Exploring Tele-Assistance for Cyber-Physical Systems with MAUI

Philipp Fleck[1], Fernando Reyes-Aviles[1], Christian Pirchheim[1], Clemens Arth[1,2], and Dieter Schmalstieg[1]

[1] ICG, Graz University of Technology, Inffeldgasse 16/2, 8010 Graz, Austria
[2] AR4 GmbH, Strauchergasse 13, 8020 Graz, Austria
{philipp.fleck, fernando.reyes-aviles, pirchheim, dieter, arth}@icg.tugraz.at

Abstract. In this paper we present an improved version of MAUI [9] (*MAUI - Maintenance Assistance User Interface*), extending the user-study, giving detailed insight into the implementations and introducing a new User-Interface for mobile use. MAUI is a novel take on tele-assisted tasks on cyber-physical systems. In its core we do not only provide real-time communication between workers and experts, but also allow an expert to have full control over the worker's user-interface.
By precisely separating levels of our software-stack, we enable features like hot-patching and hot-loading of any content or web-based application. Our results show reduced error-rate on task performance once an expert takes over virtual task to relieve the worker. Conditions evaluated include the worker operating on his own and the worker being actively supported. While in operation gauges have to be read, switches to be press and virtual user-interfaces of machinery have to be controlled. Furthermore, we explore the creation of user-interfaces through a developer-study and use the feedback to create a mobile version of MAUI.

Keywords: Remote Collaboration · Telepresence · Augmented Reality.

1 INTRODUCTION

Almost every industrial machinery (especially in production) and many customer devices are inter-connected nowadays. They can be summarized as Cyber-Physical Systems (CPS) because of their dual nature: they often have a physical interface like buttons and switches, and a virtual representation like companion Apps or Web-Interfaces for additional controls.

With increasing complexity, CPS become more and more difficult to operate, repair and maintain. Such tasks exceed the use of simple tools and often require diagnostic instruments or virtual controls *e.g.,* for re-booting machinery. Furthermore, for precise error-cause detection sensor telemetry data has to be checked and verified *e.g.,* malfunctioning valve showing increased pressure. Such

Fig. 1. Facility for production. (*top-left*) An expert in the control center dispatching; (*top-right*) Mission control with an dedicate expert Laptop; (*bottom-left*) a worker trying to fix an IT-mess wearing a HoloLens; (*bottom-right*) a worker on the shop floor, wearing a Microsoft Hololens, to fix machine downtime problems, taken from [9].

situation (*e.g.,* a broken machinery with physical damage and virtual configuration) are often combined with stress and can quickly overwhelm maintenance personal.

In particular with respect to industrial use cases, an important aspect is that machine manufacturers are not the actual machine operators usually. The engagement of support personnel from the manufacturer side in fixing maintenance problems on the operator side is a huge cost factor, which drives the inclusion of the tele-assistance concept in the maintenance workflow. Fig. 1 depicts and experts and workers operating such facilities. In its most simple form, this is realized through a regular phone-call and through picture sharing using smartphones and services like email. An even more evolved concept includes apps like *WhatsApp*, video-conferencing and remote-desktop systems like *TeamViewer*. Overall, remote support is used as problem solving and guidance tool.

The pure implementation of a tele-presence concept like this does not solve all problems automatically, however. One of the critical issues remaining is establishing context: *This button? – no the left one! The side where the red light is? – No, on the other side!* On the one hand, there is the spatial context, *i.e.,* the relation of objects in the environment, and on the other hand the virtual context, *i.e.,* any interact-able user-interfaces. Using simple audio calls or even picture sharing, context is still establishing through audio communication, instructions and human interpretation. Traditional remote-desktop software in turn does not allow for *integrated* operation like swapping user-interface of the worker to a more task-related one. Control inputs are only allowed including mouse or keyboard. These concepts leave a lot of room for errors and are solutions not as efficient as required by most tasks.

In this work, we present a tele-assistance concept and implementation, that allows a remote expert to further support the worker by taking over virtual tasks and therefore freeing mental capacities. In this regard, to guarantee hands-free operation, we let the worker wear a Head-Mounted-Display (HMD) with built-in camera and microphone like the Microsoft Hololens. Thereby the remote expert is provided with the video- and audio-feed from the worker and a rich data connection to be used for additional content. Support can be given by sharing annotation in the shared video, by spoken instructions and by interaction with the workers virtual environment *directly*.

Since we hereby extend our previous work *MAUI - Maintenance Assistance User Interface* [9], we further focus on industrial use cases, but remain in the same way applicable to applications involving consumer grade hardware. We illustrate the ease to write commands, also giving details on technical solutions to enable real-time performance on the Hololens. Such commands allow the expert to send almost any content to the worker, to directly visualize or open it and to manipulate the UI of the worker. Because of the complexity of CPS, workers can be overwhelmed at handling it. It comes as a relief that the expert can take over the virtual interaction. We demonstrate that by extending common concepts of remote desktop software into AR, UI manipulation is entirely doable beyond keyboard and mouse controls through such a concept allowing hot-patching. Finally we performed an extended user-study focusing on the mental and physical demand of a typical maintenance task and give insights into occurred task-error-rates of the participants.

MAUI was designed with the intention of more efficient subdivision of work between expert and worker, providing the following contributions:

- We analyze the requirements of AR tele-assistance for industrial jobs, where remote operation and configuration of a user interface by a remote expert is required.
- We present implementation details of MAUI, which address the aforementioned requirements with its robust abilities for sharing audio, video, digital content and control state in harsh industrial environments.
- We discuss results of a user study demonstrating lower task completion times, reduced cognitive load, and better subjective comfort on performing a maintenance task.
- We discuss the results of an exploratory study, orthogonal to the first one [9], analyzing how web developers perform when creating user interface content in the MAUI framework.

2 RELATED WORK

AR can help a worker by purely displaying digital information. Henderson *et al.* [18] have shown the benefits of AR in the maintenance and repair domain.

However, such pre-configured information sources are often unavailable. In this case, a good alternative is to link the worker to a remote expert providing live support. Dealing with a CPS adds the dimension of a digital interface, which can be controlled locally by the worker, or alternatively by the remote expert. Thus, our work is at the nexus of remote collaboration, interaction with cyber-physical systems and remote desktop user interfaces. Background to each of these topics is provided in the remainder of this section.

2.1 Remote collaboration

Video transmission is the enabling technology for tele-assistance since the pioneering work of Kruger [23]. Early works in this space [2,34] were mostly constrained to desktop computers due to technical limitations. Recent progress in mobile and wearable computing has brought video conferences abilities to the factory floor, *i.e.,* primarily leveraging mobile handheld devices like smartphones and tablets.

However, establishing a shared spatial presence at the task location is still a challenging topic. Experts need to visually experience the worker's environment. The video stream from a camera worn by the worker will only show the worker's actual field of view [19,3,20,5,24]. Giving the remote expert independent control of a robotic camera [25,15] is usually not economically feasible.

Apart from spoken instructions, most tele-assistance solutions let the expert provide visual-spatial references, either via hand gestures [28,19,21], remote pointing [3,12,20,5], or hand-drawn annotation on the video [12,15,6,20,30]. Hand-drawing is either restricted to images from a stationary camera viewpoint [20,15,19,3,6,12,21,24] or requires real-time 3D reconstruction and tracking [5,26,14,13].

2.2 Cyber-physical system Interaction

None of the remote collaboration systems mentioned in the last section takes into account the special requirements of a task that must be performed on a cyber-physical system. The dual nature of a cyber-physical system implies that each task will commonly consist of a physical task (*e.g.,* physical part mounted with screws) and a virtual task (*e.g.,* re-initializing a device after repair). It is crucial for the worker to receive support on both aspects of cyber-physical tasks.

Recent work in interaction design is starting to consider such interactions with CPSs. Rambach *et al.* [31] propose that every cyber-physical system serves its own data (*e.g.,* sensor information, control interface) to enable new ways of interaction. Alce *et al.* [1] experimentally verify different methods of device interaction while increasing the number of devices. A common design are device-specific controls embedded in the AR user interface [8,7].

Other recent work investigates AR in production and manufacturing industries. Kollatsch *et al.* [22] show how to control an industrial press simulator within an AR application by leveraging its numerical interface, *i.e.,* a device interface allowing to run simulations which are partially executed on the machine. Han *et al.* [16] concentrate on automated situation detection and task generation to give better and more responsive instructions, *e.g.,* for fixing paper jams in printers.

Cognitive conditions during task performance are a key element to success in many industrial situations. Maintenance workers must frequently perform multiple repair tasks during one shift, requiring a high level of flexibility and concentration. Therefore, recent research has considered how reducing factors like frustration, stress and mental load can improve overall performance. For instance, Baumeister *et al.* [4] investigate mental workload when using an AR HMD. Funk *et al.* [11,10] compare instructions delivered via HMD to tablet computers and plain paper. Recently, Tzimas *et al.* [33] reported findings on creating setup instructions in smart factories.

2.3 Remote desktop user interfaces

Remote desktop tools, such as Skype[3] or TeamViewer[4], combine video conferencing with remote operation. In theory, these tools have the features required for worker-expert collaboration and can be made to run on AR headsets such as the HoloLens. However, a closer inspection reveals that the similarities to the desired solution are shallow. Desktop user interfaces are operated using mouse and keyboard. They do not work very well when one user has reduced resources (*e.g.,* when using a phone with a small screen) or when network connectivity is unstable. Workers do not want to retrieve files and navigate them manually, while they are tending to a task. Moreover, shared spatial presence between worker and expert is not considered at all in desktop tools. Even re-using parts of desktop tool implementation in an AR applications turns out to be hard because of the differences between desktop and mobile operating systems.

Perhaps closest to our approach in this respect is the work of O'Neill *et al.* [29] and Roulland *et al.* [32]. Like us, they present a concept for remote assistance, focused on office printer maintenance. However, unlike ours, their work relies on schematic 3D rendering of a printer device, delivered on the printer's built-in screen, and very few details are provided on the implementation and extensibility of the system. In contrast, MAUI is a comprehensive tele-assistance framework. We describe details about its implementation, and evaluate the system's development and use.

3 REQUIREMENT ANALYSIS

3.1 Analysis of Collaborator Feedback

First, we need to discuss a few relevant questions important for the design process, supported by collected statements from our collaborators, operators of multiple production facilities around the world manufacturing 24/7.

[3] Skype: `https://www.skype.com`
[4] TeamViewer: `https://www.teamviewer.com`

What kind of problems do we face? Within manufacturing industries time is precious and costs caused by unexpected stalls are high. Therefore on-call staff (*i.e.,* workers) is always ready to jump in an fix any occurring problems. Broken machinery usually signals some alarm and state, and in certain cases production will stall. As a consequence, downtimes are kept short to reduce costs and additional failures. Increasing the efficiency of repair personal is key in such situations. Unfortunately, knowing about the downtime costs even increases the mental pressure on the individual and can lead to less concentration and further errors.

How are common tasks in industries performed? Often workers call and negotiate with remote personal on how the repair is done, what exactly is the issue and if other systems are affected. The identified tasks are problem explanations, collaborative solution finding, data observation and (guided) problem solution. Since all machinery nowadays is interconnected and represents a CPS, solving tasks does not only include manual fixing but also interaction with virtual interfaces of devices. Workers need to be knowledgeable in both areas.

What are the most important features? Due to the complexity of installations, while performing operations machine telemetry and on-machine gauges have to be monitored constantly. Therefore this information have to be ideally visualized in the field-of-view of an operator at all times.

Obviously, it is counter-productive to replace a complex system (CPS) with an another complex one. By slowly introducing new features through levels of expertise, it is possible to pick everyone up at his own learning speed. AR enabled and hands-free solutions are considered a particularly important feature. As a consequence, minimizing interactions with gaze-based controls (including air-taping and Bluetooth clickers) enables a further improvement and acceptance of solutions.

Finally, in industries the use of established software technologies is favored over proprietary solutions. A web-centric approach, inspired by Argon [27], leveraging existing developer skills and allowing for run-time extensibility is considered important.

What kind of content is mostly used? Remote support stuff contains well trained experts in their fields. Those usually have a good understanding of CAD models and how CPS operate and behave, forming the ideal counterpart to the workers in the field. Often sharing an instruction manual or a cross-section of a part can drastically improve understanding and support the solution finding process. This fact suggests that experts have precise control about the worker's UI and other contents for supporting the work in solving a common problem. The desktop application (*i.e.,* the expert side) can be more complex and offer more gradual controls, while the operator side should be as lightweight as possible.

What about non-technical challenges? As a somewhat surprising outcome, bi-directional video feeds (as opposed to one-directional from the worker to the expert) were considered of utter importance. The most important argument was that this establishes trust between the expert giving instructions, and the operator on the shop floor.

Another challenge, but also opportunity, was identified to be significant age distribution, which ranges from some *digital natives* to more workers in their mid fifties, also with varying educational background. In western countries this gives us ten more years of the same workers with an, in part, antiquated view for digital contents. On the one hand, to on-board all groups of workers, features have to be easy to handle not to overwhelm the individual. On the other hand, designing an easy-to-use system enables the formation of an expert pool consisting of those operators most experienced in practice, essentially increasing their efficiency through sharing their knowledge as experts themselves.

3.2 MAUI Design decisions

Following this feedback, we distilled a set of core requirements and made a few design decisions for MAUI. Our system primarily aims at helping workers while being in a stressful situation solving an unknown problem. Using a HMD frees the worker's hands and allows any interaction with machinery and tools utilizing a minimal design to reduce distractions. While having a Audio-Video face-to-face conversation, the expert should be capable to manipulating (change, alter, replace, control) the worker's user-interface, share multimedia contents (images, PDF, screen-capture) and point within the video feed. Any UIs should be as easy as possible to understand.

On the technical level, this relates to the following tasks to be solved for the worker side:

- Initiating an audio/video connection with easy connection management including a phonebook and online user discovery;
- 3D mesh transfer functionality to enables the worker to send a scanned 3D model of the current environment to the expert, including the worker's current position (Leveraging device capabilities *e.g.,* accessing the HoloLens mesh, but also allowing to plugin other 3D-reconstruction systems);
- Functionality to visualize information, and corresponding methods to manipulate this information, *i.e.,* the UI, remotely.

In turn, for the expert application, the following list of features is important:

- screen capture functionality, to allow the expert to share screenshots of running applications (*e.g.,* CAD model viewer showing a cross section of a broken machinery part) with the worker;
- use of multimedia content in a representation-agnostic form (PDF, HTML, images, links) to be transferred and displayed;
- use of multi-page documents, such as PDF files, respectively synchronized navigation between an expert and the worker; optionally enriching this content with shared annotations;

– functionality to control the worker's UI, including web content, but also to trigger native interfaces of the cyber-physical system.

The last feature, remote UI control, implies that a custom UI must be dynamically embedded in the tele-assistance application. This discharges the worker from switching apps or configuring the user experience. The ability to change the UI without touching the underlying application makes it also easy to embed tutorial functions into the UI itself, which can be step-by-step enabled. As an advantageous side-effect, this also facilitates a gentle learning curve for the worker. Once the worker is sufficiently familiar with the UI, more features can be enabled, *e.g.,* transitioning from 2D to 3D visualizations or enabling additional controls for physical devices.

4 IMPLEMENTATION DETAILS

Head-worn AR devices like the HoloLens have not reached mass-adoption yet, and will remain expert tools for a couple of years. In the smartphone and tablet sector however, new AR technology is introduced more frequently, including *e.g.,* ARKit and ARCore SDK features. This implies that MAUI should be able to handle a wide range of current and future devices and to be inherently cross-platform.

4.1 Basic Software Design

We chose Unity3D as our 3D Engine because of its wide support for many devices (AR, VR, iOS, Android, Windows and others). We thereby ensure reuse-ability of our code base. Performance hungry components like Audio/Video streaming in real-time are implemented as native libraries with hardware-acceleration, which becomes especially important on mobile platforms. Direct data-paths to such modules are used to reduce latency by surpassing Unity's `Cs` layer. MAUI is deployed as UWP (Universal Windows Application) for the HoloLens (Worker) and Windows10 on a desktop Computer (expert).

Dynamic libraries can be loaded at run-time to add new functionality *e.g.,* to enable platform specific features. Such native libraries can be scripted from `C#` and Javascript allowing not only direct calls but also asynchronous communication via callbacks and message passing. The `C#` and the native layer represent the core framework offering minimally defined functionality and interfaces for tasks like calls, data-transmission and remote UI-commands. Also support for native device functions like enabling the camera and turning on the flashlight.

4.2 Data Transmission

The foundation of remote assistance is the communication between involved parties. Time critical tasks require real-time capable communication. Given a somewhat capable up- and downlink, the software has to be powerful and efficient enough to ship the data in time to the network driver.

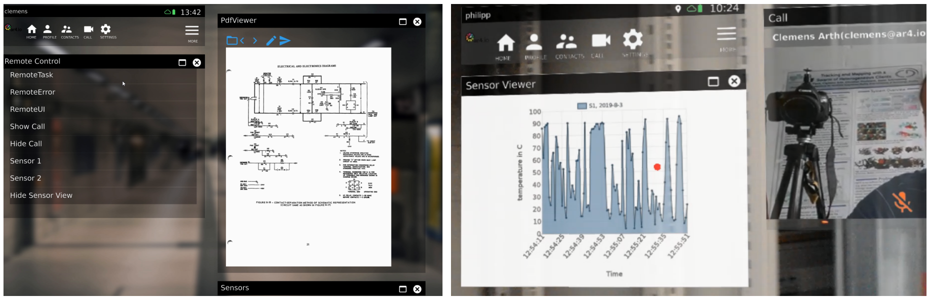

Fig. 2. Example of a remote maintenance demo application. (left) Expert interface with an open document and shortcuts of remote commands, (right) worker UI within a call (A/V, data, commands) and live view of telemetry data shown as a graph, as presented in our previous work [9].

Our communication module implements state-of-art `C++` techniques (multi-threaded, state-less, smart-pointer, single-allocations) to achieve such. We utilize a modified version of Raknet[5] library to interface the network stack. Experiments showed, especially on the HoloLens, bad performance of higher level implementations. Within our native layer we establish direct data-paths between the Audio/Video capturing, the hardware accelerated H.264 encoding and the communication library. Additionally we introduce a secondary data-channels between peers with soft real-time constrains to transmit non real-time related data. Coping with variable bandwidth, we can gradually control the data-flow rate attributes like video compression to optimise for the current connection characteristics.

4.3 Web based User-Interface in Unity

Our web-based UI is fully implemented in `Js`/`Html`/`Css`. It thereby allows us to use well established techniques and concept from the web community. Since Unity lacks an in-game `Html`-Engine, we use a modified `Html`-renderer and `Js`-interpreter [6] to bring web into Unity.

Unfortunately, we do not get full browser support which limits the use of some external libraries. Created UIs are rendered into texture maps displayed on any polygonal structure inside Unity. Responsive designers and applications can easily be created by web-developers and others following web-principles. Written applications (UI and business logic) for our framework are comparable to different web-sites each serving a dedicated purpose and user-loadable at any time. Fig. 2 shows the UI run on an desktop computer (left) and on the HoloLens (right). Icons are as well supported, as `Css` to create appealing themes. Furthermore, widgets are spawned using the data driven template library `json2html`[7].

[5] RakNet: `https://github.com/facebookarchive/RakNet`
[6] PowerUI: `https://powerui.kulestar.com`
[7] json2html: `http://json2html.com`

For ease of use we created a Js framework to directly bind functionality from our intermediate layer (C#) to Js-objects, allowing us *e.g.,* for code-autocompletion in VisualStudio for even faster prototyping. Features like an Audio/Video call are done simply by calling MAUI.Com.Call(id); in Js. Additionally the Js-interpreter allows to import namespaces at run-time allowing us to call Unity-related functions like spawning a cube:

```
var UE = importNameSpace("UnityEngine");
var go = UE.GameObject.CreatePrimitive(PrimitiveType.Cube);
```

Thus, combining the 2D and 3D world we allow Html-DOM-elements to interact with GameObjects. Still being web-centric we can use REST, AJAX and others to connect to any services and CPS.

4.4 Remote UI orchestration

A well known and often used practice in web-development is dynamically loading functionality from external sources, where *Lazy loading* is a common technique to load when needed. Utilizing the described Js-interpreter we implement such techniques in Unity, allowing us to hot-fetch code and data. Therefore transmitted data is either a audio/video stream or a message wrapped in JSON. In Js we can directly access the wrapped object and start using its contents. Commands are transmitted or received JSON objects. Since we can store Js-code in JSON objects, we can wrap data in a self-presenting manner *e.g.,* by shipping an visualization command together with an image. Since we can interpret the command at run-time we do not have to enforce a distinction between code and data at all.

Fundamental to commands is the generic web-based structure by using JSON objects. Our UI implementation allows us to append new JS code the current instance for later use. This is done either by adding <script> tags, by writing .js files which are linked within the DOM, by appending with jQuery.loadScript or by just adding it to a JS-runtime-object as function:

```
var receivedFct = "function(x) {return x+1;}";
var myFunc = new Function("return " + receivedFct)();
var res = myFunc(1); // res = 2
```

For single time use, we can run external code through eval() or create a function object for instant execution. In practice, the expert can alter, extend or replace the worker's UI. Thereby, functionality can be added ad-hoc and the UI can be controlled on a very granular level. The experts application provides shortcuts to common functions, like hide widgets, load an instruction application, bring back the worker's UI. For instance, an annotated captured frame of the video stream is send as ShowAnnotatedImage which translates to the following:

```
{
    "type":"CMD", "cmd":"ShowAnnotatedImage",
```

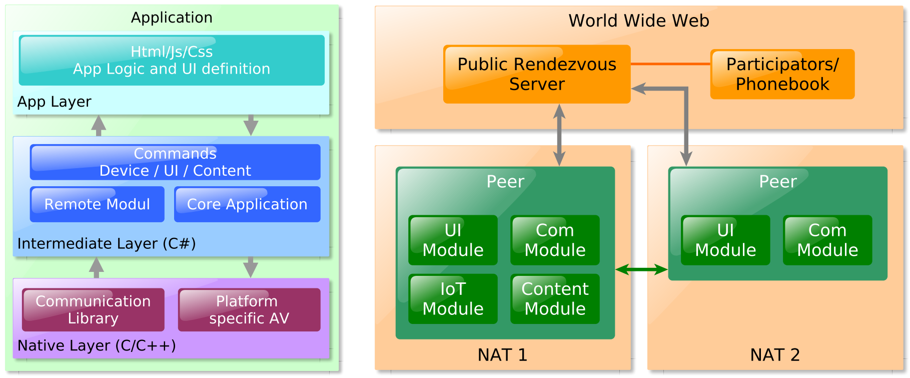

Fig. 3. *(left)* The application stack of MAUI. Data can be bi-directional passed up and down across software layers. The native layer *(purple)* holds hardware close modules for real-time communication and Audio-Video access. The intermediate layer *(blue)* holds wrappers for native libraries and modules for higher level functionality like special 3D Engine commands. The application layer *(turquoise)* holds all business logic and UI implementation. *(right) shows peer-to-peer and peer-2-server communication across the world wide web, where the server (orange) establishes the direct connection between peers (green).*

```
    "args":[ b64ImageData, AnnotationObject ]
}
```

A `cmd`-type message is consumed by `ShowAnnotatedImage`, a function defined in the `Js`-framework, which takes the `args` as arguments. Internally the function renders the image into a `<img>`-tag and draws an optional stroke-based annotations over it. Since we can vary the structure of the sent `JSON`, we can choose to be more specific or more general. Additionally, we can store any object for later using *e.g.,* like an email. Of course, this concepts allows to control the CPS directly from the expert side by calling available functions. In contrast to traditional remote desktop software, the expert does not have to navigate to a button and click it, instead he can directly activate its functionality.

In the current implementation, we categorize commands into three distinct categories: *UI Commands*, including command manipulating the UI like loading new content, where the experts replace the default-worker-UI (Fig. 2, right) with interactive step-by-step instructions; *Device Commands* including commands controlling the hardware like the RGB-camera; and *Content Commands*, including commands adding new contents like PDFs and images.

4.5 Backend

In order to keep track about users participating in a maintenance session, we use a phone-book similar to any modern real-time messaging application (Fig. 3, right). Furthermore, peer-to-peer calls are established between peers using RakNet's builtin functionality for NAT-Translation.

The beauty of our concept enabling hot-swapping of content unfolds in the management stage of AR, respectively UI content. We can simply host different UI versions, workflows and even application functionality remotely. These contents are downloaded by the clients when needed (*e.g.,* when new update are available, or the local cache runs out of date). This enables fully transparent and easily maintainable representations.

All components are included in Docker[8] containers, such that deployment of components on a distributed server infrastructure is straightforward.

5 EXTENDED USER STUDY

In our previous work [9], we performed a user study showing significant benefits when an experts takes over parts of the virtual interaction, freeing more mental capacities for physical tasks of the worker. In this work we extend the user-study to cover a larger diversity amongst participants from different occupations, including teachers and students in marketing, economics, psychology and other non-computer-science topics.

In the following we summarize our procedures to conduct the extended user-study and discuss our findings in detail. Following the target use-case (tele-assisted maintenance tasks), we analyze how a worker performs a multi-step procedure of repairing a CPS with and without the help of a remote expert. In particular, We focus on errors made under each condition to represent crucial real-world situations.

5.1 Evaluated conditions

Two conditions of interest where evaluated where the worker has to repair a broken CPS following step-by-step instructions: *SelfNavigation* and *ExpertHelp*.

SelfNavigation (SN) All interactions are done by worker himself. There is no external support whatsoever. The worker can decide on his/her own without any feedback or confirmation. As far as the UI of the worker allows it, the worker can use all available windows, but cannot alter them. The focus is on how well the worker can follow written instructions across the virtual and the physical world.

ExpertHelp (EH) While performing the task and working along the step-by-step instructions, the worker is connected to the remote expert with audio/video and can ask for support at any time. The expert gives advice and takes over interactions with the workers UI inside the HMD. Following, the worker can spent more time and attention to the physical task and neglect most of the UI-interactions

[8] Docker: `https://www.docker.com/`

Fig. 4. (left) User study equipment: faulty device (electronics box with light-bulb, with device id label on top, HoloLens and Bluetooth keyboard), (center) Power cord with labeled outlets Labeled outlets, (right) The worker wearing the HoloLens has completed the task, and the light-bulb turns green, as presented in our previous work [9].

in the virtual world. Furthermore, the expert can decide which information is presented to the worker. This can be a dedicated control window for a CPS or a new UI(Application) designed around single purpose *e.g.,* interactive step-by-step instruction. Reducing the amount of tasks of the worker, we expect to free (mental) capacities, which reflects in reduced time-to-start(as discussed in [9]) and reduced error-rates.

The extended user-study has the main focus on evaluating the process of task completion, in particular if any error occurred. Since workers are instructed to follow step-by-step instruction, occurring errors caused by lack of concentration can tell a lot about cognitive load and the stress level. In the real-world, such tasks are often mentally demanding, are conducted under high time pressure and often tolerate no errors (manufacturing, power plants, and others).

5.2 The repair task

We want to examine the impact of a shared virtual and physical task on a worker. Our theory is that the expert help in controlling the UI should reduce the worker's mental workload and cognitive effort while maintaining a machine. Our prototypical CPS (the faulty device) is represented by a smart light-bulb[9], which includes a virtual control interface. A software-module was added to the application stack to control a variety of smart-home appliances and to capture user-based telemetry data like click-events. We created one app for each condition along with a look-and-feel alike training app to make users comfortable with the system.

The task contains both physical and virtual actions. Physical actions contain unplugging and re-plugging a power cord, pressing buttons and typing on a keyboard. Virtual actions include searching for the device, controlling the device(light-bulb), interacting with the device and reading instructions. Fig. 4 depicts the faulty device (light-bulb in the box), the outlet and a participant successfully mastering the task. The task is divided into 5 stages:

[9] `https://www2.meethue.com/`

1. At first, the faulty device lights up red.
2. In both conditions (*SN* and *EH*), the worker has to open the instructions within the UI (a list with nine step-by-step instructions) for the task.
3. The worker has to open the "light-bulb" widget to control the smart light-bulb. In *SN*, the worker must do this by searching through the menu. In *EH*, the list is opened by the remote expert.
4. While following the instruction, the worker has to perform a sequence of operations: switching off power, unplugging, re-plugging, and switching power on again. Power outlets are labeled by ID, and the instructions refer to particular outlet IDs for the re-plugging step.
5. The worker has to press the start button in the device-interface widget of the UI. The task finishes when the device lights up green.

5.3 Experimental procedure

The conditions *SN* and *EH* were tested using a within-subject design. The tasks were alternated in the order of the conditions and between participants. Additionally the position of the power-plug was altered to further balance the study. Before starting the actual study, a quick training task was performed by each participant. The training was aimed to get used to the UI (same kind of widgets, but with different naming and contents) and to understand how to control the smart-light. No power-cord interaction where performed while training.

The study procedure included pre- and post-study questionnaires to elaborate on personal experience, the preferred method and subjective difficulty. Each participant was asked to comfortably put on the HoloLens and adjust it accordingly. Furthermore, they were put in the role of the worker instructed to repair the faulty device. At start the device lights up red and the app starts.

According to the description in our previous work, Fig. 4 depicts the faulty device and its device-id, the HoloLens with the bluetooth keyboard and the switch with the labeled outlets. A participant completing the task (green light) is shown on the far right of Fig. 4. A NASA TLX [17] questionnaire was conducted after each condition.

This user-study acts as an foundation to future studies planned in this field with MAUI. Therefore we are simply interested in the performance of the worker and his cognitive state. *Will the study follow our assumptions generated from conversations with our collaborators and their workers?*. We avoided using a volunteer expert in this first study, due to the big efforts of training, especially regarding consistency and the big unknown in operating the expert application. We decided to have an author play the expert role in a passive and consistent way, offering just as much help as needed to the worker. In *EH*, the expert performed the device search using the vocally communicated device-ID, pointing out to carefully read through the step-by-step instructions and ask if help was wanted.

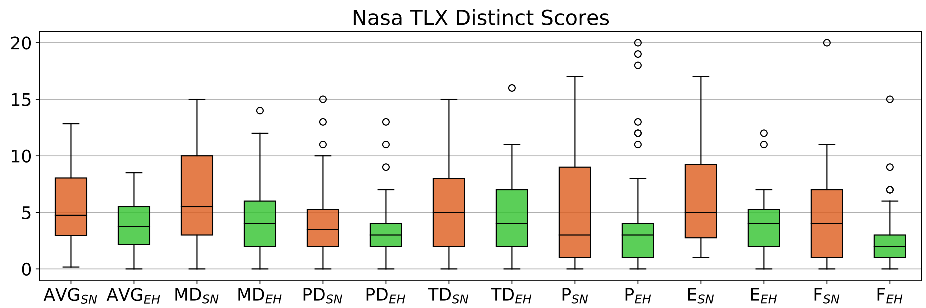

Fig. 5. A one-way ANOVA across all categories of the NASA TLX questionnaire show stastistical significance in the favor of **EH** in the categories **AVG**(Total Score, $p < 0.05$, $F_{1,86}, = 6.713$), **MD**(Mental Demand, $p < 0.05$, $F_{1,86}, = 6.712$), **E**(Effort, $p < 0.05$, $F_{1,86}, = 6.973$) and **B**(Frustration, $p < 0.05$, $F_{1,86}, = 5.947$). The categories PD(Physical Demand), TD(Temporal Demand) and P(Performance) showed not statistical difference whatsoever. (*Orange*) show distributions of *SN* and (*green*) distribution of *EH*.

5.4 Extended Results

We tested 44 participants (6f), aged 21-40 (avg 28.95, median 29). As shown in [9], the age distribution is below the current [10] (approx. ∼10 years), but is a representative cut through future generations. To check the technical fitness of participants, we specifically asked about the familiarity with PCs and step-by-step instructions. All participants performed step-by-step instructions and all of them use PCs on a regular basis. Half of the participants never used a HoloLens or a similar device before and a quarter never repaired or disassembled a PC. All participants found the training procedure suitable to get used of to the system.

A one-way repeated measures ANOVA was performed and we found main effects on NASA TLX scores supporting our hypothesis, *EH* being less mentally demanding and easier to operate over *SN*. We found statistical significance (significance level of 5%) towards *EH* in the overall average NASA TLX score ($p < 0.05$, $F_{1,86}, = 6.713$) and partially in its sub-scores Mental Demand (MD with $p < 0.05$,$F_{1,86}, = 6.712$), Effort (E, with $p < 0.05$, $F_{1,86}, = 6.712$) and Frustration (F with $p < 0.05$, $F_{1,86}, = 5.947$). No significance was found for Physical Demand (PD), Temporal Demand (TD) and Performance (P). However, our extend result supports the hypothesis of MAUI with *EH* relieving the worker while performing CPS heavy tasks. Furthermore, we see noteworthy improvements in a decreased Frustration and a reduced Effort, as well in a lower Mental demand. Fig. 5 depicts the distribution over all measured NASA TLX Category scores pairwise for *SN* (orange) and *EH* (green).

A more detailed look on the error-rate shown in Fig. 6 we find more often errors done while performing *SN* over *EH*. Since we alternate the order of *EH* and

[10] Workforce Age Distribution

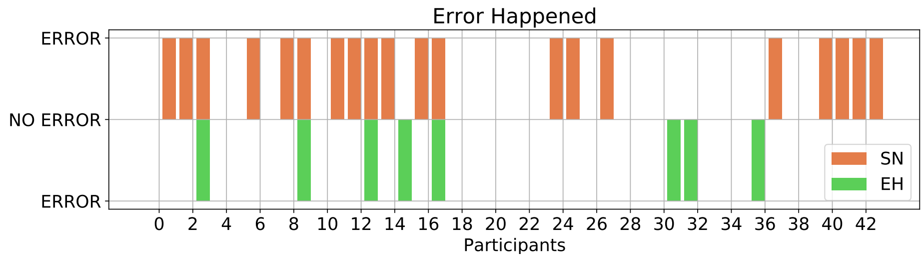

Fig. 6. Occurred errors over all 44 participants. The orange bars show occurred error while performing *SN* and green bars show errors when performed *EH*. Overall, *EH* shows far more occurred error compared to *SN*. No pattern was found correlated with which method was first performed.

SN, we can exclude the first condition being more error-prune than the second one. Throughout the whole study participants make more errors when performing *SN*. Common errors include, error due to inattentively reading, pressing the wrong start-sequence for the device in the UI and unplugging without turning off. This errors are not directly related to this specific scenario, but are common mistakes when operating machinery, which goes hand-in-hand with experiences from industry. Counting any occurred error results in 45% of the participants making an error in SH and 18% making an error in *EH*. This results coincide with our previous work, where me only evaluated critical errors (such as missing a step from the instruction) where 25% of the participant made an error while performing *SN* where only 15% errors happened in *EH*.

Questionnaire results At the end of the study, each participant was asked to answer selective and Likert-style questions about his or her perceived impressions. 90.6% preferred *EH* over *SN* when asked about which method was easier to operate. Following this trend, 86.36% found it easier to retrieve information needed to complete the task from the system in *EH*. In some cases participant preferred *SN* over *EH*. A possible reason is low trust in the expert combined with being overzealous. Such aspects have a bigger HR (Human Resources) aspect to it, especially in industries like manufacturing.

Likert-style questions at the end of the examination give further insights. Most of the participants where somewhat familiar with AR or had heard about it. We did not find any correlation between being familiar with AR and the performance of participants, however. Over 30 participants found it neutral to easy to navigate within the UI, concluding that the UI is not an obstacle at first-time use. Just 5 out of 44 participants found it difficult to find the device interface (the widget for controlling the device, showing up only when entering the correct device-ID into the device search). Fig. 7 depicts the distribution of the Likert question with trends towards easy/moderately familiar. Some of the participants did not read careful enough through the instructions and produced easily avoidable errors like wrong order or jump over an instruction. One par-

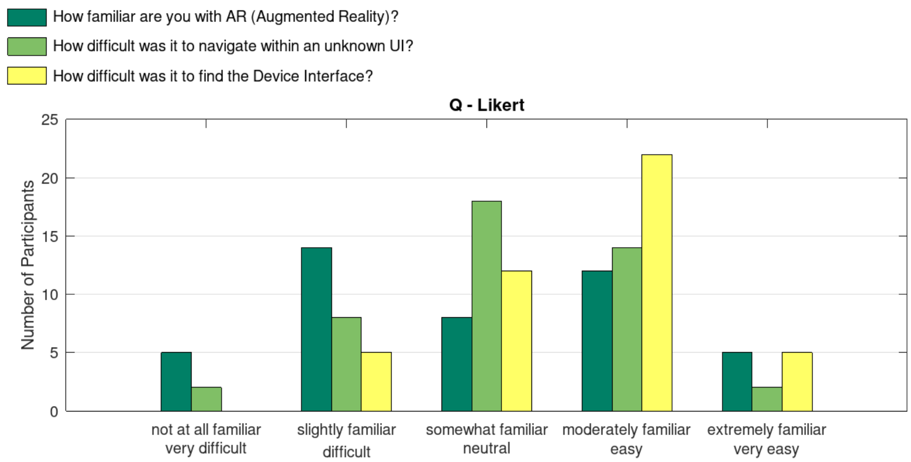

Fig. 7. Likert questionnaire results: Most of the participants where slightly familiar to moderately familiar with AR (dark green); Most of the participants found it neutral to easy to navigate an unknown UI (green); Most of the participants found i neutral to easy to find the device interface(yellow).

ticipant pointed out that he was *tired to look for an outlet device-ID* where it took him a while to find it. Such cases showed up in small numbers, but more often in *SN* slightly proving that they did not spend enough attention. Following results from [9], problems caused due to the lack of concentration are avoidable (*e.g.,* caused through tiredness) and the cost of expert support can be better justified.

User feedback It was the first time using a device like the Hololens for half of the participants. It turned out, that air-taping with the head-locked cursor (dot as crosshair) was not always well perceived, cumbersome and tiring to use. Our UI design was liked and text was clear to read. Their perceived behaviour was homogeneous and in most case their performance matched with what they believed. Better hand tracking [35] will further improve the experience and make it less cumbersome to use for participants. No errors occurred because of miss-clicking.

Further findings based on our observations show that smaller widgets are good for seated (static) use, but too small for standing experiences. Clutter within the narrow field of view of the HoloLens is always tricky to handle. In MAUI the expert can take care of that by *e.g.,* turning of unused parts of the UI or by locking the UI to the environment.

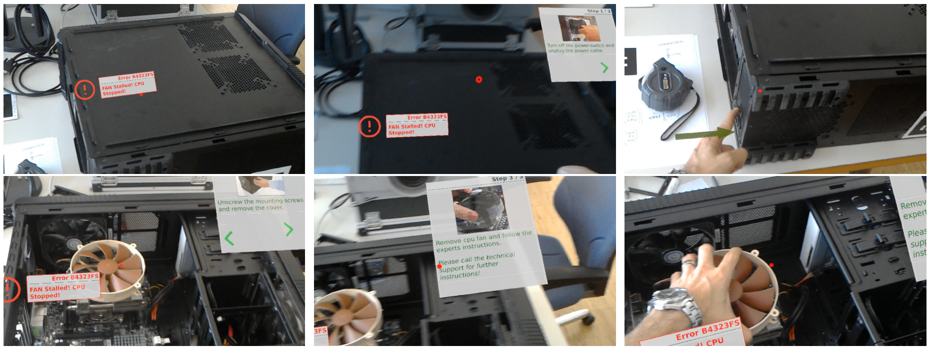

Fig. 8. Workers view of an PC repair: Error reported by the CPS (*top-left*) and step-by-step instruction are enabled by the expert (*top-center*). Step-1: Turn off the power (*top-right*) and remove the case-cover (Step-2, *bottom-left*); Step-3: Remove the fan under instructions of the expert (*bottom-center*) and (*bottom-right*).

6 BUILDING A MOBILE UI

In our previous work [9] we performed a qualitative developer user-study about the ease of use of the presented framework and to assess how easy it is for a new developer to extend and write new features or applications.

6.1 Use Case

A worker faces an issue with a personal computer, which fails to boot. Therefore, the worker uses an AR-enabled device (in our case a HoloLens) and calls the expert by selecting the experts ID in the phonebook. First, an audio-video call is established. The expert can perceive the workers outfacing camera view, while the worker sees a live image of the expert.

After identifying the malfunctioning PC, the expert is able to obtain a diagnostic message from the PC over the network: "the fan is stalled". The expert raises the error to the worker by selecting the corresponding command in the remote commands menu. The remote command is shipped to the worker and displayed. The worker, however, is unfamiliar with this error and not really knowledgeable in disassembly of personal computers, so he asks for help and further guidance.

The expert in turn replaces the error message with step-by-step-instructions, depicted in Fig. 8. While going through the steps the worker requires more detailed instructions. To better describe the actual location (*i.e.,* transfer spatial understanding from the expert side to the worker), the expert decides to take a screenshot of the worker's view and to annotate the important areas where the fan-mounts are placed. By pressing the *Send* button, the annotation is sent and displayed in the workers view.

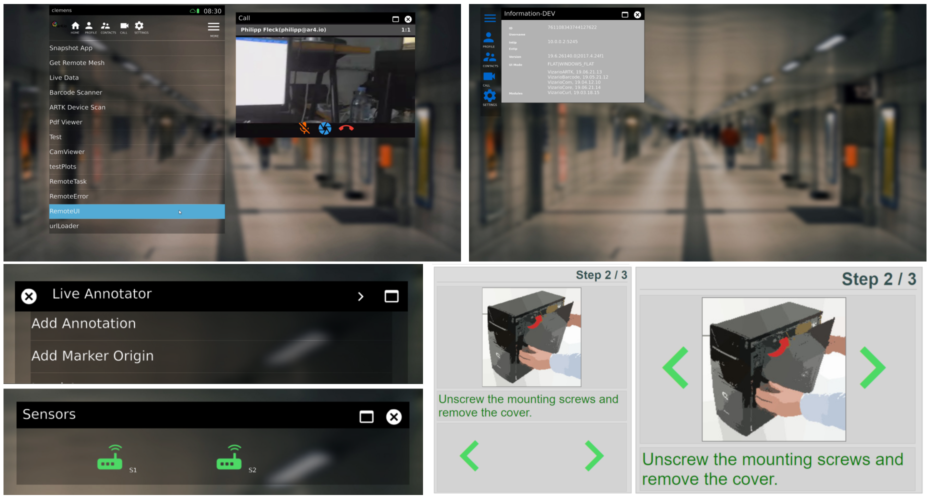

Fig. 9. Top: Comparison of the default expert UI layout (left) *vs.*A's improved layout (right, as shown in [9]). A's version shows a better use of screen-space. Bottom: B's improved layouts vs the default ones. *Close* separated from *maximize* (left) and a more ergonomic placement of the controls (right) to increase usability.

6.2 Developers

Each of both developers (A and B) was instructed to use our UI-framework to improve or modify the application of repairing a stuck fan in a PC (see Fig. 8). This application follows the same principles (UI exchangeable and controllable by the expert, step-by-step instructions, A/V connectivity), but with a more tangible use case.

Both had a web-development background in creating web-applications (`Html`, `Css`, `Js`). No inputs where given on what task to perform at any time and both participants had free choice. The procedure consisted of a one hour workshop as introduction into the framework followed by three hours of development. Due to the lack of an exhaustive documentation, one author answered typical documentation questions like *Where can I find this function?* Both were in there 30's and work in a technical field.

Changes by A: A chose to improve the layout of the expert application due to the bad usage of available screen space. Fig. 9 top left depicts our version of the expert's application with a two-column design, smaller content widgets and navigation-bar in the top section on the left and A's improved version on the top right. By re-arranging the navigation-bar vertically to the left hand side of the screen, it allows us to have floating widgets in different sizes unbound from the two column design.

Changes by B: B chose to tackle usability issues found. One being *close* and *maximize* being next to each in the Workers UI (Fig. 9, bottom-left) and the

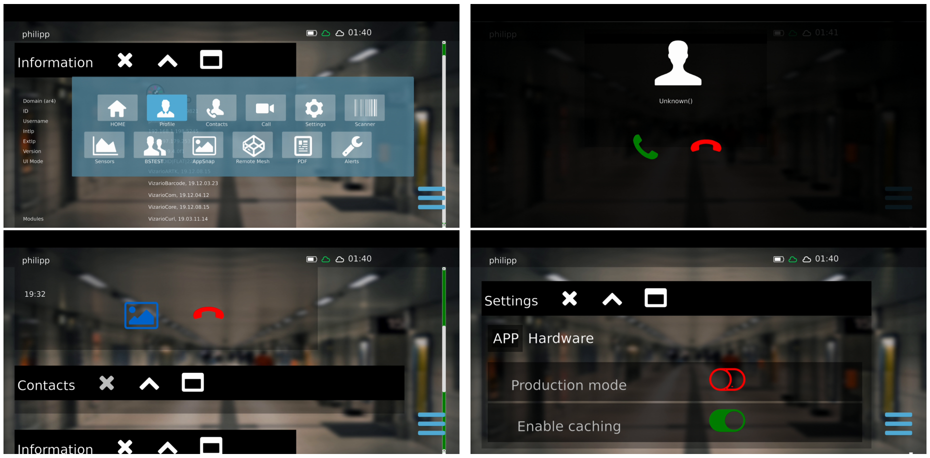

Fig. 10. Mobile UI based on the feedback and outcome of the user-studies. (top-left) main-menu for various functions; (top-right) Incoming call screen; (bottom-left) active widgets within the scroll-able ticker-style view; (bottom-right) the settings widgets with tabs and toggles.

other being the placement of the controls in the step-by-step instructions. By moving *close* to the far left and keeping *maximize* on the right B could increase usability for the worker. Furthermore he updated the widget template to apply the change to all widgets. Step-by-step instructions were improved by moving the controls left and right of the center part instead of heaving them below (Fig. 9, bottom right). This improved aesthetics and the use of space, allowing for more uncluttered content. Slight improvements like this have an big impact in usability, especially the first one can reduce frustration when using the application.

Both participants were able to produce reasonable results in the given timeframe and even surpassed our expectations. The overall feedback was that it was easy to use with some knowledge on web-technologies. They could use the framework and its provided functions and templates to produce reasonable results. The assessed NASA TLX questionnaires showed little Frustration for B and a higher level for A, correlating with the bigger modifications. Both had a good estimate of their performance, even if the authors would judge them less self critical.

6.3 Handheld AR

The promising outcome of the developer study and the feedback of the user-study let us to rethink concepts of the UI for mobile handheld use. Therefore we ported our application to mobile phones and tablets, implementing a new mobile UI for Android and iOS use, respectively.

Fig. 10 depicts the main-menu, an incoming call, the ticker view with multiple widgets open and the settings widget. One of the main reasons to reconsider handheld usage is the inherent request for this by industries, allowing for mobile use at-sight. We increased the icon-size and implemented a scrolling widgets approach (ticker-view), allowing widgets to pop-up either at the end or at the beginning of the ticker-view. Incoming calls and messages are shown in full-screen and need to be acknowledged.

The core changes include modifying the web templates (`JSON`-transforms for `json2html`), modifying `Css`-stylesheets for small screen use and adding. Those changes did not affect the application-framework and only utilized our `Js`-framework. Icon size where fit to be finger-friendly and font-sizes adapt to the actual handheld device. Again, the beauty of our UI management approach fully unfolds, as minor changes are sufficient to move between head-worn and handheld devices.

7 DISCUSSION AND CONCLUSION

Easy-to-adapt and *easy-to-extend* are key aspect in the industrial world. A hand full of web-developers are always part of development teams at factories to realize custom in-house solutions. Our extendable approach allows such teams to add their content and business logic and to fully integrate into existing infrastructure.

We have extended our previous work MAUI, where we introduce an new mobile UI implemented by only using our `Js`-framework, an extended user-study showing that repairing a CPS is more error-prone when a worker has to handle all virtual and physical tasks and we given important details about our implementation and show how easy it is to extend and modify MAUI. Furthermore, applications can easily be hot-loaded and live-patched allowing for continuous deploy- and development.

The extended quantitative user-study covered more participants from non-technical professions like teaching. The majority of the participants preferred EH over SN, which is further supported by less errors made in EH. NASA TLX shows a higher demand when operating alone all physical and virtual tasks (SN).

Results of the developer study and inputs were used to create a mobile version of MAUI capable of running on Android and iOS. Insights on how to use our framework have been described, making it easy to create new applications. The developer study shows that developers with basic web knowledge can perform meaningful changes to existing applications and, given enough time, could rather easily create whole applications themselves. Furthermore, all modifications can be implemented using web-technologies and do not require special knowledge in Unity, `C#`, `C++` or `compilation` which was also observed by MacIntyre *et al.* [27]. Furthermore, we showed how to use `Js` running inside Unity to write code which interacts with the 3D world of Unity.

In the future, we plan run further tests in production facilities to gather more insights and further improve MAUI. Additionally, UI experts can further improve the usability and support improved user-interfaces with user-studies.

We can further improve the widget-placement and behaviour to allow for bigger flexibility of the worker and the expert. A user interface management to improve continuous delivery will further smooth the process.

ACKNOWLEDGEMENTS

The authors wish to thank Denis Kalkofen. This work was supported by FFG grant 859208.

References

1. Alce, G., Roszko, M., Edlund, H., Olsson, S., Svedberg, J., Wallergård, M.: [poster] ar as a user interface for the internet of things - comparing three interaction models. In: ISMAR-adj. pp. 81–86 (Oct 2017). https://doi.org/10.1109/ISMAR-Adjunct.2017.37
2. Barakonyi, I., Fahmy, T., Schmalstieg, D.: Remote collaboration using augmented reality videoconferencing. In: Proc. of Graphics Interface (GI). pp. 89–96. Canadian Human-Computer Comm. Society (2004), `http://dl.acm.org/citation.cfm?id=1006058.1006070`
3. Bauer, M., Kortuem, G., Segall, Z.: "where are you pointing at?" a study of remote collab. in a wearable videoconf. system. In: ISWC. pp. 151–158 (Oct 1999). https://doi.org/10.1109/ISWC.1999.806696
4. Baumeister, J., Ssin, S.Y., ElSayed, N.A.M., Dorrian, J., Webb, D.P., Walsh, J.A., Simon, T.M., Irlitti, A., Smith, R.T., Kohler, M., Thomas, B.H.: Cognitive cost of using augmented reality displays. TVCG **23**(11), 2378–2388 (Nov 2017). https://doi.org/10.1109/TVCG.2017.2735098
5. Chastine, J.W., Nagel, K., Zhu, Y., Hudachek-Buswell, M.: Studies on the effectiveness of virtual pointers in collaborative augmented reality. 3DUI pp. 117–124 (2008)
6. Chen, S., Chen, M., Kunz, A., Yantaç, A.E., Bergmark, M., Sundin, A., Fjeld, M.: Semarbeta: mobile sketch-gesture-video remote support for car drivers. In: Augmented Human International Conference (AH) (2013)
7. Ens, B., Hincapié-Ramos, J.D., Irani, P.: Ethereal planes: A design framework for 2d information spaces in 3d mixed reality environm. In: SUI. ACM (2014)
8. Feiner, S., MacIntyre, B., Haupt, M., Solomon, E.: Windows on the world: 2d windows for 3d augmented reality. In: UIST. pp. 145–155. ACM (1993). https://doi.org/10.1145/168642.168657, `http://doi.acm.org/10.1145/168642.168657`
9. Fleck, P., Reyes-Aviles, F., Pirchheim, C., Arth, C., Schmalstieg, D.: Maui: Tele-assistance for maintenance of cyber-physical systems. In: VISIGRAPP 2020 - Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (2020). https://doi.org/10.5220/0009093708000812
10. Funk, M., Bächler, A., Bächler, L., Kosch, T., Heidenreich, T., Schmidt, A.: Working with ar?: A long-term analysis of in-situ instructions at the assembly workplace. In: PETRA. pp. 222–229. ACM (2017). https://doi.org/10.1145/3056540.3056548, `http://doi.acm.org/10.1145/3056540.3056548`

11. Funk, M., Kosch, T., Schmidt, A.: Interactive worker assistance: Comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions. In: UBICOMP. pp. 934–939. ACM (2016). https://doi.org/10.1145/2971648.2971706, http://doi.acm.org/10.1145/2971648.2971706
12. Fussell, S.R., Setlock, L.D., Yang, J., Ou, J., Mauer, E., Kramer, A.D.I.: Gestures over video streams to support remote collaboration on physical tasks. Hum.-Comput. Interact. **19**(3), 273–309 (Sep 2004)
13. Gauglitz, S., Nuernberger, B., Turk, M., Höllerer, T.: In touch with the remote world: Remote collaboration with augmented reality drawings and virtual navigation. In: VRST. pp. 197–205. ACM (2014). https://doi.org/10.1145/2671015.2671016, http://doi.acm.org/10.1145/2671015.2671016
14. Gauglitz, S., Nuernberger, B., Turk, M., Höllerer, T.: World-stabilized annotations and virtual scene navigation for remote collaboration. In: UIST. pp. 449–459. ACM, New York, NY, USA (2014). https://doi.org/10.1145/2642918.2647372, http://doi.acm.org/10.1145/2642918.2647372
15. Gurevich, P., Lanir, J., Cohen, B., Stone, R.: Teleadvisor: A versatile augmented reality tool for remote assistance. In: CHI. pp. 619–622. ACM (2012). https://doi.org/10.1145/2207676.2207763, http://doi.acm.org/10.1145/2207676.2207763
16. Han, F., Liu, J., Hoff, W., Zhang, H.: [poster] planning-based workflow modeling for ar-enabled automated task guidance. In: ISMAR-adj. pp. 58–62 (Oct 2017). https://doi.org/10.1109/ISMAR-Adjunct.2017.32
17. Hart, S.G., Staveland, L.E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. Human mental workload **1**(3), 139–183 (1988)
18. Henderson, S., Feiner, S.: Exploring the benefits of augmented reality documentation for maintenance and repair. TVCG **17**(10), 1355–1368 (Oct 2011). https://doi.org/10.1109/TVCG.2010.245
19. Huang, W., Alem, L.: Handsinair: A wearable system for remote collaboration on physical tasks. In: CSCW. pp. 153–156. ACM (2013). https://doi.org/10.1145/2441955.2441994, http://doi.acm.org/10.1145/2441955.2441994
20. Kim, S., Lee, G.A., Sakata, N.: Comparing pointing and drawing for remote collaboration. In: ISMAR. pp. 1–6 (Oct 2013). https://doi.org/10.1109/ISMAR.2013.6671833
21. Kirk, D., Stanton Fraser, D.: Comparing remote gesture technologies for supporting collaborative physical tasks. In: CHI. pp. 1191–1200. ACM (2006). https://doi.org/10.1145/1124772.1124951, http://doi.acm.org/10.1145/1124772.1124951
22. Kollatsch, C., Schumann, M., Klimant, P., Lorenz, M.: [poster] industrial augmented reality: Transferring a numerical control connected augmented realty system from marketing to maintenance. In: ISMAR-adj. pp. 39–41 (Oct 2017). https://doi.org/10.1109/ISMAR-Adjunct.2017.27
23. Krueger, M.W., Gionfriddo, T., Hinrichsen, K.: Videoplace - an artificial reality. In: CHI. pp. 35–40. ACM, New York, NY, USA (1985). https://doi.org/10.1145/317456.317463, http://doi.acm.org/10.1145/317456.317463
24. Kurata, T., Sakata, N., Kourogi, M., Kuzuoka, H., Billinghurst, M.: Remote collaboration using a shoulder-worn active camera/laser. In: ISWC. vol. 1, pp. 62–69 (Oct 2004). https://doi.org/10.1109/ISWC.2004.37

25. Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., Mitsuishi, M.: Gestureman: A mobile robot that embodies a remote instructor's actions. In: CSCW. pp. 155–162. ACM, New York, NY, USA (2000). https://doi.org/10.1145/358916.358986, `http://doi.acm.org/10.1145/358916.358986`
26. Lee, T., Hollerer, T.: Viewpoint stabilization for live collaborative video augmentations. In: ISMAR. pp. 241–242. IEEE Computer Society (2006). https://doi.org/10.1109/ISMAR.2006.297824, `https://doi.org/10.1109/ISMAR.2006.297824`
27. MacIntyre, B., Hill, A., Rouzati, H., Gandy, M., Davidson, B.: The argon ar web browser and standards-based ar application environment. In: ISMAR. pp. 65–74 (Oct 2011). https://doi.org/10.1109/ISMAR.2011.6092371
28. Oda, O., Sukan, M., Feiner, S., Tversky, B.: Poster: 3d referencing for remote task assistance in augmented reality. In: 3DUI. pp. 179–180 (March 2013). https://doi.org/10.1109/3DUI.2013.6550237
29. O'Neill, J., Castellani, S., Roulland, F., Hairon, N., Juliano, C., Dai, L.: From ethnographic study to mixed reality: A remote collaborative troubleshooting system. In: CSCW. pp. 225–234. ACM (2011). https://doi.org/10.1145/1958824.1958859, `http://doi.acm.org/10.1145/1958824.1958859`
30. Ou, J., Fussell, S.R., Chen, X., Setlock, L.D., Yang, J.: Gestural communication over video stream: Supporting multimodal interaction for remote collaborative physical tasks. In: ICMI. pp. 242–249. ACM, New York, NY, USA (2003). https://doi.org/10.1145/958432.958477, `http://doi.acm.org/10.1145/958432.958477`
31. Rambach, J., Pagani, A., Stricker, D.: [poster] augmented things: Enhancing ar applications leveraging the internet of things and universal 3d object tracking. In: ISMAR-adj. pp. 103–108 (Oct 2017). https://doi.org/10.1109/ISMAR-Adjunct.2017.42
32. Roulland, F., Castellani, S., Valobra, P., Ciriza, V., O'Neill, J., Deng, Y.: Mixed reality for supporting office devices troubleshooting. In: VR. pp. 175–178 (March 2011). https://doi.org/10.1109/VR.2011.5759458
33. Tzimas, E., Vosniakos, G.C., Matsas, E.: Machine tool setup instructions in the smart factory using augmented reality: a system construction perspective. IJIDeM (Mar 2018). https://doi.org/10.1007/s12008-018-0470-z, `https://doi.org/10.1007/s12008-018-0470-z`
34. Wellner, P., Freemann, S.: The Double DigitalDesk: Shared editing of paper documents. Tech. Rep. EPC-93-108, Xerox Research (1993)
35. Xiao, R., Schwarz, J., Throm, N., Wilson, A.D., Benko, H.: Mrtouch: Adding touch input to head-mounted mixed reality. TVCG **24**(4), 1653–1660 (April 2018). https://doi.org/10.1109/TVCG.2018.2794222, `doi.ieeecomputersociety.org/10.1109/TVCG.2018.2794222`