass icon again or by using the voice command “hide guides”.

Figure 16. Detailed sensor view with one of the human sensors (temperature) on red

Figure 17. Detailed sensor view with one of the environment sensors (humidity) on yellow
The content placed in 3D space can be images, videos, 3D models, text notes or various predefined symbols, see example in Figure 19. Figure 20 shows the available action symbols. So when the user is instructed to e.g. rotate a certain potentiometer, the rotate cw symbol can be placed in the location of the potentiometer to indicate both the action location and the desired action. In addition to the action symbols, there is also a full set of safety and warning symbols of ISO 7010 standard (see Figure 21) (ISO 7010:2012).
Figure 20. Action symbols

Figure 21. ISO 7010 symbols
3. Application layer

Application layer is implemented in Unity3D. Application layer is configured by the ARLEM JSON files. The workplace JSON is used for setting up all the things, persons and places together with detectables and sensors. The activity JSON is used for telling what content should be placed to the things, persons and places in each action step. The following describes this procedure in detail.

3.1. Workplace manager

When the player is started, the Unity scene contains just the empty parent objects for different workplace object types (Figure 22). The workplace manager parses the ARLEM workplace JSON file and creates the child objects based on the JSON configuration. Figure 23 shows a simple configuration for a thing object and Figure 24 shows the result in player Unity scene. The same approach applies to all workplace object types. So e.g. a detectable configuration creates a detectable object as a child of Workplace/Detectables object.

![Figure 22. Player scene starting state](image)
Figure 23. Example thing configuration

```json
{
  "id": "thingExample",
  "name": "Thing Example",
  "type": "extended",
  "detectable": "wekltLogo",
  "pois": [
    {
      "id": "default",
      "offset": "-1.3, -0.2, -0.1"
    },
    {
      "id": "potentiometerA",
      "offset": "-0.2, -0.3, 0.05"
    },
    {
      "id": "potentiometerB",
      "offset": "-1.3, 0.3, -0.1"
    },
    {
      "id": "buttonA",
      "offset": "-1.3, 0.3, 0.0"
    },
    {
      "id": "buttonB",
      "offset": "-1.3, 0.3, -0.75"
    }
  ]
}
```
Workplace objects are linked to the physical 3D space via the detectable objects. Detectable objects can be either Vuforia targets (Vuforia, 2018) (Figure 25 and Figure 26) or spatial anchors (Microsoft mixed reality, 2018) used by the HoloLens. Things, persons or places can be linked to detectable objects, which will then simply pass their transform values to the linked object (Figure 27, Figure 28 and Figure 29) and tell the linked object to either show or hide its content based on the detectable state. The detectable transform value sets the transform of the parent thing, person or place object. The position of e.g. an individual potentiometer (Figure 30) is combination of the position of the detectable feeding its transform to the thing object parent and then the thing child object potentiometer position in relation to the thing itself defined by the thing poi configuration.
Figure 25. Detectable configuration for Vuforia image target

Figure 26. Vuforia image target for a device and the device itself

Figure 27. Detectable configuration for spatial anchor
Figure 28. Detectable transform

Figure 29. Linked thing object transform
3.2. Activity manager

Activity manager parses the ARLEM activity JSON file and handles the activity flow based on the configuration. The activity JSON files contains a list of actions that contain the configuration of each action steps (Figure 31). The main components of the configuration are the enter and exit loops with activation and deactivation in both, the triggers and the instruction.
Figure 31. Configuration for a single action step

The enter loop is activated when an action step is activated. The enter activation loop then activates objects based on the configuration. The configuration defines what type of content should be activated, should it be initialized and where the content should be placed. The activation “id” field is mapped to a thing, person or place object defined in the workplace JSON file. This means that the “id” is a game object in the player Unity scene created by the workplace manager. The field “poi” defines in which point of interest inside the parenting thing, person or place object should the content be placed. Field “type” can be either “tangible”, “action” or “reaction”. Tangible types are for activated content and the “predicate” field defines the content type. The rest of the fields are used for configuring the content type as needed (Figure 32 and Figure 33). The deactivation loop deactivates content, actions or reactions. If the content is activated, it remains in the scene until it is deactivated. This way it is possible to easily have some warnings etc. active during the whole activity. The exit loop is activated when an action or reaction is deactivated.
The triggers are used for triggering the exit loop of the currently active action. The possible trigger types are click, voice and IoT. Click triggers enables the user to simply click the next card on the activity card stack or click the check box in the tasklist view to launch the trigger. The voice trigger enables the "next" voice command to launch the trigger. The IoT trigger can be used with various sensors to launch the trigger when a desired sensor state is achieved. To enable a sensible logic flow, the action exit activation loop should contain activation for the next action step.
3.3. Sensor interface

Player supports various real-time sensors that can be used for simply providing and visualizing data or to trigger action steps. Sensors are divided into three categories which are human, environment, and device sensors. Human sensors can be linked to workplace person objects, environment to workplace place objects, and device sensors to workplace thing objects. The human and environment sensors are used for providing data to the main UI sensor display and the different limit values are configured in the workplace JSON file. This way the system can support e.g. different heart rate ranges of the users. Device sensor visualization is presented on the linked thing object. Also, the device sensor is configured with the workplace file, where the variable types, ranges, and limits are defined (Figure 34). Figure 35 shows a device sensor display in HoloLens view. Supported sensor protocols are MQTT and UDP.
Figure 34. Device sensor configuration with variable types, ranges and limits
3.4. Processing

The overall process starts from the selection and loading of an activity file. If the user is loading an activity from an url, all the content is downloaded and stored on the local file system. Once the activity is loaded, it is added to the list of available actions. The user can then load an activity by simply clicking the button showing the title of the desired activity.

Once an activity is loaded, the workplace JSON defined in the activity JSON is parsed by the workplace manager. The parsing of workplace JSON creates all the content placeholder objects, detectables and sensors to the player scene. Once the workplace JSON is parsed, the workplace manager informs the activity manager that everything is ready. This brings out the main UI window with the activity overview view active.

If the workplace file contains detectables that rely on the spatial anchors and the anchors are not yet in the device specific anchor database, the system informs that the anchors has to be calibrated first. This is done by using a predefined calibration target (Figure 36). The workplace JSON detectable configuration for the anchor types should contain offset values that are calculated from the origin of this calibration target. This means that there should be a predefined fixed location used for the calibration target and every time the calibration is done, the marker
should be in exactly the same place. The calibration routine picks up the location of the calibration marker in the HoloLens coordinate system and creates the anchors to the positions defined by the workplace JSON offset definitions. The calibration needs to be done only once per device, since the created anchors are stored on the device anchor database that is persistent. By using this calibration process, it is possible to replay activities on different HoloLenses that were not used for recording the content.

![Workplace calibration tool](image)

**Figure 36.** Predefined anchor calibration target

When the user starts the activity by clicking the activity start button or by using the voice command “start” the action step defined as the starting action in the activity JSON file is activated. This activates the enter activation loop of this starting action. If there is some content activations defined in this step, all the content is instantiated on the scene based on the activate configuration. The configuration tells where this content should be placed and the content is instantiated as a child of the target object and point of interest. The transform of this target is updated by the detectable object linked to the target object. This enables the placement of content in the 3D space tracked by the HoloLens tracking and/or Vuforia tracking. In addition to the content placed in the 3D space, there are instructions available on the main UI action card and tasklist views. If activated content is either audio, image or video file, the same content is also added to the main UI additional content panel. If the user is having difficulty to locate the content in 3D space, the guiding feature can be enabled by the action step configuration or by the user either clicking the magnifying glass button in the main UI or by using voice command “show guides”.
The completion of an action step depends on the triggers configured in the currently active action step. Action step can be completed by either clicking the next card in the activity cards view or the checkbox in the tasklist view or by using voice command “next”. A step can also be completed automatically when a monitored IoT value matches the IoT trigger configuration. When a trigger is launched, the exit loop of the currently active action step is launched. The deactivation loop is launched first. This removes all the content defined in the deactivate configuration. If the activated content is not explicitly deactivated, it remains in the scene from step to step. The exit activation loop should contain an activate configuration for the next step to enable sensible step by step activity flow.

When the next step is activated, the previous step is marked completed by adding the checkmarks in the activity cards and tasklist views. In the activity cards view, the activated step is moved to the middle card of the stack, the previous as the left card from the middle and the next step as the right card from the middle. Once again, the content is activated based on the action step configuration enter activate loop item configurations and the trigger configurations define how the user can move on from this step.

When the user reaches the last step, by default, a restart button is enabled in the main UI view. By clicking this button, the activity is simply reloaded and it starts from the beginning again. It is also possible to force the activity to load a different activity JSON on completion of the first one or return back to the main menu.
4. Data layer

Figure 37 shows the main principles for data content in WEKIT Player. The previous chapter describes the details of the WEKIT Player layer and on this chapter, we concentrate on the content repositories.

As described in chapter 3.4, WEKIT Player starts with activity selection screen. The user has a possibility to load an external ARLEM activity JSON file from an url. This file can be located in any place reachable by Unity 3D WWW class (WWW class, 2018). This can be either a dedicated cloud storage solution or e.g. a Dropbox folder. The activity file contains reference to the workplace file (see Figure 38) so the both key files can be loaded with a single url. Also, if the activity file contains references to any additional activity JSON files, they all are loaded during the load process.
Figure 38. ARLEM activity file showing the reference to the workplace file stored in a Dropbox folder

Figure 39 shows the content flow from different content repositories to the WEKIT Player runtime. Both the activity file and the workplace file contain references to content behind different URLs. Activity file contains references to images, videos, sound clips etc. and the workplace file contains references to resources needed for e.g. Vuforia image targets etc. During the activity file loading process from an external URL, all this content is downloaded and stored to local file system together with all the JSON configuration files. Workplace file contains the references to various sensors. The sensors are configured and connected based on the workplace file configuration and they feed values that can be displayed in real-time in the player runtime.

Figure 39. Content flow from different repositories to the WEKIT Player runtime

During the runtime process, the activity file configuration defines what content should be fetched to the runtime UI in each activity step (see Figure 40). During the runtime process, the content is already fetched and stored locally, so the activated content is always fetched from the device local storage.
More detailed information of the data layer could be found in deliverable D2.3 ‘Final Architecture and Learning Experience Content Model’.

Figure 40. Activation of a video file in activity file activity step configuration
5. Conclusions

The WEKIT Deliverable 4.5 assesses the completion of the final prototype of the WEKIT Player (Re-enactment). WEKIT Player will be evaluated in the trials by our industrial partners (Lufttransport, Ebit, and ALTEC) using scenarios described in D6.1, D6.2 and D6.3.

Based on three Human Centred Design cycles and several demos we can conclude that the ‘one-size-fits-all’ 3D UI does not work. WEKIT Player was modified based on this assumption. User has three options to interact with UI: (1) gestures, (2) voice commands and (3) clicker. Same analogy work with UI layout. User could choose (1) list type or (2) card type of UI presentation.

WEKIT.One Player performances were better than the expectations. The results of the prototype in terms of usability were really good. In the trial 1, the System Usability Scale (SUS) score was ~69 (on an average of 70) indicating that the system usability is already acceptable in the first prototype. We could say that after several testing and UI modification the system usability should be even better.
References


Unity3D, https://unity3d.com/, last accessed 2018/05/21

ISO 9241-210:2010 Ergonomics of human-system interaction -- Part 210: Human-centered design for interactive systems

eMotion Faros™ Cardiac Monitors, http://ecg.biomation.com/faros.htm, last accessed 2018/05/21

DHT temperature & humidity sensors, https://learn.adafruit.com/dht/overview, last accessed 2018/05/21

ISO 7010:2012 Graphical symbols. Safety colours and safety signs. Registered safety signs

Vuforia, https://www.vuforia.com/features.html, last accessed 2018/05/21


WWW class in Unity3D, https://docs.unity3d.com/ScriptReference/WWW.html, last accessed 2018/05/24