Virtual Reality (2025) 29:73
https://doi.org/10.1007/s10055-025-01149-6

ORIGINAL ARTICLE

# ARTHUR: authoring human–robot collaboration processes with augmented reality using hybrid user interfaces

Rasmus Lunding[1] · Sebastian Hubenschmid[2] · Tiare Feuchtner[1,2] · Kaj Grønbæk[1]

Received: 16 August 2024 / Accepted: 9 April 2025



## Abstract

While augmented reality shows promise for supporting human–robot collaboration, creating such interactive systems still poses great challenges. Addressing this, we introduce ARTHUR, an open-source authoring tool for augmented reality-supported human–robot collaboration. We thereby propose to combine desktop interfaces and touchscreen devices for effective authoring with head-mounted displays for testing and in-situ refinements. ARTHUR supports 20 types of multimodal *feedback* to convey robot, task, and system state, 10 *actions* that enable the user to control the robot and system, and 18 *conditions* for *feedback* customization and triggering of *actions*. By combining these elements, users can create interaction spaces, controls, and information visualizations in augmented reality for collaboration with robot arms. To demonstrate the general applicability of ARTHUR for human–robot collaboration scenarios, we replicate representative examples from prior work. Further, in an evaluation with five domain-savvy participants, we reflect on the usefulness of our hybrid user interface approach and the supported functionality, highlighting its potential and directions for future work.


**Keywords** Human–robot collaboration · Human–robot interaction · Augmented reality · Authoring · Hybrid user interface

## 1 Introduction

Recent research in human–robot interaction (HRI) has demonstrated the benefits of collaborative robots (cobots) for assisting with assembly tasks (Lunding et al. 2023; Malik and Pandey 2022), as they enable close collaboration between humans and robots without extensive safeguards (Cheon et al. 2022). However, despite great interest and desire to adopt these new technologies, the manufacturing industry struggles to integrate collaborative robots into their processes for various reasons, such as lacking knowledge and safety concerns. From our industry partners in the MADE FAST project[1] we have observed that when cobots are introduced they are rarely collaborated with, as the human worker only steps in to troubleshoot or takes over parts of the task within a separate workspace in turn-taking fashion. To overcome such limitations and make “genuine” human–robot collaboration (HRC) possible, the operator must be made aware of the ongoing and planned robot procedures and be able to coordinate these. Information presented in augmented reality (AR) can be beneficial for conveying the robot’s intent (Pascher et al. 2023; Suzuki et al. 2022), visualizing safety information (Ganesan et al. 2018; Hietanen et al. 2020), and highlighting the procedures or tasks which the user has to do (Dimitropoulos et al. 2021; Lunding et al. 2023). Researchers have investigated the potential of AR for supporting collaboration (Ganesan et al. 2018; Rosen et al. 2019), thereby exploring the suitability of different devices (Hietanen et al. 2020; Rosen et al. 2019) and individual visualization types (Arevalo et al. 2021; Cogurcu

✉ Rasmus Lunding
rsl@cs.au.dk

Sebastian Hubenschmid
sebastian.hubenschmid@uni-konstanz.de

Tiare Feuchtner
tiare.feuchtner@uni-konstanz.de

Kaj Grønbæk
kgronbak@cs.au.dk

1 Department of Computer Science, Aarhus University, Aarhus, Denmark

2 Department of Computer and Information Science, University of Konstanz, Konstanz, Germany

1 MADE FAST: https://www.made.dk/en/made-fast/.

Published online: 24 April 2025

and Maddock 2023; Gruenefeld et al. 2020), or proposing systems for solving specific assembly tasks (Andronas et al. 2021; Ganesan et al. 2018).

Notably, only few researchers have so far explored the usefulness of visualization combinations, noting drawbacks such as redundancy (e.g., convey planned robot movement through a path visualization and holograms of task-relevant objects) (Lunding et al. 2023; Rosenholtz et al. 2007) and the need for dynamic visualizations that show up only when relevant (Lunding et al. 2024). We argue that, in addition to investigating static visualizations in isolation (e.g., Cleaver et al. 2021; Cogurcu and Maddock 2023; Rosen et al. 2019), we must evaluate visualizations within the context of the overall system and throughout representative workflows, to better understand the users' needs. However, the process of designing, testing, debugging, and refining AR content is often tedious and challenging: AR setups for HRI are typically authored on a desktop system, requiring accurate simulation of the real-world environment in which the virtual content should be anchored. Even then, there is often a notable discrepancy between the simulation of AR content and its situated visualization upon deployment on the AR head-mounted display (HMD).

To break out of the sequential design-deploy-refine process that requires time-consuming alternation between the personal computer (PC) and the HMD, we have recently demonstrated the potential of authoring visualizations directly in AR (i.e., *in-situ*) (Lunding et al. 2024). Users can thereby immediately test out variations of visualizations and their appearance properties, without needing to recompile and deploy the application. On the other hand, in-situ authoring in AR can limit the user's effectiveness compared to desktop-based interaction: mid-air text entry on a virtual keyboard is inferior to physical typing (Grubert et al. 2023), and manipulation of traditional interface elements (e.g., buttons and lists) can be strenuous and difficult due to imprecise mid-air interaction (Chan et al. 2010). Recent works have therefore argued for combining multiple devices as complementary interfaces (Elmqvist 2023; Zagermann et al. 2022) (e.g., AR HMDs and mobile touchscreen devices) to leverage the advantages of each technology. In this paper, we propose adopting this approach for HRC.

Addressing the aforementioned challenges, we present ARTHUR, an AR-based **A**utho**r**ing **T**ool for **Hu**man–**R**obot collaboration scenarios that supports the creative workflow through a hybrid user interface (i.e., across a desktop, tablet, and HMD). We expand upon prior work by facilitating in-situ authoring not only of visualizations (*feedback*) but also of user *actions* and *conditions*, thus creating a holistic AR environment for designing HRC processes. Besides enabling the general setup and in-situ authoring, our proposed system fluidly facilitates switching to the operation phase, allowing users to instantaneously try out their authored workflows on one or more robots (Fig. 1). We demonstrate the potential of ARTHUR by replicating a variety of scenarios from prior work. Further, we evaluated the usefulness of the supported features and the suitability of our hybrid user interface approach in a usage evaluation with experts in AR system design and development.

In summary, we contribute (1) an AR authoring tool for HRC based on three types of design components (*feedback*, *actions*, *conditions*) that supports (2) a flexible authoring workflow through a hybrid user interface. Further, we highlight opportunities and challenges in authoring HRC systems with a hybrid user interface approach through (3) demonstrations and an expert evaluation using ARTHUR. The source code[2] and supplementary videos[3] are available online.

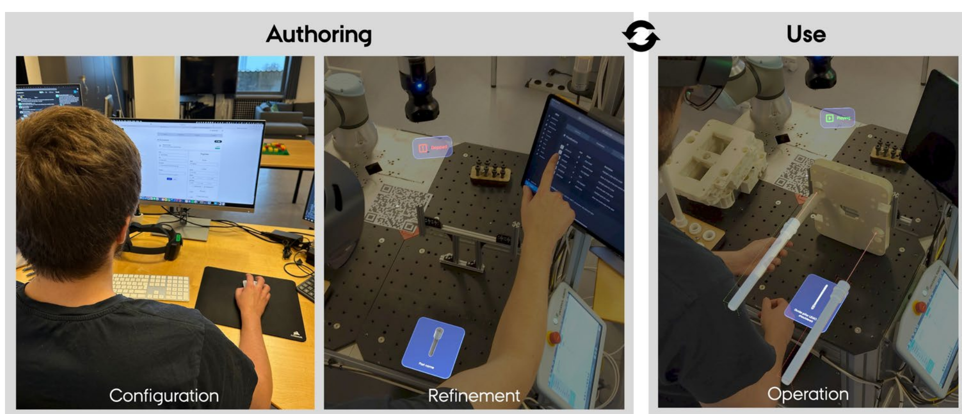


**Fig. 1** ARTHUR is an authoring tool for augmented reality-supported human–robot collaboration. It supports (1) creating the initial system configuration in the web interface (PC) (left), (2) refining the setup in-situ using a hybrid of web interface (tablet) and AR interface (HMD) (center), and (3) testing and using the authored system on the AR interface (HMD) (right)

[2] ARTHUR project repository: https://gitlab.au.dk/arthur.

[3] ARTHUR video playlist: https://www.youtube.com/playlist?list=PLhjFAueqW0cuRR-ZruATwfFXfasdi5Gli.

## 2 Related work

We review existing approaches for *AR authoring for HRI*, discuss *design elements* for AR-supported HRI, and review the concept of *hybrid user interfaces*.

### 2.1 AR authoring for HRI

Recent surveys (Ens et al. 2019; Sereno et al. 2020; Fidalgo et al. 2023; Ratcliffe et al. 2021; Marques et al. 2022, 2024) show that collaborative mixed reality systems (e.g., for HRI) are now mature enough to "focus deeply on the nuances of supporting collaboration" (Ens et al. 2019), such as the use of heterogeneous hardware (Sereno et al. 2020), moving between reality and AR during collaborative tasks (Fidalgo et al. 2023), or the potential for in-depth data collection (Ratcliffe et al. 2021). In the context of authoring for HRI, AR environments have primarily been explored to assist non-technical users in defining the robots' behavior. Noteworthy authoring tools include GhostAR (Cao et al. 2019a), V.Ra (Cao et al. 2019b), KineticAR (Fuste et al. 2020), and PRogramAR (Ikeda and Szafir 2024). However, these are concerned with robot programming and aspects related to that, but not the authoring of the AR content that supports HRI itself. For authoring AR content, Microsoft Dynamics 365 Guides (Microsoft 2025) is an interesting commercial tool that allows users to author assembly instructions. However, it does not allow for integration with robots and is therefore limited to instructions without input to and output from the robot. A third category of relevancy are tools for data visualization or debugging, such as ARViz (Hoang et al. 2022; Ikeda and Szafir 2022). These often provide visualization functionality that is customized towards developers and do not necessarily address usability challenges, e.g., visual clutter and supporting user input. In this context, recent works (Lee et al. 2023; Martins et al. 2022) highlight how such situated visualizations can be integrated into a working environment to support decision-making.

We partially addressed these shortcomings in our prior work on RoboVisAR (Lunding et al. 2024): an AR authoring tool that enables users to create situated visualizations of robot data (e.g., status and movement path). It employs a timeline-based approach where a recording is made by executing the robot program before the authoring process begins, as was also proposed in prior work (Leiva et al. 2021). AR content is then designed based on this recording before deployment for live execution, while interaction and feedback are handled entirely through an AR-HMD. In our view, RoboVisAR still has three key limitations for being used in a broader context, e.g., collaborative assembly, as it is: (1) restricted to robot visualizations, thus excluding visualization of additional/external content, such as assembly instructions; (2) user input is not possible while the system is running (i.e., unidirectional instead of bidirectional information flow from robot to operator); and (3) it suffers from challenges of mid-air interaction, in particular when manipulating 2D user interface elements during authoring (Lunding et al. 2024, 2023). We aim to address these limitations with ARTHUR, which supports assembly instructions, robot visualizations, input from the user to the system, and a hybrid authoring interface to limit virtual menu interaction. Combined, these properties should allow users to author the most relevant aspects of AR user interfaces for collaborative assembly.

### 2.2 Design components for AR-supported HRI

Recent reviews reveal that a broad range of visual design components exist not only within AR-supported human-centered collaboration (Ghamandi et al. 2023; Marques et al. 2022), but also AR-supported HRI (Suzuki et al. 2022; Walker et al. 2023). Suzuki et al. (2022) identify three groups of design components: *UIs and widget*, *spatial references and visualizations*, and *embedded visual effects*, each consisting of further subcategories. Some of these components are highly relevant for collaborative assembly: points and locations (Chan et al. 2020), paths and trajectories (Andronas et al. 2021; Lunding et al. 2023), areas and boundaries (Ganesan et al. 2018; Hietanen et al. 2020), information panels (Lunding et al. 2023), and labels and annotations (Andersen et al. 2016). The virtual design element taxonomy by Walker et al. (2023) provides further classification that can be used for HRI in mixed reality and also includes task-related design elements. Prior work by Li et al. (2019) breaks down visualizing relevant information regarding tasks, parts, tools, and processes by describing how abstract representations (e.g., arrows) can be integrated with 3D models. These review papers informed the *feedback* types implemented by ARTHUR.

Besides feedback for the user, Suzuki et al. (2022) categorize different levels of interactivity, ranging from *only output*, to *implicit*, *explicit and indirect*, and *explicit and direct*. With ARTHUR we aim is to support all these levels, such that the users can choose whatever their scenario requires. Suzuki et al. (2022) further categorize interaction modalities and techniques, gaze, gesture, and proximity, most of which are supported in ARTHUR.

### 2.3 Hybrid user interfaces

Hybrid user interfaces employ cross-device interaction (Brudy et al. 2019) to combine "heterogeneous display and interaction device technologies" (Feiner and Shamash 1991), such as using AR HMDs simultaneously with smartphones, tablets, or desktop systems. The potential of AR in HRI makes this combination especially compelling, as commonly-used devices (e.g., tablets) can be seamlessly extended with superimposed content (e.g., Hubenschmid et al. 2023; Langner et al. 2021; Reipschläger et al. 2021). In addition, hybrid user interfaces have been found to facilitate better performance for two-dimensional input such as text entry (Grubert et al. 2023) and navigation (Büschel et al. 2019; Wieland et al. 2024), likely due to high familiarity (Butscher et al. 2018; Hubenschmid et al. 2021a), high input accuracy, and the availability of haptic feedback (Knierim et al. 2021). In the context of authoring and deploying AR systems, recent work has explored the asynchronous use of hybrid user interfaces (Hubenschmid et al. 2021b), involving switching between an authoring environment on one device (e.g., a desktop computer) and content inspection on another (e.g., AR-HMD): For example, Hubenschmid et al. (2022) combined a familiar 2D desktop interface for visual analytics with immersive virtual reality, to allow traditional data visualization on a 2D screen, as well as immersive visualizations in-situ. The user can thereby flexibly switch to the appropriate interface (i.e., 2D ex-situ or 3D in-situ), as the tasks demand. This approach may also prove beneficial for authoring HRC processes, allowing users to switch from a familiar setup on a desktop for the initial programming to an in-situ approach for inspection and fine adjustments of the content.

However, prior studies on such distributed systems also highlight disadvantages, such as increased cognitive load due to repeated visual attention switches or context switches (Kim and Dey 2009; Rogers and Monsell 1995), for example when switching between displays (Rashid et al. 2012a, b) or content on an AR-HMD and physical screen (Hubenschmid et al. 2023; Grubert et al. 2015; Normand et al. 2018). Although context switches are not the primary focus of our work, we carefully designed ARTHUR to reduce the cost of switching between interfaces during the different authoring phases that involve a desktop for the initial configuration, a hybrid user interface for in-situ refinements, and a pure AR interface for operation. In addition, by facilitating in-situ AR instructions, our system may reduce or eliminate switches that are currently necessary in more traditional workflows (e.g., between robot and manuals).

## 3 ARTHUR: Authoring Tool for Human–Robot collaboration scenarios

ARTHUR is a hybrid authoring tool for creating complete AR-based human–robot collaboration interfaces, integrating on predefined robot behaviors and task-related information (e.g., assembly instructions, bill of materials). We exemplify ARTHUR based on a real scenario informed by our main industry partner (a world-leading toy manufacturer), which involves the assembly and disassembly of plastic injection molds. An injection mold can weigh up to 1.000 kg and consists of a great number of different parts (e.g., metal plates, bolts, pins, o-rings) that must be put together in the correct sequence using a variety of different tools. To lower physical strain and risk of injury, such assembly processes can be supported by robot arms, where the robot might be responsible for repetitive steps, such as fastening bolts or the robot can act as a flexible fixture for the partially-assembled mold to support ergonomic posture of the operator, who then inserts pins, attaches o-rings, or applies grease. Such collaboration requires fluent communication, e.g., the robot notifying the operator, when it is waiting for components to be attached, as well as joint workspace awareness between operator and robot, e.g., to prevent collisions. This can be supported through situated information visualizations with AR (e.g., Lee et al. 2023; Martins et al. 2022). To ensure applicability in real-world scenarios, we must accommodate different workspaces, mold designs, and operator preferences. This calls for a flexible authoring solution that allows for ad-hoc modifications of the system configuration, which we propose to support with a hybrid user interface. Thus, the target group for such a system is UX designers, who design and create processes and guidance materials for assembly workers.

The system setup in our lab is comprised of four main hardware components: an HMD (HoloLens 2), a Robot (UR5e Universal Robots 2024b), a tablet (iPad Pro M1 13"), and a PC (Asus PN51-E1). The user interacts through two main interfaces (Web interface, AR Interface) that were built with Vue and Unity respectively, and rely on a range of services, as illustrated in Fig. 2 (see project repository[2] for further technical details). The web interface supports efficient authoring of feedback and actions using traditional UI elements (e.g., menu selection, text input) on a PC or tablet. The *AR application* presents situated visualizations and allows testing and refining the system as well as manipulating content directly in the workspace. For this, the HMD supports various input modalities, such as gaze, gesture interaction, and speech.

### 3.1 Authoring workflow with a hybrid user interface

We divide the overall workflow with ARTHUR into three phases, between which the user can flexibly transition: (1) *configuration*, (2) *refinement*, and (3) *operation*. These utilize the web and AR interfaces to different degrees, as is described below and illustrated in Fig. 1. In designing these two interfaces, focus was put on maintaining consistency

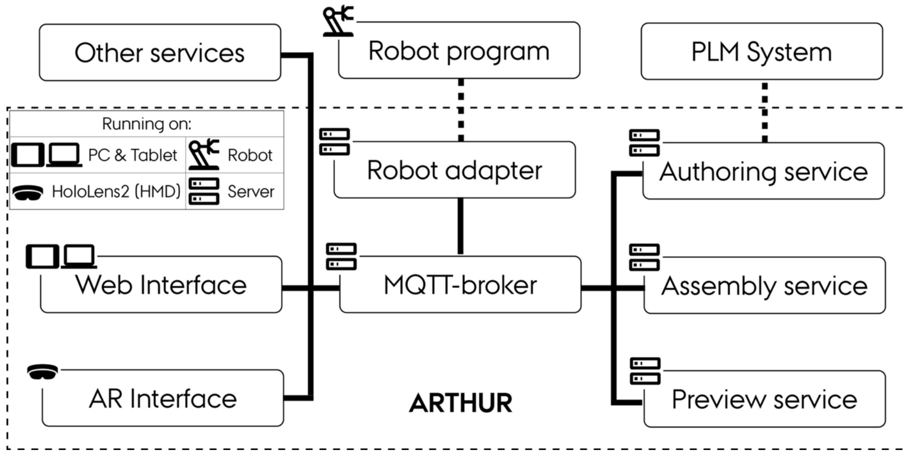


**Fig. 2** Figure showing all interfaces and services and which device they live on. The boundary indicated by a dotted line highlights what is included in ARTHUR and what must be provided, e.g., the robot program is not created in ARTHUR or is a direct part of it but communicates with the system through the robot adapter service

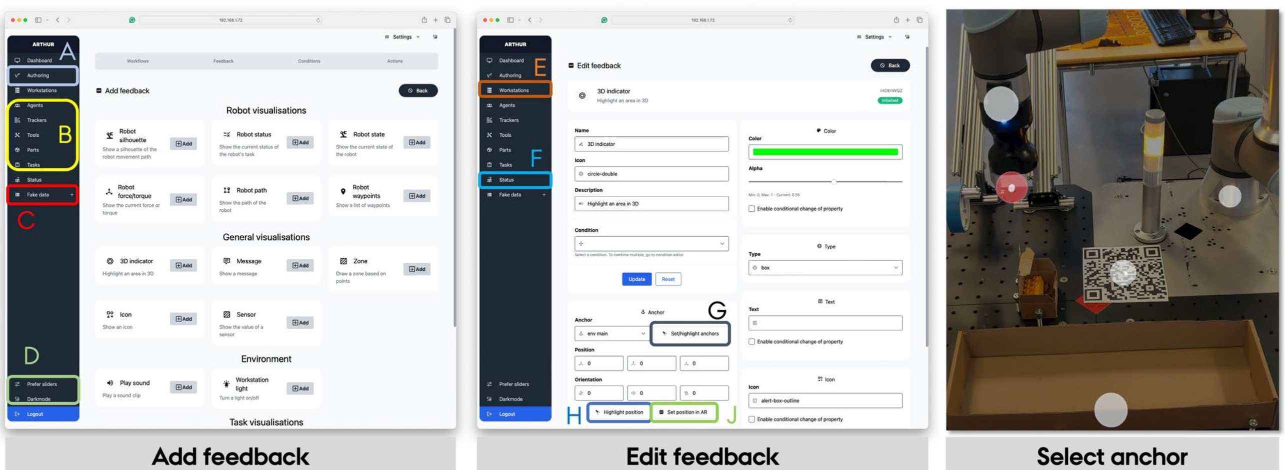


**Fig. 3** Left and middle show screenshots from the web interface, where the rendering is the same across devices (PC and tablet). Right shows the AR interface after clicking "Set/highlight anchors" (G). The pages are similar for all design components (Fig. 4), with a page for viewing, creating, and editing feedback, actions, and conditions. A position can also be viewed in AR (H) and set (J). The sub-menus for authoring is found under (A), it is possible to create a new system setup from the "Workstations tap" (E), edit agents, trackers, tools, parts, and tasks (B), get a status off all services (F), generate "fake data", e.g.,path, waypoints, zones, and messages, which can be used to test the appearance of some visualizations in current lack of any real data (C), and finally customize the interface, e.g.,by showing sliders for number input instead of raw number input (D)

in functionality and appearance across devices, to support recognizability and lower the cost of context switching.

In the *configuration* phase, the general setup is defined on a desktop computer, where we can make full use of the interoperability of several applications (e.g., Product Lifecycle Management systems; PLM) to import existing data (e.g., parts, tools, assembly sequences), as recommended by prior work (Hubenschmid et al. 2022). Considering the example of the injection mold case, one would start by creating a new configuration (under the Workstation tab, Fig. 3, E). Agents can then be added, which in our scenario are a robot (UR5e) and an operator (see Fig. 1). Next, a tracker (QR-code) is added to anchor the robot and virtual content (Fig. 3, Right). Then information about the procedure (e.g., task sequence, tools, and parts) can be added. In our case, these details were exported from the Siemens TeamCenter PLM system, converted from *.XML* into our *.json*-format[4] and then imported into the system.

With all the basic information in place, the authoring of AR content can begin (Fig. 3, A) by creating and configuring the initial set of feedback, actions, and conditions. The *web interface* offers a great variety of design components as building blocks for AR-based HRC workflows, which are further detailed in the next section. Figure 3 (left) shows a list of possible feedback that can be created and Fig. 3 (middle) shows how properties of a feedback component can be edited. The views for *actions* and *conditions* are similar. In our injection mold scenario the AR content could be: *robot path* that highlights the expected movement of the robot,

[4] The *.json*-format conversion is similar as described by Park et al. (Park et al. 2023); sample scripts for conversion are available in the project repository.

*robot state* visible when stopped, *robot status* visible when running showing the current task it is working on and its progress, *step model highlight* for the operator tasks showing a hologram of where parts are to be placed, *tool* and *part highlights* showing what tools and parts to use when relevant, maybe a *step model highlight* for the robot's tasks with reduced opacity, and finally a warning *sound* when the robot initiates a movement. Updating the labels, names, icons, and descriptions for design components in this phase is preferable, as the interface offers the best overview and includes a physical keyboard.

Once the general configuration is made, the user equips an AR-HMD and transitions to the robot workspace to begin the *refinement* phase (see Fig. 1, "Refinement"), for fine adjustments content position and appearance, and testing interactivity. While augmentation of the environment may also be achieved with alternative technologies, such as handheld devices, or projection based approaches (Suzuki et al. 2022; Costa et al. 2022), HMDs allow situated visualization in mid-air, keep the hands free and uninstrumented, and are capable of handling the various design components for AR-supported HRI. The *AR interface* displays the previously authored visualizations in situ and allows editing through mid-air interaction within the robot's workspace including direct manipulation by grabbing with the pinch gesture (e.g., to reposition objects and content anchors, see Fig. 3), activating virtual buttons through a poke gesture, and distant content selection through gaze-pinch. Simultaneously, more complex changes or fine adjustments to the workflow may still be made through the previously described *web-interface*. This can now be accessed through a tablet that is mounted in the near periphery of the robot workspace or hand-held, whereby touch input is supported in place of mouse and keyboard. We hereby capitalize on the *hybrid user interface* concept, benefiting from immediate in-situ visualization in AR, while effectively authoring changes on the touchscreen. Thus, contrary to using Unity or other tools that require some reloading or rebuilding of the application, changes are immediately visible in the *AR interface*.

In our injection mold scenario we may want to align the *task image panel* such that it is just in front of the workpiece on the table, as illustrated in Fig. 1 (see "Refinement"). Some elements cannot be re-positioned, such as the *robot path* that is "attached" to the robot, and the *step model highlight* as the position for each step in an assembly sequence is defined in the Bill of Process (BoP). However, we can verify that the robot and root element of the assembly sequence are correctly positioned in relation to the tracker (physical QR-code) and adjust the offset, if needed. All parts and tools defined in the Bill of Material (BoM), e.g., a hammer, bolts, or a brush and grease bucket, can also be localized, so that they may be highlighted when relevant. ARTHUR can be extended with dynamic localization of tools and parts (e.g., through a vision system), though this is currently not included in the system. We may also wish to adapt the color, line widths and sizes of visualized content (e.g., robot path) for good visibility and minimal occlusion. Finally, we test that the physical buttons in our setup, which can be used to start/stop the robot and mark tasks in the sequence as completed, work as expected.

Lastly, in the *operation* phase, the user tests and uses the fully authored workflow on the HMD by performing the task in collaboration with the robot. The authored system may also be handed over to a different user, who may wish to make ad-hoc changes (e.g., to address a spontaneous change in procedure or accommodate personal preferences) before completing the job.

Our hybrid user interface approach allows the user to fluidly switch between devices, opting for the most convenient technology in each step of the authoring, testing, and deployment procedure, thereby speeding up the design-deploy-refine process. To give a small but very frequently occurring example: naming design components in *configuration* phase requires text input, which is most effectively supported with a physical keyboard on the PC. Typing is required less often in *refinement* phase, when components need to be renamed or new ones added. This is then supported through touch typing on the tablet, which—though inferior to typing on a physical keyboard—is preferable to typing in mid-air using the HMD (Lunding et al. 2024). In contrast, while content positions can be efficiently initialized on the PC, its refinement is left for the *refinement* phase, where the HMD supports in-situ inspection of content alignment with the physical workspace. Here, coarse re-positioning of content is supported through direct manipulation, while the tablet interface offers fine-tuning controls.

### 3.2 Design components

ARTHUR offers three kinds of design components as building blocks (see Fig. 4): *feedback* communicates the state of the robot and system to the user (e.g., as in-situ visualizations); *actions* allow the user to control the robot and system; and *conditions* serve to automatically trigger *actions* or dynamically adjust *feedback* properties. All components can be individually customized through properties (e.g., width and color of robot path visualization), which are instantly synchronized across all connected interfaces. In addition, ARTHUR supports *trackers and anchors* to easily and consistently place content in the real world, and *task-related data* (see Sect. 3.2.5).

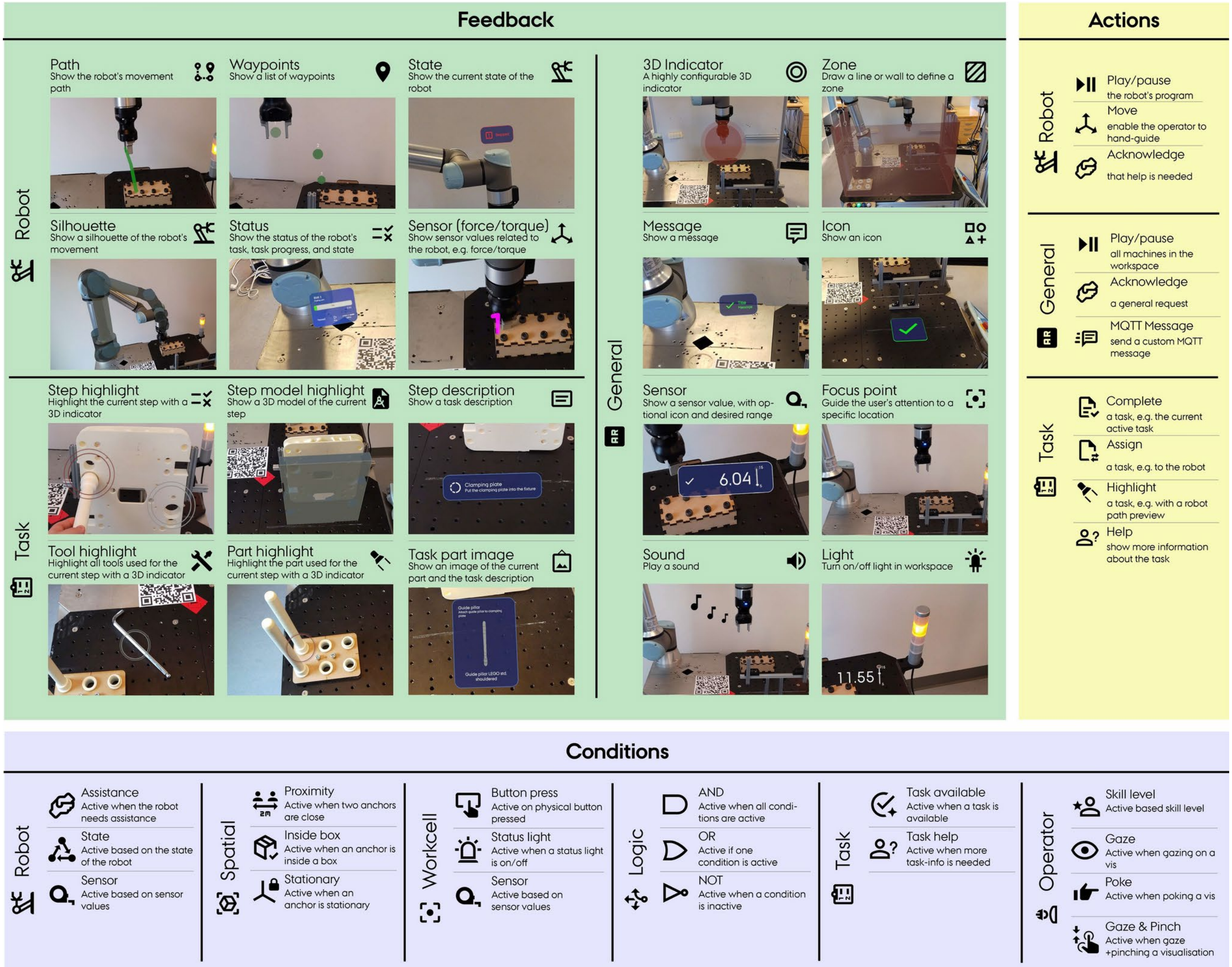


**Fig. 4** Overview of all *feedback*, *actions*, and *conditions* currently implemented in ARTHUR

### 3.2.1 *Feedback*

ARTHUR currently supports 20 kinds of *feedback* (see Fig. 4) to inform users about the state of the robot and system. While we currently focus on visual *feedback* (e.g., visualizations), we aim to explore other modalities afforded by AR HMDs (e.g., audio, haptics) in future work. We group *feedback* into visualizations that concern the *robot*, the *task*, or *general* purpose.

*Robot*-related *feedback* includes visualizations about the robot's movement path, waypoints, silhouette, state, or sensor readings. In addition, users can add a visualization about the current task and its progress. While most of the data can be gathered directly from the robot in real time, some feedback (e.g., path, silhouette) requires estimations about future movements. We support this through the *Preview service*, which is described in Sect. 3.3.

*Task*-related feedback guides the user through the workflow. We support step-by-step instructions (e.g., task description) and situated highlights (for task, task model, tool, or part), allowing users to quickly identify relevant components. Further, the overall task status allows them to monitor progress and coordinate activities with the robot.

Lastly, *generalfeedback* enables to customization of the user's workspace and environment. This includes the capability to show 3D indicators, icons, or zones, thus enabling support for a wide range of scenarios (see Sect. 4.1). For example, a 3D indicator can be turned into a virtual button by utilizing its "poke" condition or to create a "stay-out area" where the user is not supposed to be inside while the robot is moving. Likewise, icons can be used to convey the state of the system (e.g., if a sensor value is at its desired level or an object is placed correctly). Finally, physical lights within the workspace and spatial audio can be used to provide feedback.

#### 3.2.2 *Actions*

*Actions* are ways for the user to communicate or express commands to the system. Similar to prior work (e.g., Suzuki et al. 2022), we support input modalities commonly available in current AR HMDs, such as gaze, speech, and gestures, as well as relative object position (e.g., proximity). While *actions* vary greatly by usage scenario, we implemented a basic set of 10 common *actions* based on our related work analysis (see Fig. 4). We again differentiate actions that concern the *robot*, *task*, or are *general* purpose.

*Robotactions* allow the user to play or pause the robot program, send a confirmation, or trigger move mode (i.e., hand-guiding). It should be noted that these *actions* might not be applicable or possible on all types of robots and in every context. *Task*-related *actions* give the user the opportunity to control the sequence of tasks, for example by manually confirming the completion of a task, reassigning it to a different agent (i.e., task scheduling), or selecting a specific task to view more information about it. Finally, *generalactions* work on a broader scale and currently include a general acknowledgment *action*, a global play/pause function (e.g.,for all machines in the workspace), and transmit custom MQTT (OASIS MQTT Technical Committee 2024) messages, thus allowing the operator to send input to other connected systems.

#### 3.2.3 Conditions

*Conditions* can be used to control when, where, and how *feedback* appears or to trigger *actions*. We have implemented 18 *conditions* in 5 different categories: *spatial*, *operator*, *robot*, *environment*, *task*, and *logic* (see examples in Fig. 4). To illustrate a potential usage scenario: We might apply *Spatialconditions* to hide detailed visual information for a workspace the operator is not currently attending to by detecting when they are more than three meters away (*proximity condition*). Another example is the *robot assistancecondition* which can be triggered to alert the operator to this workspace when needed, e.g., because the material dispenser is empty.

*Conditions* are generally created through the *web interface*. However, some *feedback* types and *actions* imply *conditions*, which are automatically created. For example, the creation of interactive visualizations, such as the *3D Indicator* and *Task model highlight*, automatically includes a corresponding gaze, gaze+pinch, or poke *condition*. All *conditions* can further be customized: they are either active or inactive and some provide additional parameters. For example, gaze, gaze+pinch, and poke refer to the respective visual element.

#### 3.2.4 Trackers and anchors

To register virtual content in the real world (e.g., the robot movement path) it is necessary to align the coordinate systems of the robot and HMD. Furthermore, it is desirable for some virtual content to be fixedly situated in the environment. This can be authored using *trackers* and *anchors*.

*Trackers* are used to localize the HMD in the world. Currently, this is supported through QR-codes that can be applied to physical surfaces and scanned with the HMD (HoloLens2). All trackers have an *anchor* attached, which then can be used to fix virtual content and position robot(s) in relation to it. Additional *anchors* can be created to specify fixed content locations, e.g., at the base, joints, and tool-center point of the robot. For those familiar with Unity3D or similar concepts: a tracker can be seen as a game object with an anchor at its local origin (0,0,0). Additional anchors can be created in relation to these tracker anchors.

By default, anchors are also attached to the user's hands and head, which is tracked by the HMD. This can serve to define a *proximity condition* that activates based on a distance-threshold between two anchors (e.g., distance between user's hand and end effector of the robot).

#### 3.2.5 Other components

ARTHUR allows the definition of agents (i.e., operators and robots) with specific attributes. For example, operators have a skill level that may modulate the level of detail of visualized information. Robots have a type, (e.g., UR5e, UR10e, KUKA iiwa), tools (e.g., gripper), and a position relative to a tracker in the environment. This serves to instantiate and situate the corresponding model of the robot.

Further assembly-relevant data includes the tools and components (BoM), and tasks (BoP). These are imported from external sources, e.g., a PLM system, such as SIEMENS Teamcenter. A similar approach was pursued by Sääski et al. (2008). To facilitate correctly situated visualizations based on this data, the location of tools, parts, and tasks can be specified as part of the authoring process, e.g., by relating it to anchors or by adding a custom Tool Recognition Service (e.g., using vision-based scene segmentation approaches). The supported assembly procedures were informed by prior work (Park et al. 2023; Sääski et al. 2008) and our industry partners in MADE FAST.

### 3.3 Services and extensibility

ARTHUR offers multiple services that support the two interfaces and various aspects of HRC. An overview of the system architecture is given in Fig. 2.

The *authoring service* represents the system's core and is responsible for handling and storing all changes to the configuration of the authored setup. Thus the authoring service is required for ARTHUR to function.

The *assembly service* is an optional service responsible for bookkeeping during an assembly process, where it loads the assembly sequence from a database shared with the authoring service and manages the task scheduling (task assignments and completion status).

The *preview service* is an optional service that implements a simple way of getting robot motion previews, used for path and silhouette feedback, without the need for a simulation. The service automatically records the tool-center point (TCP) and joint angles of the robot when it is doing a task. This data can then be replayed on request, e.g., for displaying the robot movement path to convey motion intent and planned actions.

A *robot adapter service* is responsible for handling the communication with the robot, like retrieving joint angles and TCP, and informing the robot about tasks that it must perform. Currently, ARTHUR supports a Universal Robots (UR) (Universal Robots 2024b) adapter. The UR-adapter uses the RTDE-interface (Universal Robots 2024a) to get data from the robot and a XML-RPC server from which the robot program can retrieve information about its next task and send progress updates. The robot program is not a part of ARTHUR as it will be unique to every context. Communication between the robot and its program is intended to happen through the robot adapter. Additional robots can be added in the future. From the perspective of ARTHUR, this is done by implementing a new *robot adapter service* for that specific type of robot. The adapter must implement the minimum required functionality, e.g., data from the robot like joint angles, TCP, and robot state and data to the robot, such as which task(s) it may perform. The details, likely to change over time, can be found in the repository[2]. The robot adapter can be ROS-based if applicable.

To handle communication between all user interfaces and services, ARTHUR relies on MQTT (OASIS MQTT Technical Committee 2024). This allows clients to exchanging JSON-encoded messages via publish and subscribe to topics via an MQTT-broker (Eclipse foundation 2024). The integration of additional services and systems is well-supported by MQTT. For example, the system can be extended by an additional 'safety-zone' service, as described in Sect. 4.1.1 or to communicate with ROS-services by using a ROS-MQTT-bridge.

If additional design components (i.e., *feedback*, *actions*, and *conditions*) are needed, ARTHUR allow developers to extend the system with those. This is a fairly simple process, which requires changes to two parts of ARTHUR: (1) Basic information about the component: name, icon, description, and a list of all properties must be added to the *authoring service*. The following 12 properties are currently supported: *boolean, integer, float, string, anchor, pose, vector3, condition, color, agent, enum, multi select enum*. As an example, the path visualization has the following properties: agent, width (float), and color. (2) The *AR interface* must also be updated with the actual implementation. Each type of component has a specific interface that must be implemented, which generally comprises three methods: *init*, *cleanup*, and *ListOfProperties*. By implementing this interface, the new design component gains access to all data in the system. More details are available in the documentation[2].

## 4 Evaluation of ARTHUR

Aiming to explore whether our proposed system is viable for adoption in the industry, we first reflect on the potential of ARTHUR through *demonstration* (Ledo et al. 2018) of replicated scenarios (see Sect. 4.1). Secondly, we verify the suitability of our hybrid user interface approach in a qualitative *usage evaluation* (Ledo et al. 2018) with AR experts (see Sect. 4.2).

### 4.1 Demonstration of application scenarios

To exemplify the capabilities of ARTHUR, we explore three application scenarios that involve: *replicating a system* (Hietanen et al. 2020), creating an environment for *comparing visualizations* (Arevalo et al. 2021; Cleaver et al. 2021; Cogurcu and Maddock 2023; Gruenefeld et al. 2020), and *displaying sensor values* (De Franco et al. 2019; Fuste et al. 2020; Renner et al. 2018), as described below and illustrated in our supplementary videos[3].

These scenarios were chosen based on replicability (i.e., all central components were described in sufficient detail) and as they represent major current directions in AR-HRC research, or address central pain points in the manufacturing industry. The scenarios further highlight the challenges of current authoring approaches (e.g., Lunding et al. 2024), requiring a mixture of in-situ (e.g., placing objects in 3D, choosing color and opacity of virtual objects, defining suitable thresholds and distances) and ex-situ authoring (e.g., importing data). It should be noted however, that ARTHUR was not specifically designed for any of these scenarios and may be more widely applied.

#### 4.1.1 Scenario 1: Replicating a system

In this scenario, we replicate the HoloLens setup as described by Hietanen et al. (2020). The original setup consist of six visual elements, though seven are used in ARTHUR to

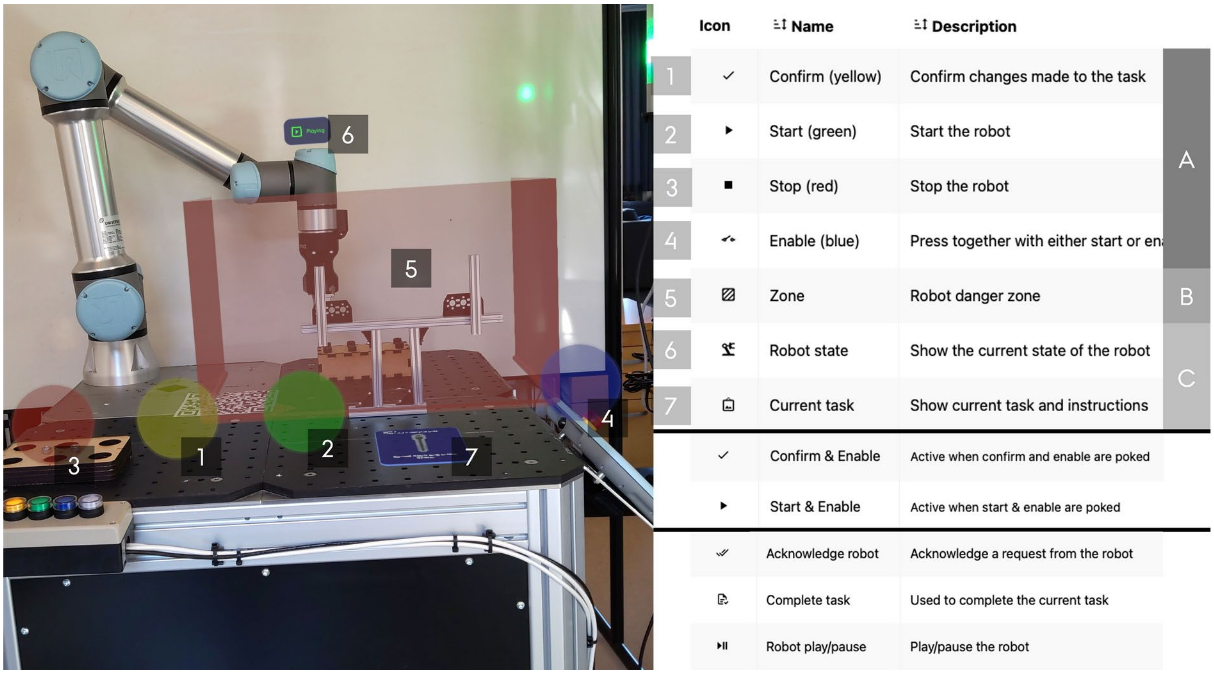


**Fig. 5** Scenario 1 (Hietanen et al. 2020) replicated in ARTHUR. The left part shows the view from the AR interface, with each design component annotated with a number. The right part depicts a list of feedback from the web interface, with corresponding numbers annotated, a list of created conditions, and finally a list of actions

achieve the same result (see Fig. 5): (A) touching four differently colored spherical objects in the workspace allows the user to *start* (green) and *stop* (red) the robot, *confirm* (yellow) changes that have been made to the task, and *enable* (blue) the start and confirm buttons, (B) a dynamically-updated safety-zone visualized as a semi-transparent wall around the robot, finally (C) a graphical box with an image and text showing the robot's current status and instructions.

The setup can be replicated in ARTHUR as follows: (A) four *3D indicator* visualizations with appropriate shape (sphere), color, and positioning. Each *3D indicator* has an associated *poke condition*, which allows for interactivity through mid-air gestures. The *stop button* (red) was set to activate a *robot play/pause action*. The *start button* (green) informs the robot that it may initiate its next task, and is thus set to trigger a *robot acknowledge action* when poked simultaneously with the enable (blue) button. This is achieved using an *AND* condition. (B) a *zone visualization* can be used to display a semi-transparent wall around the robot. While ARTHUR currently does not explicitly support automatically update of safety zones, it is easy to extend the system with a service that calculates and continuously publishes a series of points around the robot along with a zone-id through MQTT. Finally, the *confirm button* (yellow) triggers a *complete task action* through a simultaneous *poke* of this and the enable (blue) button. Lastly, (C) is split into two elements: a *task image visualization* which shows a panel with a description of the next task, retrieved from the task description, and a *robot state visualization* that shows the current state of the robot (i.e., playing, paused, stopped).

#### 4.1.2 Scenario 2: Comparing visualizations of prior work

There is a considerable amount of related work exploring the benefits of individual visualizations types in isolation, where multiple variants of a visualization type is compared, such as: different visual cues to assist with grasping (Arevalo et al. 2021), ways to configure visual safety zones (Cogurcu and Maddock 2023), and comparison of strategies for robot motion intent through various visual representations of paths (Cleaver et al. 2021; Gruenefeld et al. 2020). However, the distinct physical setups and application scenarios make findings difficult to compare and transfer to other scenarios.

As ARTHUR already contains an extensive collection of *feedback*, *actions*, and *conditions*, it provides a good starting point for recreating and comparing variants of visualizations from prior research. For example, we can recreate various visualizations for robot movement intent, as illustrated in Fig. 6 for direct comparison in ARTHUR. Our hybrid authoring approach allows users to easily realize such visualizations following our three phases: In *configuration* phase, users can set up a general workspace with different visualization and interaction elements, to switch between these visualizations on the fly (similar to Scenario 1). In *refinement* phase, they can fine-tune these visualizations on

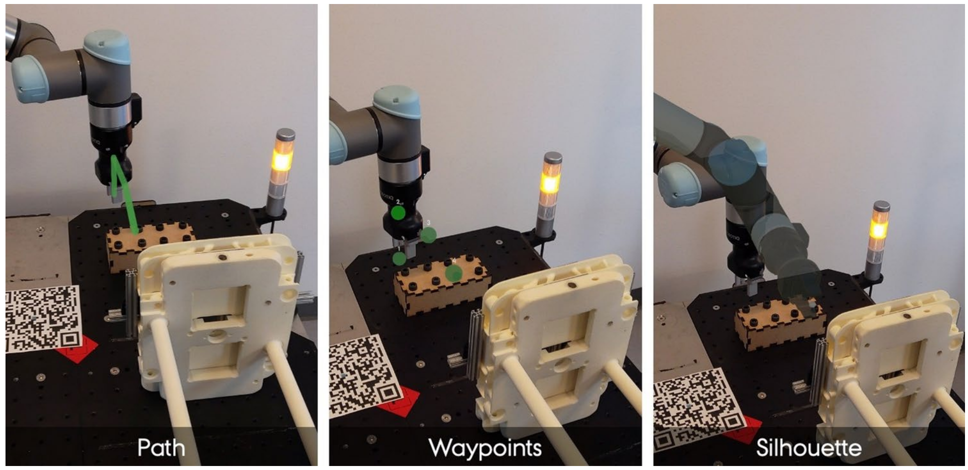


**Fig. 6** Setup for comparing previews of robot motion. Similar setups can be found in Cleaver et al. (2021), Cogurcu and Maddock (2023), Rosen et al. (2019), Arevalo et al. (2021)

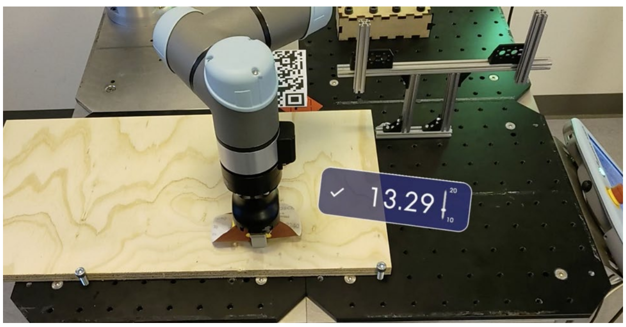


**Fig. 7** Pressure-sensing sanding setup (De Franco et al. 2019) replicated in ARTHUR

a tablet and observe the output in situ with the AR-HMD (e.g., making sure the color of visualizations are appropriate). Finally, in the *operation* phase, users can seamlessly test, switch, and compare the visualizations.

Additionally, ARTHUR makes it easy to compare variations of a specific visualization in a broader context, as the existing collection of design components is readily available.

#### 4.1.3 Scenario 3: Sensor values

Integrating and visualizing sensor values (e.g., from robot or attached tools) is a common scenario for assisting the user during HRC (e.g. De Franco et al. 2019; Fuste et al. 2020; Renner et al. 2018). This might be most relevant when programming the robot, but can also be useful during operation to verify that the robot operates within the desired limits. For example, prior work has explored battery indicators (Renner et al. 2018) or displaying a scale to inform about the amount of loaded material (Fuste et al. 2020).

ARTHUR can be used to easily replicate similar setups, such as the work by De Franco et al. (2019), where the amount of pressure applied during sanding is visualized in an AR interface (see Fig. 7). Using our set of design components, we can create a *sensor visualization* and connect it a digital scale visualization via our MQTT-interface. The general setup and configuration (e.g., reading documentation on the scale, setting up MQTT topics) is best performed on a desktop during *configuration*. In contrast, the position of the *sensor visualization* can be either attached to an anchor within the scene or placed through mid-air gestures and fine-tuned in-situ—thus, this step is best done using an AR-HMD and tablet during *refinement*. Lastly, users can immediately test and validate the visualization by switching to the *operation* phase.

### 4.2 Usage evaluation with AR experts

To verify the suitability of the supported authoring functionality across the different interfaces of our hybrid UI, we invited five experts in AR application development to test ARTHUR. Our goals were to (1) observe and assess how ARTHUR is used to author a holistic HRC process, (2) examine the qualitative usability and utility of ARTHUR, and (3) validate our hybrid approach of distributing the authoring across multiple phases and devices. While the target group for ARTHUR may be described as UX designers who create manuals with task instructions for assembly workers, our study participants' professional expertise within AR and AR-HRC enabled them to judge not only on the potential and usability of ARTHUR, but also to spot missed opportunities or possible issues based on their technological knowledge.

All participants were male, between 22 and 36 years of age, and identified as researcher (P1), PhD student (P2), graduate student (P3), and software developers (P4, P5). On average, participants rated their experience with head-mounted AR as high and working/interacting with robots as medium, but indicated little to no experience with hybrid user interfaces and robot programming.

#### 4.2.1 Procedure

After welcoming participants to the lab, they gave their written consent for voluntary participation and data collection, and filled a demographic questionnaire. We then introduced the study task, which was to replicate the system described in Scenario 1, and explained the different phases of ARTHUR. We explained the hybrid authoring approach, which included a walkthrough of all sub-menus in the web-interface (see Fig. 3), an exemplary creation of a message (*feedback*), workstation button (*condition*), and play/pause robot (*action*) on a desktop computer, followed by inspecting the output on the HMD and starting a dummy robot-program by triggering the play/pause robot *action*. We also demonstrated the cross-device interaction by starting the 3D placement of the created message (*feedback*) on the tablet, then switching to the AR-HMD to place the message within the workspace, and switching back to the tablet to confirm its position. Once participants felt comfortable with the system, we explained Scenario 1 by showing them the intended outcome (see Fig. 5) and describing all visual elements and their expected functionality.

Participants then started the authoring process at a desktop computer (*configuration* phase), before transitioning to the physical setup with the AR-HMD and Tablet (*refinement* phase), and then finally testing their setup (*operation* phase). Once they completed the task, they were provided with additional tasks to further explore the system, e.g., creating 3D step instructions, adding auditory feedback to button interactions, and only showing the zone when the robot was active. They were then invited to try out their additions and freely explore the system further before the session concluded with a semi-structured interview. Each session lasted 75 min, whereby the explanation of scenario one and the three phases took up 25–30 min.

#### 4.2.2 Main usage evaluation findings

We performed a thematic analysis of the experimenter's written protocol of each study session that was further refined by inspecting video and audio recordings and transcripts from interviews. In summary, all participants were able to successfully complete the study task by replicating the setup presented in Scenario 1, as well as adding and configuring design components of their choice. Furthermore, all participants agreed that ARTHUR works well and is easy to use, that the options were plentiful, and that distributing tasks across different devices in the authoring process is beneficial: "The combination of a decent input/menu device (tablet or PC) and immediate feedback on the HoloLens is great." - P5

In particular, participants appreciated having a tablet as a menu, instead of virtual menus on the HoloLens. P4 and P5 also positively highlighted that AR assembly instructions are not authored per-instruction, but by loading the BoP and using generalized visualizations, which has the potential to drastically reduce authoring time for step instructions and effortlessly allow for changes to BoP.

Participants also liked that the hybrid authoring approach allowed them to use AR for coarse object positioning and the tablet for fine-tuning: "It was easy to use the tablet simultaneously with the HMD - they complement each other well." - P3. Four participants further highlighted the benefits of a mixture of devices, for example explaining that "The utility of the PC increases as the system becomes more complex. But for smaller, simpler setups, the iPad is great, as it is much more nimble, and you can be in-situ with the [AR HMD] on." - P1.

Importantly, participants reported no issues when switching between authoring phases, but valued that they could "quickly do things here [on the computer] and then go over to place stuff [in-situ]" - P2. The context switches were not perceived as a problem, as "You can transition between the tablet and HoloLens so easily. You just press [on the tablet] and then it immediately updates in AR. No need to wait for a configuration to load or similar, you just click and then it updates." - P4. This participant similarly found the context switching between PC and tablet effortless as the visual interface is identical.

While participants were overall pleased with ARTHUR, they noted some usability issues regarding the AR authoring workflow. When setting a position in AR, they unintentionally rotated the object. Here, participants suggested a rotation lock or separating the 3D manipulation tasks. Furthermore, all *feedback* is currently hidden while positioning an object in AR, which reduces visual clutter, but also makes it difficult to align objects in relation to each other. A solution could be to reduce the opacity of all visualizations or add a button for toggling visibility. P4 also asked for better ways to align content, which could be achieved with 'snapping', copying the position of other objects, and enable the user to constrain the movement to specific axis.

## 5 Discussion

While prior research presents custom implementations of AR guidance or feedback for robot behavior (Andronas et al. 2021; Ganesan et al. 2018; Hietanen et al. 2020), these do not permit modification by the end-user in situ. Authoring AR-based HRC workflows with existing tools on "traditional interfaces" (e.g., designing content in simulated workspaces on desktop PC systems) remains a challenge for various reasons, such as discrepancies between the development environment and the deployed application. Recently proposed in-situ authoring tools (e.g., Lunding et al. 2024) attempt to close this gap, but report limitations of usability due to mid-air interaction modalities. Prior work in the field of visual analytics has indicated that a hybrid approach (i.e., combining heterogeneous devices) allows users to switch between appropriate interfaces as they see fit. However, this was mostly studied in isolation, i.e., either as a discrete switch (cf. Hubenschmid et al. 2022) or simultaneous use of both interface (cf. Hubenschmid et al. 2021a; Langner et al. 2021; Reipschläger et al. 2021). By combining these use patterns (e.g., discretely switching between the *configuration* and *refinement* phase, but continuously switching between tablet and AR-HMD in the *refinement* phase), we showcase such hybrid approaches within a holistic system. Initial results from our evaluation indicate that this works well for authoring, as it supports users in transitioning between different tasks. While there may be increased cognitive load from switching between interfaces (cf. Rashid et al. 2012b), we think that the apparent benefits far outweigh possible downsides in our system. However, further studies are obviously necessary to confirm this claim.

### 5.1 Limitations of the evaluation

While our initial evaluation demonstrated the general feasibility of our hybrid authoring approach, further studies are needed to investigate the large number of possible design parameters. For example, while we are conscious about the importance of reusing familiar interaction concepts, and meaningful icons and descriptive labels in the UI, further refinement or customization to specific application domains (e.g., distinct manufacturing companies) will likely reduce the learning curve. Also, although participants reported no significant issues with context switches, more controlled experiments may yield further insights into their potential impact and further improvements when switching between phases and interfaces.

While we intentionally recruited AR and HRC experts to gain insights on technical and design aspects of ARTHUR, evaluating our system with domain specialists such as robotics engineers, UX designers, and assembly workers at a manufacturing company could contribute further highly valuable perspectives. While UX designers, whom we see responsible for authoring AR-HRC systems in the wild, were partially represented by study participants P4 and P5, a more extensive formal evaluation with target users from the production facility was beyond scope for the present paper, which focuses on the investigation of a hybrid user interface for authoring HRC workflows. However, in a recent publication (Lunding et al. 2025) we report on an in-the-wild study involving all three stakeholder groups. This exploration revealed that assembly workers mostly appreciated a robot arm as a strong extra set of hands, and liked AR for assembly instructions and communication of robot intent. UX designers were positive about the offered authoring functionality, but also requested further features. Importantly, the engineers confirmed the feasibility of integrating ARTHUR in future semi-automated workflows. Beyond this, more formal in-the-wild studies with assembly workers are necessary to confirm the real-world applicability and benefit of ARTHUR (Suzuki et al. 2022) for example in terms of satisfaction and performance when performing relevant tasks. This would also permit assessing the potential need for prior special training of assembly workers when adopting AR-based HRI in the manufacturing process. Finally, a comparison of ARTHUR with state-of-the-art solutions could provide further insight into actually achievable improvements in work performance. Alas, to the best of our knowledge, current openly available systems only support either robot interaction (e.g., RoboVisAR Lunding et al. 2024) or assembly guidance [e.g., Microsoft Guides 365 (Microsoft 2025)], but not both. We therefore encourage and invite researchers to compare their custom solutions to ARTHUR and thereby offer this much-needed evidence.

### 5.2 Beyond robot arm assembly and HMDs

While ARTHUR focuses on the authoring experience for collaborative assembly with robot arms, we see potential in applying the same system to other areas. For example, support for mobile robots could be added, provided that suitable tracking is supported. Although fiducial markers (e.g., QR codes) already offer decent stability and precision for this scenario, current AR hardware still has significant limitations (e.g., low frequency of marker detection) that make a real-world evaluation difficult on mobile robots. However, if tracking is solved in an environment with mobile robots, a broad range of robot-related *feedback*, *actions*, and *conditions* can be used without any modifications.

One could also imagine using ARTHUR for tasks other than assembly, such as supervising robot programming. While it is possible to support robot programming with ARTHUR, it will likely require additional types of visualizations and data integration to maximize its value. Despite our primary focus on robot-supported assembly tasks, we argue that our concepts also apply to a wider range of HRC applications (e.g., robot-assisted inspection, domestic service robots). Our system can be used as a testbed for other researchers, as it accelerates the creation and exploration of AR-supported HRC setups.

Furthermore, ARTHUR could be extended to integrate other display technologies such as spatial (e.g., via projectors) or handheld (e.g., via smartphones) AR devices. Our proposed workflow generally supports alternative display technologies, though some changes would be required to properly support our *feedback* components. For example, the use of spatial projections could address some issues with ergonomics arising through the bulkiness and poor wearability of current HMDs. However, projection-based solutions often face technical challenges like presenting content on reflective surfaces, dealing with occlusion, and convincingly situating mid-air visualizations. Further, statically mounted tablets have been found useful for remote assistance, providing the operator with an augmented window to look through (Rasmussen et al. 2022), but whether such arrangements are feasible in HRC is yet to be explored.

With regards to the interaction modality, ARTHUR currently leverages standard mid-air interaction capabilities as supported by the HoloLens2 for the authoring task. The designer then also has the opportunity to include different interaction modalities (e.g., particular gestures) for use, by setting corresponding conditions (see Sect. 3.2.3). In future extensions of ARTHUR we aim to support additional modalities, such as speech, or further standard gestures supported by the AR device/MRTK.

### 5.3 Configuration recommendations and workflows

To further improve usability and reduce authoring time, ARTHUR could be extended with a recommendation system that enables the user to add *groups* of *feedback* and *actions* that play well together instead of having to configure everything from scratch. Such preconfigurations or templates could be generated based on individual tasks, workflows, or more general groups of components that are typically used in existing assembly procedures. Alternatively, these could be based on a shared library of user-defined procedures. However, defining, collecting, and presenting such recommendations for an HRC setup will require more research to elucidate the needs, suitable strategies, and effective techniques (Suzuki et al. 2022).

In addition to recommendations, it could be useful to automatically detect all connected sub-systems. For instance, detecting which sensors, buttons, and lights are available in the workstation and only presenting these as options could help avoid errors by guiding the user to only author *feedback* and *actions* that are supported by the physical setup.

## 6 Conclusion

We propose ARTHUR for in-situ authoring of augmented reality guidance for human–robot collaboration through a hybrid user interface. We thereby leverage the familiarity and advanced capabilities of a desktop computer or tablet for menu interaction and an AR headset for 3D visualization and spatial configuration of virtual content through mid-air interaction. Hence, our system addresses limitations from previous systems by (1) extending the possible AR guidance to include assembly instructions; (2) establishing bidirectional communication between robot and operator (by defining *actions* for the operator); and (3) combining strengths of multiple types of devices and user interfaces to provide a better user experience.

With this paper, we aim to highlight some of the most central opportunities and challenges of hybrid user interfaces for authoring AR-supported HRC systems. We have implemented the system and shown its capability for replicating a broad range of systems from related work. Lastly, we have shown a path for further exploration of AR-supported HRC by utilizing ARTHUR, thereby facilitating the integration of robots to relieve strenuous human manual labor in manufacturing processes.

**Acknowledgements** Thanks go to Roberta, for always doing what it was told to do, no matter how stupid. This work was supported by the Innovation Fund Denmark (IFD Grant No. 9090-00002B) for the MADE FAST project—part of the Manufacturing Academy of Denmark (MADE) and the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation)—Project-ID 251654672—TRR 161.

**Author Contributions** All authors were involved in defining the research objective and main research questions. The concept of ARTHUR was jointly developed by R.L., S.H., and T.F.. The system was implemented by the first author (R.L.), who also planned and realized the demonstration of application scenarios. The expert evaluation was planned by R.L. and T.F., while R.L. alone was responsible for preparing and conducting the studies, as well as collecting and analyzing the data. R.L., S.H., and T.F. wrote the main manuscript text, R.L. prepared all figures, and all authors reviewed and revised the manuscript.

**Funding** Open access funding provided by Aarhus Universitet
This work was supported by the Innovation Fund Denmark (IFD Grant No. 9090-00002B) for the MADE FAST project—part of the Manufacturing Academy of Denmark (MADE) and the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation)—Project-ID 251654672—TRR 161.

**Materials availability** Supplementary video clips, e.g., of the demonstrated application scenarios, can be found in the ARTHUR video playlist on YouTube: https://www.youtube.com/playlist?list=PLhjFAueqW0cuRR-ZruATwfFXfasdi5Gli.

**Code availability** The source code is available in the ARTHUR project repository: https://gitlab.au.dk/arthur.

## Declarations

**Conflict of interest** The authors have no competing financial or non-financial interests to disclose.

**Ethics approval** The study was conducted in accordance with the standards and policies for responsible research conduct set by Aarhus University and follow national guidelines for ethical research, as well as GDPR rules regarding personal data protection.

**Consent to participate** All participants were informed about the study's purpose and procedure and consented to volunteer without financial compensation.

**Consent for publication** All participants gave consent for the publication of results based on anonymized data recorded during their study sessions.



## References

Andersen RS, Madsen O, Moeslund TB, Amor HB (2016) Projecting robot intentions into human environments. In: 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE, United States, pp 294–301. https://doi.org/10.1109/ROMAN.2016.7745145

Andronas D, Apostolopoulos G, Fourtakas N, Makris S (2021) Multimodal interfaces for natural human–robot interaction. Procedia Manuf 54:197–202. https://doi.org/10.1016/j.promfg.2021.07.030

Arevalo Arboleda S, Rücker F, Dierks T, Gerken J (2021) Assisting manipulation and grasping in robot teleoperation with augmented reality visual cues. In: Proceedings of the 2021 CHI conference on human factors in computing systems (CHI'21). ACM, New York, NY, USA, pp 1–14. https://doi.org/10.1145/3411764.3445398

Brudy F, Holz C, Rädle R, Wu C-J, Houben S, Klokmose CN, Marquardt N (2019) Cross-device taxonomy: survey, opportunities and challenges of interactions spanning across multiple devices. In: Proceedings of the 2019 CHI conference on human factors in computing systems (CHI'19). ACM, Glasgow, Scotland UK, pp 1–28. https://doi.org/10.1145/3290605.3300792

Büschel W, Mitschick A, Meyer T, Dachselt R (2019) Investigating smartphone-based pan and zoom in 3D data spaces in augmented reality. In: Proceedings of the 21st international conference on human-computer interaction with mobile devices and services (MobileHCI'19). ACM Press, Taipei, Taiwan, pp 1–13. https://doi.org/10.1145/3338286.3340113

Butscher S, Hubenschmid S, Müller J, Fuchs J, Reiterer H (2018) Clusters, trends, and outliers: how immersive technologies can facilitate the collaborative analysis of multidimensional data. In: Proceedings of the 2018 CHI conference on human factors in computing systems (CHI'18). ACM, New York, New York, USA, pp 1–12. https://doi.org/10.1145/3173574.3173664

Cao Y, Wang T, Qian X, Rao PS, Wadhawan M, Huo K, Ramani K (2019a) Ghostar: a time-space editor for embodied authoring of human–robot collaborative task with augmented reality. In: Proceedings of the 32nd annual ACM symposium on user interface software and technology (UIST'19). ACM, New York, NY, USA, pp 521–534. https://doi.org/10.1145/3332165.3347902

Cao Y, Xu Z, Li F, Zhong W, Huo K, Ramani K (2019b) V.ra: an in-situ visual authoring system for robot-iot task planning with augmented reality. In: Proceedings of the 2019 on designing interactive systems conference (DIS'19). ACM, New York, NY, USA, pp 1059–1070. https://doi.org/10.1145/3322276.3322278

Chan L-W, Kao H-S, Chen MY, Lee M-S, Hsu J, Hung Y-P (2010) Touching the void: direct-touch interaction for intangible displays. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI'10). ACM, Atlanta, GA, USA. https://doi.org/10.1145/1753326.1753725

Chan WP, Hanks G, Sakr M, Zuo T, Loos HF, Croft E (2020) An augmented reality human-robot physical collaboration interface design for shared, large-scale, labour-intensive manufacturing tasks. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, Las Vegas, NV, USA, pp 11308–11313. https://doi.org/10.1109/IROS45743.2020.9341119

Cheon E, Schneiders E, Diekjobst K, Skov MB (2022) Robots as a place for socializing: influences of collaborative robots on social dynamics in- and outside the production cells. In: Proceedings of the ACM human–computer interaction 6(CSCW2), pp 1–26 https://doi.org/10.1145/3555558

Cleaver A, Tang DV, Chen V, Short ES, Sinapov J (2021) Dynamic path visualization for human–robot collaboration. In: Companion of the 2021 ACM/IEEE international conference on human–robot interaction (HRI'21) companion. ACM, New York, NY, USA, pp 339–343. https://doi.org/10.1145/3434074.3447188

Cogurcu YE, Maddock S (2023) Augmented reality safety zone configurations in human–robot collaboration: a user study. In: Companion of the 2023 ACM/IEEE international conference on human–robot interaction (HRI'23). ACM, New York, NY, USA, pp 360–363. https://doi.org/10.1145/3568294.3580106

Costa GdM, Petry MR, Moreira AP (2022) Augmented reality for human–robot collaboration and cooperation in industrial applications: a systematic literature review. Sensors 22(7):2725. https://doi.org/10.3390/s22072725

De Franco A, Lamon E, Balatti P, De Momi E, Ajoudani A (2019) An intuitive augmented reality interface for task scheduling, monitoring, and work performance improvement in human-robot collaboration. In: 2019 IEEE international work conference on bioinspired intelligence (IWOBI). IEEE, Budapest, HU, pp 75–80. https://doi.org/10.1109/IWOBI47054.2019.9114472

Dimitropoulos N, Togias T, Michalos G, Makris S (2021) Operator support in human–robot collaborative environments using ai enhanced wearable devices. Procedia CIRP 97:464–469. https://doi.org/10.1016/j.procir.2020.07.006

Eclipse foundation: Mosquitto, May 09, 2024. https://mosquitto.org. Accessed 23 Jan 2025

Elmqvist N (2023) Data analytics anywhere and everywhere. Commun ACM 66(12):52–63. https://doi.org/10.1145/3584858

Ens B, Lanir J, Tang A, Bateman S, Lee G, Piumsomboon T, Billinghurst M (2019) Revisiting collaboration through mixed reality: the evolution of groupware. Int J Hum–Comput Stud 131:81–98. https://doi.org/10.1016/j.ijhcs.2019.05.011

Feiner S, Shamash A (1991) Hybrid user interfaces: breeding virtually bigger interfaces for physically smaller computers. In: Proceedings of the 4th annual ACM symposium on user interface software and technology (UIST'91). ACM, Hilton Head, South Carolina, USA, pp 9–17. https://doi.org/10.1145/120782.120783

Fidalgo CG, Yan Y, Cho H, Sousa M, Lindlbauer D, Jorge J (2023) A survey on remote assistance and training in mixed reality environments. IEEE TVCG 29(5):2291–2303. https://doi.org/10.1109/TVCG.2023.3247081

Fuste A, Reynolds B, Hobin J, Heun V (2020) Kinetic ar: a framework for robotic motion systems in spatial computing. In: Extended abstracts of the 2020 CHI conference on human factors in computing systems (CHI'20). ACM, New York, NY, USA, pp 1–8. https://doi.org/10.1145/3334480.3382814

Ganesan R, Rathore Y, Ross H, Ben Amor H (2018) Better teaming through visual cues: How projecting imagery in a workspace can improve human-robot collaboration. IEEE Robot Autom Mag. https://doi.org/10.1109/MRA.2018.2815655

Ghamandi RK, Hmaiti Y, Nguyen TT, Ghasemaghaei A, Kattoju RK, Taranta EM, LaViola JJ (2023) What and how together: a taxonomy on 30 years of collaborative human-centered XR tasks. In: 2023 IEEE international symposium on mixed and augmented reality (ISMAR). IEEE, Sydney, Australia, pp 322–335. https://doi.org/10.1109/ISMAR59233.2023.00047

Grubert J, Heinisch M, Quigley A, Schmalstieg D (2015) MultiFi: multi fidelity interaction with displays on and around the body. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems (CHI'15). Association for Computing Machinery, Seoul, Republic of Korea, pp 3933–3942. https://doi.org/10.1145/2702123.2702331

Grubert J, Witzani L, Otte A, Gesslein T, Kranz M, Kristensson PO (2023) Text entry performance and situation awareness of a joint optical see-through head-mounted display and smartphone system. IEEE Trans Vis Comput Graph 30(8):1–16. https://doi.org/10.1109/TVCG.2023.3309316

Gruenefeld U, Prädel L, Illing J, Stratmann T, Drolshagen S, Pfingsthorn M (2020) Mind the arm: realtime visualization of robot motion intent in head-mounted augmented reality. In: Proceedings of the conference on mensch und computer (MuC'20). ACM, Magdeburg, Germany, pp 259–266. https://doi.org/10.1145/3404983.3405509

Hietanen A, Pieters R, Lanz M, Latokartano J, Kämäräinen J-K (2020) Ar-based interaction for human–robot collaborative manufacturing. Robot Comput Integr Manuf 63:101891. https://doi.org/10.1016/j.rcim.2019.101891

Hoang KC, Chan WP, Lay S, Cosgun A, Croft EA (2022) Arviz: an augmented reality-enabled visualization platform for ros applications. IEEE Robot Autom Mag 29(1):58–67. https://doi.org/10.1109/MRA.2021.3135760

Hubenschmid S, Zagermann J, Butscher S, Reiterer H (2021a) STREAM: exploring the combination of spatially-aware tablets with augmented reality head-mounted displays for immersive analytics. In: Proceedings of the 2021 CHI conference on human factors in computing systems (CHI'21). ACM, Yokohama, pp 1–14. https://doi.org/10.1145/3411764.3445298

Hubenschmid S, Zagermann J, Fink D, Wieland J, Feuchtner T, Reiterer H (2021b) Towards asynchronous hybrid user interfaces for cross-reality interaction. In: ISS'21 workshop proceedings: "transitional interfaces in mixed and cross-reality: a new frontier?". KOPS Universität Konstanz, Konstanz. https://doi.org/10.18148/kops/352-2-84mm0sggczq02

Hubenschmid S, Wieland J, Fink DI, Batch A, Zagermann J, Elmqvist N, Reiterer H (2022) ReLive: bridging in-situ and ex-situ visual analytics for analyzing mixed reality user studies. In: CHI conference on human factors in computing systems (CHI'22). ACM, New Orleans, LA, USA, pp 1–20. https://doi.org/10.1145/3491102.3517550

Hubenschmid S, Zagermann J, Leicht D, Reiterer H, Feuchtner T (2023) ARound the smartphone: investigating the effects of virtually-extended display size on spatial memory. In: Proceedings of the 2023 CHI conference on human factors in computing systems (CHI'23). ACM, Hamburg, pp 1–15. https://doi.org/10.1145/3544548.3581438

Ikeda B, Szafir D (2022) Advancing the design of visual debugging tools for roboticists. In: 2022 17th ACM/IEEE international conference on human–robot interaction (HRI). IEEE, Sapporo, pp 195–204. https://doi.org/10.1109/HRI53351.2022.9889392

Ikeda B, Szafir D (2024) Programar: augmented reality end-user robot programming. J Hum–Robot Interact 13(1):1–20. https://doi.org/10.1145/3640008. (**Just Accepted**)

Kim S, Dey AK (2009) Simulated augmented reality windshield display as a cognitive mapping aid for elder driver navigation. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI'09). ACM, Boston, pp 133–142. https://doi.org/10.1145/1518701.1518724

Knierim P, Hein D, Schmidt A, Kosch T (2021) The SmARtphone controller: leveraging smartphones as input and output modality for improved interaction within mobile augmented reality environments. i-com 20(1):49–61. https://doi.org/10.1515/icom-2021-0003

Langner R, Satkowski M, Büschel W, Dachselt R (2021) MARVIS: combining mobile devices and augmented reality for visual data analysis. In: Proceedings of the 2021 CHI conference on human factors in computing systems (CHI'21). ACM, New York, pp 1–17. https://doi.org/10.1145/3411764.3445593

Ledo D, Houben S, Vermeulen J, Marquardt N, Oehlberg L, Greenberg S (2018) Evaluation strategies for hci toolkit research. In: Proceedings of the 2018 CHI conference on human factors in computing systems (CHI'18). ACM, New York, pp 1–17. https://doi.org/10.1145/3173574.3173610

Lee B, Sedlmair M, Schmalstieg D (2023) Design patterns for situated visualization in augmented reality. In: IEEE TVCG, pp 1–12. https://doi.org/10.1109/TVCG.2023.3327398

Leiva G, Grønbæk JE, Klokmose CN, Nguyen C, Kazi RH, Asente P (2021) Rapido: prototyping interactive ar experiences through programming by demonstration. In: The 34th annual ACM symposium on user interface software and technology (UIST'21). ACM, New York, pp 626–637. https://doi.org/10.1145/3472749.3474774

Li W, Wang J, Jiao S, Wang M, Li S (2019) Research on the visual elements of augmented reality assembly processes. Virtual Real Intell Hardw 1(6):622–634. https://doi.org/10.1016/j.vrih.2019.09.006. (**VR/AR in industry**)

Lunding RS, Lystbaek MN, Feuchtner T, Grønbaek K (2023) Ar-supported human–robot collaboration: facilitating workspace awareness and parallelized assembly tasks. In: 2023 IEEE international symposium on mixed and augmented reality (ISMAR). IEEE, Sydney, Australia, pp 1064–1073. https://doi.org/10.1109/ismar59233.2023.00123

Lunding R, Lunding M, Feuchtner T, Petersen M, Grønbæk K, Suzuki R (2024) Robovisar: immersive authoring of condition-based ar robot visualisations. In: Proceedings of the 2024 ACM/IEEE international conference on human–robot interaction (HRI'24). ACM, Boulder, pp 462–471. https://doi.org/10.1145/3610977.3634972

Lunding RS, Feuchtner T, Grønbæk K (2025) Investigating ar assistance for human–robot collaboration in mould assembly "in the wild". In: Proceedings of the 2025 ACM/IEEE international conference on human–robot interaction

Malik AA, Pandey V (2022) Drive the cobots aright: guidelines for industrial application of cobots. In: International design engineering technical conferences and computers and information in engineering conference (IDETC-CIE2022). American Society of Mechanical Engineers, St. Louis. https://doi.org/10.1115/detc2022-90777

Marques B, Silva S, Alves J, Araujo T, Dias P, Santos BS (2022) A conceptual model and taxonomy for collaborative augmented reality. IEEE Trans Vis Comput Graph 28(12):5113–5133. https://doi.org/10.1109/TVCG.2021.3101545

Marques B, Silva S, Pedrosa E, Barros F, Teixeira A, Santos BS (2024) Towards unlimited task coverage and direct (far-off) manipulation in extended reality remote collaboration. In: Companion of the 2024 ACM/IEEE international conference on human–robot interaction (HRI'24). Association for Computing Machinery, New York, pp 745–749. https://doi.org/10.1145/3610978.3640737

Martins NC, Marques B, Alves J, Araújo T, Dias P, Santos BS (2022) Augmented reality situated visualization in decision-making. Multimed Tools Appl 81(11):14749–14772. https://doi.org/10.1007/s11042-021-10971-4

Microsoft (2025) Welcome to dynamics 365 guides. https://learn.microsoft.com/en-us/dynamics365/mixed-reality/guides/overview. Accessed 23 Jan 2025

Normand E, McGuffin MJ (2018) Enlarging a smartphone with AR to create a handheld VESAD (virtually extended screen-aligned display). In: 2018 IEEE international symposium on mixed and augmented reality (ISMAR). IEEE, Munich, pp 123–133. https://doi.org/10.1109/ISMAR.2018.00043

OASIS MQTT Technical Committee: MQTT. https://mqtt.org, May 09, 2024. https://mqtt.org Accessed 23 Jan 2025

Park J, Casser F, Schlette C (2023) Digitalizing manual processes using digital twins and product lifecycle management for safe human–robot interaction scenarios. In: 2023 28th international conference on automation and computing (ICAC). IEEE, Birmingham, pp 1–6. https://doi.org/10.1109/ICAC57885.2023.10275241

Pascher M, Gruenefeld U, Schneegass S, Gerken J (2023) How to communicate robot motion intent: a scoping review. In: Proceedings of the 2023 CHI conference on human factors in computing systems (CHI'23). ACM, New York, pp 1–17. https://doi.org/10.1145/3544548.3580857

Rashid U, Nacenta MA, Quigley A (2012a) Factors influencing visual attention switch in multi-display user interfaces: a survey. In: Proceedings of the 2012 international symposium on pervasive displays—PerDis'12 (PerDis 2012). ACM Press, Porto, pp 1–6. https://doi.org/10.1145/2307798.2307799

Rashid U, Nacenta MA, Quigley A (2012b) The cost of display switching: a comparison of mobile, large display and hybrid UI configurations. In: Proceedings of the international working conference on advanced visual interfaces (AVI'12). ACM, Capri Island, pp 99–106. https://doi.org/10.1145/2254556.2254577

Rasmussen T, Feuchtner T, Huang W, Grønbaek K (2022) Supporting workspace awareness in remote assistance through a flexible multi-camera system and augmented reality awareness cues. J Vis Commun Image Represent 89:103655. https://doi.org/10.1016/j.jvcir.2022.103655

Ratcliffe J, Soave F, Bryan-Kinns N, Tokarchuk L, Farkhatdinov I (2021) Extended reality (XR) remote research: a survey of drawbacks and opportunities. In: Proceedings of the 2021 CHI conference on human factors in computing systems (CHI'21). ACM, Yokohama, pp 1–13. https://doi.org/10.1145/3411764.3445170

Reipschläger P, Flemisch T, Dachselt R (2021) Personal augmented reality for information visualization on large interactive displays. In: IEEE TVCG, vol 27(no 2), pp 1182–1192. https://doi.org/10.1109/TVCG.2020.3030460; arXiv:2009.03237

Renner P, Lier F, Friese F, Pfeiffer T, Wachsmuth S (2018) Wysiwicd: what you see is what i can do. In: Companion of the 2018 ACM/IEEE international conference on human–robot interaction (HRI'18). ACM, New York, p 382. https://doi.org/10.1145/3173386.3177533

Rogers RD, Monsell S (1995) Costs of a predictible switch between simple cognitive tasks. J Exp Psychol Gen 124(2):207–231. https://doi.org/10.1037/0096-3445.124.2.207

Rosen E, Whitney D, Phillips E, Chien G, Tompkin J, Konidaris G, Tellex S (2019) Communicating robot arm motion intent through mixed reality head-mounted displays. Int J Robot Res 38(12–13):1513–1526. https://doi.org/10.1177/0278364919842925

Rosenholtz R, Li Y, Nakano L (2007) Measuring visual clutter. J Vis 7(2):17–17. https://doi.org/10.1167/7.2.17

Sääski J, Salonen T, Hakkarainen M, Siltanen S, Woodward C, Lempiäinen J (2008) Integration of design and assembly using augmented reality. Springer, Boston, pp 395–404. https://doi.org/10.1007/978-0-387-77405-3_39

Sereno M, Wang X, Besancon L, Mcguffin MJ, Isenberg T (2020) Collaborative work in augmented reality: a survey. IEEE Trans Vis Comput Graph 1:1. https://doi.org/10.1109/TVCG.2020.3032761

Suzuki R, Karim A, Xia T, Hedayati H, Marquardt N (2022) Augmented reality and robotics: a survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces. In: CHI conference on human factors in computing systems (CHI'22). ACM, New York, pp 1–33. https://doi.org/10.1145/3491102.3517719

Universal Robots (2024a) RTDE client library—Python Feb 22, 2024. https://github.com/UniversalRobots/RTDE_Python_Client_Library. Accessed 23 Jan 2025

Universal Robots (2024b): Robot arm Technical specification. May 09, 2024. https://www.universal-robots.com/media/1826690/01_2023_collective_data_sheet-1.pdf. Accessed 23 Jan 2025

Walker M, Phung T, Chakraborti T, Williams T, Szafir D (2023) Virtual, augmented, and mixed reality for human-robot interaction: a survey and virtual design element taxonomy. J Hum Robot Interact 12(4):1–39. https://doi.org/10.1145/3597623

Wieland J, Cho H, Hubenschmid S, Kiuchi S, Reiterer H, Lindlbauer D (2024) Push2ar: enhancing mobile list interactions using augmented reality. In: 2024 IEEE international symposium on mixed and augmented reality (ISMAR) (ISMAR'24). IEEE, Seattle, pp 671–680. https://doi.org/10.1109/ISMAR62088.2024.00082

Zagermann J, Hubenschmid S, Balestrucci P, Feuchtner T, Mayer S, Ernst MO, Schmidt A, Reiterer H (2022) Complementary interfaces for visual computing. Inf Technol 64(4–5):145–154. https://doi.org/10.1515/itit-2022-0031