Empowering Users to Create Augmented Reality-Based Solutions – Deriving Design Principles for No-Code AR Authoring Tools

Julia Bräker
University of Hamburg, julia.braeker@uni-hamburg.de

Julia Hertel
University of Hamburg, julia.hertel@uni-hamburg.de

Martin Semmann
University of Hamburg, martin.semmann@uni-hamburg.de

Empowering Users to Create Augmented Reality-Based Solutions – Deriving Design Principles for No-Code AR Authoring Tools

Completed Research Paper

Julia Bräker
University of Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg
julia.braeker@uni-hamburg.de

Julia Hertel
University of Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg
julia.hertel@uni-hamburg.de

Martin Semmann
University of Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg
martin.semmann@uni-hamburg.de

Abstract

Grounded on an experimental study with 18 participants, we derive 15 design principles for no-code AR authoring tools in an organizational setting. The study consists of two distinct treatments that aim to augment lightweight processes with AR. The outcomes are two interactive tutorials utilizing AR instructions. Following the no-code approach, the participants were empowered to create relevant AR content using a reduced interface and no need for advanced configurations or coding. The study thus combines two research streams with the aim of better understanding mechanisms for AR use in a professional context. As prior work has shown, despite the potential benefits, the adoption of AR authoring tools is limited because ramping up AR to productive use is heavily dependent on consulting and custom software solutions. Our novel approach bears the potential to broaden application domains and empower professionals to apply AR.

Keywords: User study, mixed reality, thinking aloud, digital innovation, IT adoption

Introduction

As the pace of innovation cycles increases, organizations face the challenge of utilizing novel approaches to realize underlying benefits (Wehking et al., 2021). Likewise, users need to cope with all such ever-changing environments, and a need for adaptation arises. This area of tension results in failed innovation projects, abandoned proof of concepts or increased reluctance of employees. A lever to overcome these issues is seen in no-code and low-code platforms because they aim to empower users to design and apply innovative solutions within their workplace (Atkins, 2020; Bock & Frank, 2021). User-driven innovations are core characteristics of such platforms (Elshan et al., 2023). By providing a simplified interface and pre-built components, no- and low-code platforms allow users with no or little programming experience to create individual solutions. This reduces the reliance on IT departments and allows for faster implementation of ideas (Bock & Frank, 2021). An innovative technology that faces challenges in its dissemination in organizational contexts is augmented reality (AR). Especially for instructions and process support, AR offers a wide range of potential applications (Bräker, Osterbrink, et al., 2023; Hertel et al., 2022). However, adoption in business contexts remains scarce. Developments using game engines like Unity are challenging for inexperienced, non-technical designers. Professional AR authoring tools find low acceptance since they require a certain level of programming knowledge and much practice. AR authoring tools aim to create AR applications without programming experience. For example, organizations can discover whether AR adoption would be beneficial without enormous hurdles and financial investments. Existing AR authoring tools require less programming experience but are no less training-intensive and complex in their usage (Hönemann et al., 2022; Nebeling & Speicher, 2018). Therefore, we seek to understand the underlying design principles of no-code AR authoring tools to make them more accessible. Accordingly, we address the following research question:

RQ: How can no-code AR authoring tools be designed to enhance accessibility for users?

We approach this problem with an experimental user study with 18 participants that aims to design AR instructions for two exemplary processes. The no-code AR authoring tool enables users to define processes and attach AR elements to instruction steps. The outcomes are AR-based process descriptions that could be used for training purposes or to ensure a standardized process flow. Based on the study results, we derive 15 design principles in eight dimensions that guide the development of AR authoring tools that are accessible to users and do not need expert knowledge. To the best of our knowledge, this is the first study that combines AR authoring and no-code platforms to derive design knowledge for the future design of such platforms. In this way, we contribute to the understanding of no-code AR authoring tools in a theoretical way and to the concrete design knowledge at a practical level.

The remainder is structured as follows: The next section covers related work on AR and no-code/low-code platforms. After that, we present the research design with particular regard to the data collection, data analysis and derivation of the design principles. We then present our results in depth and discuss these while mirroring the state of the art. We conclude the paper with an outlook.

Related Work

Mixed reality technologies range from entirely virtual to real environments with varying degrees of virtual and real characteristics. Augmented reality (AR) extends reality with computer-generated elements, whereas virtual reality (VR) is fully immersive (Milgram & Kishino, 1994). The merging of the virtual and real world, real-time interaction, and the registration of virtual content in three-dimensional space are the characteristics of AR (Azuma, 1997).

The ability to enhance reality with virtual elements holds enormous potential for application in organizational settings. AR has been proven to assist in manufacturing, maintenance, and inspection processes such as assembly tasks (Kohn & Harborth, 2018) or general mechanical engineering tasks (Kammler et al., 2019). Besides, AR is used to assist safety-critical services (Bräker, Osterbrink, et al., 2023; Osterbrink et al., 2021), education (Mohammadhossein et al., 2022), and healthcare (Klinker et al., 2019). However, previous research shows that the development of AR applications remains challenging (Ashtari et al., 2020). Especially small and medium enterprises (SMEs) struggle to realize these potentials (Cranmer et al., 2021; Masood & Egger, 2019). The reasons for this are manifold. Limited financial resources and little experience may make it difficult to invest in AR technologies. Other challenges include the complexity of identifying and selecting potential processes and the lack of guidance in process modeling (Bräker & Semmann, 2021, 2022, 2023).

AR authoring tools for enterprises aim to create AR applications without intensive knowledge of software development. They focus on a low-code or no-code manner for creating AR content. AR content creation is multifaceted and includes, for example, placing virtual content in 3D space, creating visualizations and models, and including interactions. The authoring can be done using various hardware devices but is often desktop-based. Low-code development platforms are seen as innovation drivers because they make technologies accessible to new target groups (Elshan et al., 2023). This makes AR more accessible for SMEs, and proof of concepts can be implemented more quickly. However, existing AR authoring tools on the market require long training periods, and licenses are expensive. They come with numerous challenges and limitations. As a result, they are often operated by authoring experts. They do not cover encompassing design space. Additionally, AR authoring tools often come with a unique toolchain that makes users dependent, or they end up with a patchwork of tools (Hönemann et al., 2022; Nebeling & Speicher, 2018). An analysis across 26 AR authoring tools by Dengel et al. (2022) supports this impression. Most tools on the market require programming skills, while half of the 26 tools are not freely accessible or do not provide enough interactivity.

Research Design

We follow the research design shown in Figure 1 to generate knowledge about the design of AR authoring tools. The starting point for our study is an AR authoring tool developed as part of a research project. With its as-is solution state, the authoring tool aims to allow users to create customized AR instruction apps. We tested the AR authoring tool in a user study in order to be able to make statements about requirements and design principles as well as the usability and usefulness of the current prototype version. Users had to create AR instructions for two different processes. We combined qualitative and quantitative methods, thus following a mixed-method approach (Venkatesh et al., 2016). With established questionnaires, we measured usability, user experience, and workload. We analyzed the questionnaires with inferential statistics. We followed the think-aloud approach, which allowed us to make qualitative statements about the AR authoring tool. We recorded the screen and spoken words to document the think-aloud process. We coded the think-aloud recordings and the open questions from the questionnaires. From the codes, we derived requirements, from which we finally derived the design principles.

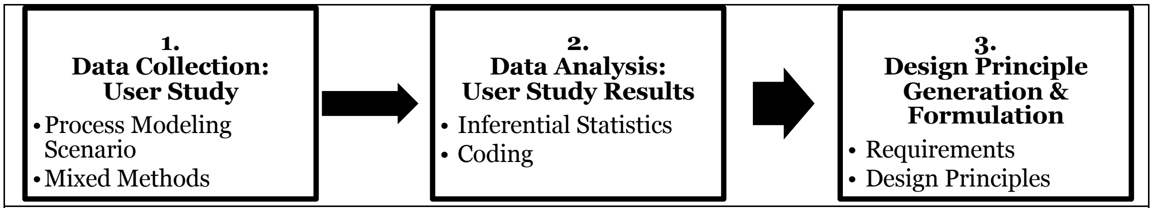


Figure 1. Research Design

User Study

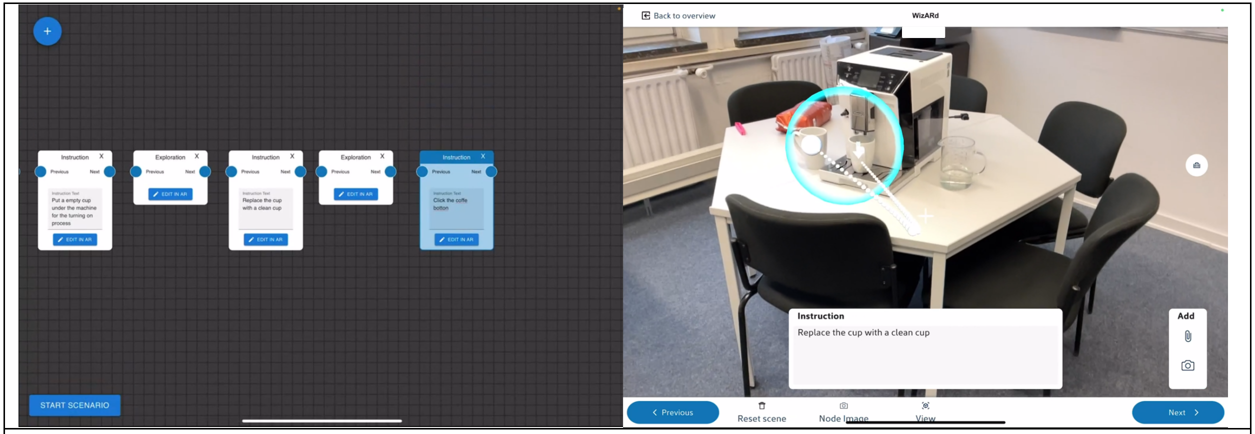
We conducted a user study during which participants interacted with the prototypical AR authoring tool to create in situ AR instructions. The goal of the application is to be deployed in organizations to facilitate the usage of AR by providing a no-code alternative for creating AR instructions. This study aims to get insights into how an AR authoring tool needs to be designed. By evaluating the status quo of the AR authoring tool, we aim to investigate whether the prototype is already a suitable alternative to programming AR instructions from scratch or whether there is improvement potential. Participants of the study were computer science students and members of the department. Most of them had programming experience. Thus, it allows them to assess their preference and the estimated effort of using the prototype compared to programming a similar app.

Besides, we focused on assessing the usability, user experience, induced workload, and other subjective opinions, thoughts, and ideas about the prototype. These metrics might be influenced by the AR authoring tool's design. We aimed to assess if the prototype was intuitive to use, to what extent it matched the participants' expectations and mental models, and which difficulties occurred when used by untrained users. By doing so, we aim to elicit specific information about usage behavior that we can use to derive design knowledge for the design of such AR authoring tools. To avoid influencing and priming the study participants, we did not provide tutorials or explicit instructions on using the app beforehand. Instead, participants freely explored the prototype while their thoughts were captured using the think-aloud protocol.

Prototypical AR Authoring App

As part of previous research, a mobile application was developed that allows users to create instructions for AR processes in a no-code manner (Konopka et al., 2022). In contrast to traditional AR authoring tools, which are complex and require much training, this tool aims to create AR instructions without training or programming skills. The tool is characterized by the fact that the instructions can be created directly in AR based on the "what you see is what you get" (WYSIWYG) principle. The instruction can be created and viewed on the same device, for example, using a tablet computer.

The app has three main screens: (1) A node editor to model the process (see Figure 2, left). Thereby, one node represents one process step. (2) An AR editor, in which the instruction AR content for the process steps can be created and edited directly in AR (see Figure 2, right). (3) An AR scenario preview, in which the user can view the result, i.e., the AR instructions. The node editor can be used to create different node types. The info node creates a 2D canvas with a title and a textual description. The exploration node allows displaying different points of interest (POIs) in AR. Two AR elements can be added to the AR exploration scene. The first is a simple marker with a location icon, and the second is a tether with the same marker at the end. For each of the POIs, a textual description can be added. The third node, the instruction node, aims to create process steps for instructions. This node has an instruction text for the scene, and a wider choice of AR elements is available. Arrows, two different tethers, sticky nodes, and a halo, i.e., a transparent circle, can be placed in the AR scene. The different nodes can be connected in the node editor screen to create the process flow. The AR scenario preview screen shows the resulting outcome – the AR instruction app. The user can view the created instructions step by step and check how the result will look.



Instruction Text
Put a empty cup under the machine for the turning on process

Instruction Text
Replace the cup with a clean cup

Instruction Text
Click the coffe botton

Instruction
Replace the cup with a clean cup

Figure 2. AR Authoring Tool, Populated Node Editor (Left) and AR Editor With Several AR Elements in the Scene (Right)

Task

To investigate the suitability of the prototype for different use cases, we included two scenarios in this experiment for which participants were asked to create AR instructions using the AR authoring app:

Scenario 1 – Coffee machine. This scenario included creating AR instructions for typical steps to prepare a coffee machine, like filling in coffee beans and water, placing a cup, plugging in the power plug and pressing a button to make a coffee. Here, the coffee machine was the only point of interest. All required items (coffee beans, water, cup) were placed on a table beside the coffee machine. The table was placed in the middle of the room to allow participants to move around it freely. For instance, to see the power plug, participants had to walk around the machine to see its back side.

Scenario 2 – Seminar room. In contrast to the first scenario, participants had to move around a room to create AR instructions while points of interest were placed around them. Participants were asked to create instructions on how to prepare a university's room for a seminar, including steps like connecting a notebook to a projector, turning on the projector on a wall-mounted interface, closing windows and writing the current date on a blackboard.

We aimed to let participants freely convert a process into instructions without explicitly providing the division into single steps. This way, we mimicked a situation of implicit knowledge, e.g., an expert knowing how to perform a process without it being written in a handbook, freely explaining it to a new worker. Thus, we decided not to provide the process text-based since such a presentation would inherently structure the process into distinct steps. Instead, we recorded videos to demonstrate the processes. For these videos, we filmed a person performing the processes, i.e., preparing the coffee machine and preparing the seminar room as described above. This way, participants had to convert the process into distinct steps and these steps into an instruction by themselves. For instance, they had to decide if "closing windows" could be one step or if they created a single step for each window that must be closed. The videos did not include auditory or textual descriptions of the process but simply showed its execution in one take. Based on these videos, participants were asked to create AR instructions for the shown process, i.e., use the app like a professional in an organization would use it to create an instruction for a process. We explicitly stressed that they should not recreate the process (e.g., not making a coffee) but instead create instructions for the process using the prototype's features.

Procedure

Before starting the study, participants signed an informed consent form and were asked to fill out a questionnaire about demographics, their experience with AR and their experience as developers. Then, they read a short informational text about AR and the prototypical AR authoring tool. To familiarize themselves with the prototype, they were handed a tablet with the prototype running. They were asked to explore the application freely. We particularly encouraged them to stand up and explore the spatial functionality of the prototype. The participants were allowed to finish this exploration as soon as they felt ready but were given a maximum of 15 minutes. After the exploration, they started with the first scenario. The order in which participants completed the two scenarios was counterbalanced to avoid any learning or fatigue effects coupled with a specific scenario. Thus, half of the participants started with the coffee machine scenario and the other half with the seminar room scenario. For each scenario, the participant first watched the video demonstrating the process. Participants were allowed to watch the video as often as they wanted before creating the instructions. They were also allowed to watch it again during the instruction creation process. They could freely slide through the video with a standard video player interface. After watching the video, they were again handed the tablet with the prototype running and an empty scenario and were asked to create instructions for the given process. For each process, a maximum of 30 minutes was given. Depending on whether the participant quickly finished the conditions or used the maximum time, the study took about 60 to 90 minutes.

Measurements

After each scenario, they were asked to complete an unweighted NASA Task Load Index (NASA TLX) questionnaire (Hart, 1986; Hart & Staveland, 1988). After completing the second scenario, they were also asked to fill out the System Usability Scale (SUS), a 10-item scale to measure usability (Lewis, 2018), the short version of the User Experience Questionnaire (UEQ), an 8-item scale to assess the pragmatic and hedonic quality of a product (Schrepp, 2015; Schrepp et al., 2017). Pragmatic quality describes how far users can reach their goals with the product. Hedonic quality focuses on aspects like novelty and fun while using the product. During the exploration mode and both conditions (scenarios), participants were asked to follow the think-aloud protocol (Van Someren et al., 1994), i.e., speaking out loud their thoughts while interacting with the prototype. Their voice was recorded using the tablet's built-in microphone. We also stored screen recordings to analyze the usage of the AR authoring prototype. Additionally, the completion time per condition was logged.

Participants

The study was completed by 18 participants (7 female, 10 male, 1 preferred not to tell). The participants' age ranged from 20 to 35 years (M = 26.3, SD = 4.7). The participants were students (N = 16) or staff members (N = 2) from a university's computer science department. When asked how often they use AR applications, most (N = 11) reported that they have used AR a few times, 5 participants have never used AR, and 2 participants use AR at least monthly. On a scale from 1 (unexperienced) to 5 (expert), they rated their experience with different types of AR as follows: mobile AR (M = 2.8, SD = 0.8), head-mounted displays (HMDs) (M = 1.8, SD = 1.1), spatial AR (M = 1.3, SD = 0.6). One participant has used AR for instruction or guidance before, 5 for design or planning, 5 for games, 2 for education and one for social media. We also assessed the participants' developer experience on a scale from 1 (unexperienced) to 5 (professional) as a developer in general (M = 2.5, SD = 1.2), as an AR developer (M = 1.2, SD = 0.5) and as a game engine developer (e.g., Unity) (M = 1.7, SD = 1.0).

Data Analysis

We collected data from 18 participants, who each performed two conditions (scenarios). In the following, we describe how we analyzed all collected data, i.e., completion times, questionnaire results and think-aloud recordings.

Completion Time

We investigated whether the time participants needed to create the instructions differed between the first and second conditions they completed to investigate if they got faster in creating instructions over time. In the following, times are reported in minutes. Due to technical issues, some time values were not stored successfully, which led to the exclusion of the data of 4 participants from the completion time analysis. For the remaining 14 participants, all the time groups were normally distributed. We performed a paired-sample t-test and found a significant difference between the first condition (M =21.5, SD = 5.14) and the second condition (M = 14.99, SD = 5.34) at the 5% significance level (t(13) = 2.97, p < 0.05), indicating a higher completion time for the first condition than for the second one.

Usability, User Experience and Workload

To investigate the prototype's usability, we used the SUS. Following the calculation method proposed by Brooke (1996), the application was rated with a mean usability score of M = 40.4 (SD = 17.0) on a scale from 0 to 100, indicating below-average usability (Lewis, 2018). The user experience was measured using the short version of the UEQ. Transformed to a scale from -3 to 3, we measured a pragmatic quality of 0.25, a hedonic quality of 1.0, and an overall user experience of 0.625. These values indicate a neutral pragmatic quality, a positive hedonic quality, and a neutral overall user experience.

After each condition, we measured the perceived workload with the unweighted NASA TLX. Figure 3 (left) shows the results, grouped by scenario. In order to investigate if the workload differs between both evaluated scenarios (i.e., coffee machine and seminar room), a paired-sample t-test was conducted. Values were log 10 transformed before the resulting residuals were confirmed to follow a normal distribution using the Shapiro-Wilk test. We found no significant difference between the coffee machine scenario (M = 36.21, SD = 19.62) and the seminar room scenario (M = 35.08, SD = 16.22) at the 5% significance level. To get a deeper insight, we investigated differences per dimension. The data did not follow a normal distribution for multiple dimensions and could not be transformed to be normally distributed with standard transformations. Hence, two-sided Wilcoxon signed-ranks tests were performed for each dimension. We found a significantly higher mental demand in the coffee machine scenario than in the seminar room scenario (Z = 96, p < 0.05). For the other dimensions, we did not find a significant difference.

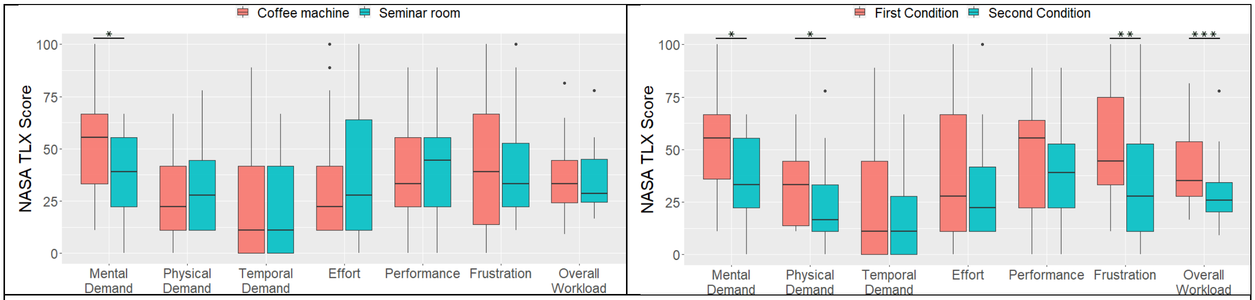


Figure 3. Measured Workload Grouped by Scenario (Left) and Condition Number (Right)

Asterisks depict significant differences (* < 0.05, ** < 0.01, *** < 0.001)

We also investigated possible learning and fatigue effects by comparing the workload of the first condition a participant completed to the second one. The grouped workload values can be seen in Figure 3 (right). Again, a paired-sample t-test was conducted, and values were log 10 transformed before the resulting residuals were confirmed to follow a normal distribution using the Shapiro-Wilk test. We found a significant difference between the first condition (M = 40.74, SD = 17.35) and the second condition (M = 30.56, SD = 17.12) at the 5% significance level (t(17) = 4.66, p < 0.001). The results suggest that the overall workload was higher in the first condition than in the second one. Again, we performed dimension-wise comparisons. The data did not follow a normal distribution for multiple dimensions and could not be transformed with standard transformations to be normally distributed. Hence, two-sided Wilcoxon signed-ranks tests were performed for each dimension. We found a significantly higher mental demand (Z = 115, p < 0.05), higher physical demand (Z = 65.6, p < 0.05), and higher frustration (Z = 112, p < 0.01) for the first condition than for the second one. For the other dimensions, we did not find a significant difference.

Participants' Feedback and Preference Estimation

Figure 4 shows the answers to the further questions that do not stem from a standardized questionnaire regarding the participants' opinions about the app's difficulty, complexity, and suitability. Each question was asked on a 5-point semantic differential scale. In the questionnaire used in the study, some questions had different positive-negative directions (e.g., difficulty was rated from 1 – easy to 5 – difficult). For a clearer presentation, we inverted some scales to reach a universal direction (i.e., a higher value indicating a more positive response). For each question, participants could optionally add a comment. We performed a one-sided Wilcoxon signed-ranks test against the neutral value of 3 for each question. For difficulty, we found a significant effect (Z = 73.5, p < 0.05) indicating a trend toward "easy" (M = 3.56, SD = 0.99). Participants significantly preferred an AR app when asked about their preference for using an AR authoring tool compared to a desktop interface (M = 4.33, SD = 0.69; Z = 136, p < 0.001). According to the comments, participants appreciate the in situ / WYSIWYG aspect (N = 6) and suggest combining AR and desktop (N=3). In contrast, the desktop could be used for more precise positioning or labeling. Participants significantly perceived mobile AR as suitable (M = 3.94, 0.88; Z = 100, p < 0.01). Three Participants mentioned the usage of an HMD, whereas one participant discussed concerns regarding cybersickness. They also proposed using a smaller device (N= 4) like a smartphone to reduce fatigue while holding the device and facilitate the text input. They also discussed the advantage of the tablet's large screen compared to smaller devices. Regarding a preference for using such a no-code authoring app instead of programming an app to show similar instructions, participants preferred the app (M = 4.67, SD = 0.49; Z = 171, p < 0.001). One participant described their preference as context-dependent and would only prefer the app for a static environment and a linear task. When comparing the time they would use for creating instructions either with the app or by programming, we observed a trend towards a longer expected time for programming (M = 4.61, SD = 0.98; Z = 160, p < 0.001). We also asked the participants to estimate the time they would expect to spend programming an app that shows similar instructions to the ones they authored in the study, given the example of the Unity engine for programming. We did not provide any time unit to avoid bias but let them freely describe their time estimation. The answers covered some hours, days, weeks, and even months. The smallest estimations were "2 hours", "probably several hours of concentrated works", "probably a day or so [...]", and "36 hours", the remaining answers clearly exceed a range of hours and days. Participants rated the app's complexity as balanced, as the answers did not significantly indicate a rate of "too complex", or "too basic" (M = 3.06, SD = 0.94; Z = 25, p = 0.8). Five participants suggested one or more additional features like logic gates (N = 1), an overview of all AR elements (N = 1), rotatable objects (N = 1), adding videos (N = 1), and blending out nodes (N = 1), and one participant suggested to limit the range of functionalities. We did not find a significant effect regarding the participants' opinion of whether they would be more suitable for small or large areas (M = 2.61, SD = 0.98; Z = 12, p = 0.1). Some participants reason their opinion for larger environments with tracking inaccuracies and argue that these issues would matter less in larger environments (N = 3).

Analysis of the Think Aloud Recordings

For the analysis of the think-aloud recordings, we were interested in the users' thoughts they spoke out loudly on the one hand and the usage behavior, i.e., what they do with the app and how they use it, on the other hand. Because of technical problems, we could not properly analyze the videos from three of the 18 participants. We excluded them from the analysis. We transcribed all video recordings of the remaining 15 participants. We enriched the transcripts with descriptions of the user behavior.

We coded the transcripts with two independent researchers following the grounded theory open coding method (Strauss & Corbin, 1994). This resulted in 144 different initial atomic codes. We then clustered these codes thematically into concepts and merged duplicates. This results in 140 concepts that describe challenges, problems, and suggestions for improvement. We derived 47 requirements for the AR authoring tool from the codes and concepts describing challenges, problems, and suggestions[1]. We have generated and formulated 15 design principles for AR authoring apps based on these requirements.

[1] All codes, concepts, requirements and resulting design principles are available at (Bräker, Hertel, et al., 2023): http://doi.org/10.25592/uhhfdm.13222

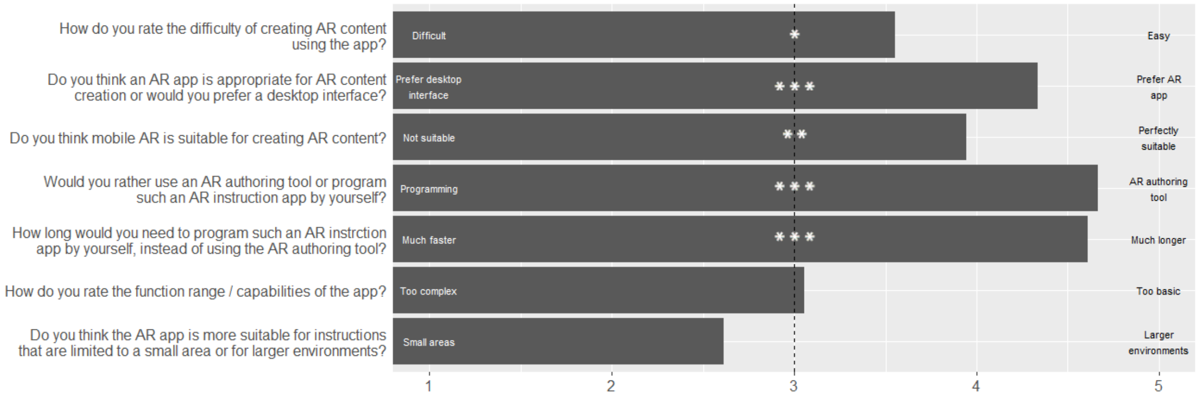


Figure 4. Mean Values of the Participants' Answers on Estimations and Preference Evaluations

Asterisks depict values significantly different from the neutral value of 3 (* < 0.05, ** < 0.01, *** < 0.001)

Design Principle Generation and Formulation

Design principles are "prescriptive statements that indicate how to do something to achieve a goal" (Gregor et al., 2020, p. 1622). They can focus on user activity, i.e., what users should be able to do with an artifact, focus on the artifact with its features, or both. Design principles are formulated in a standardized syntax. We follow the schema by Gregor et al. (2020):

"DP Name: For Implementer I to achieve or allow for Aim A for User U in Context C, employ Mechanisms M1, M2, … Mn involving Enactors E1, E2, … En because of Rationale R."

Design principles can be generated from newly designed artifacts (Purao et al., 2020). They can be inspired by what worked in the artifact and what did not work as expected. This can be evaluated, e.g., by conducting a user study. Our data basis for generating the design principles consisted of design requirements derived from the think-aloud recordings. We validated and enriched the design requirements with statements from the questionnaires. In the questionnaire, we asked the users for additional opinions, estimations and preferences. These are related to difficulties in using the app, the suitability of a tablet as hardware, the functional scope of the app and how likely they are to use it. The answers from the questionnaires go in line with the requirements from the think-aloud data. We aggregated the requirements by grouping similar goals. As Möller et al. (2020) proposed, we mapped the design requirements to design principles. Based on this, we formulated the design principles. Thus, every design requirement is mirrored in at least one design principle. According to the definition, the design principles describe what and how something must be done to achieve these goals. The resulting design principles are described in more detail in the following section.

Result – Design Principles for AR Authoring Tools

Our result is 15 design principles for AR authoring tools that emerged from the user study. These principles serve as a guideline for developing AR authoring applications. Table 1 shows the design principles. Each design principle has a title and a description, according to (Gregor et al., 2020). The design principles can be grouped into eight dimensions. These are principles for an ergonomic app usage (DP1), consideration of expectation management (DP2), guiding principles for improving understanding of the app (DP3 and DP4), basic interaction design principles (DP5-DP7), principles to support process modeling (DP8), principles for enabling comprehensive information coding in the AR app (DP9-DP11), principles for positioning AR elements (DP 12 and DP13), and for tracking the 3D environment (DP14 and DP15). In the following, we describe each design principle in more detail. We give insights and examples from our user study.

	Title	Design Principle
Ergonomics	**DP1:** Consider hardware requirements and limitations	For users to achieve a seamless and ergonomically comfortable experience using the app with a tablet, app design must avoid *(1)* holding the tablet in their hands continuously for too long while *(2)* not putting the tablet down too often because physical exhaustion due to the weight of the hardware and media breaks should be avoided.
Expectation Management	**DP2:** Manage and shape expectations for app use and outcomes in the beginning	To ensure that users do not develop false expectations and are satisfied with their outcomes, ensure expectation management in regard to *(1)* goals, tasks, and time required for app use, and *(2)* the possible outcome in the beginning because otherwise users will be disappointed and doubt their abilities.
Guidance	**DP3:** Limit free exploration to guide familiarization with the app	To allow users to use the app with little prior knowledge, employ mechanisms that provide users with guidance and assistance while leaving room for free and creative exploration because users struggle to understand the app without any guidance.
	DP4: Provide tutorials, tooltips, workflow patterns, and best practices	To allow users to acquire basic knowledge about the app, learn from existing knowledge, and get help when they encounter difficulties, employ *(1)* appropriate tutorials, *(2)* tooltips as help mechanisms, *(3)* example workflow patterns, and *(4)* best practices, because otherwise users will feel helpless, frustrated, and doubt their abilities.
Basic Interactions	**DP5:** Consider traditional interaction design and usability patterns	For the app to achieve usability, employ *(1)* traditional interaction design and usability patterns like feedback, different input mechanisms, assistance, and transparency, and *(2)* expect users to make mistakes or change their minds and therefore employ an undo and redo feature because otherwise low usability and user experience with the app leads to frustration and discomfort.
	DP6: Don't let users do unnecessary tasks that can be automated	To allow users to have a seamless experience with the app, employ mechanisms that automate unnecessary tasks because the user does not know they need to do the task, or it annoys them to do it repeatedly.
	DP7: Empower power users	To allow experienced users to use the app efficiently and effectively, employ mechanisms that empower them to quickly complete their goals because otherwise, the app seems inconvenient.
Process Modeling	**DP8:** Enable the modeling of complex processes	To enable users to model their processes as realistically as possible, enable them to (1) construct more complex processes with branches and conditions and not only linear processes and (2) combine different node types because otherwise, not all processes can be modeled correctly.
Extensive Information Encoding	**DP9:** Enable textual descriptions to enrich AR elements	To allow users to deliver comprehensive information, employ mechanisms to create additional textual descriptions for scenes, AR elements, and nodes because, in some cases, simple geometric AR elements are not expressive enough.
	DP10: Provide a variety of AR elements and style options	To allow users to deliver information in a diverse way, provide a variety of AR elements to choose from and allow individual styling options in terms of color, size, and rotation because the distinction by geometric shapes alone is not expressive enough.

	DP11: Allow adding, editing, and enriching the scene with photos and videos	To allow users to deliver information in the form of real-world snapshots, employ a mechanism for adding photos and videos and enriching/editing them because photos and videos are a simple option to capture a current state or target state of real-world objects.
AR Element Positioning	**DP12:** Facilitate the spatial perception of AR elements	To support users in the precise positioning of AR elements, employ mechanisms that (1) give feedback on surface detection and (2) support distance estimation because the placement of AR elements requires a clear idea of the spatial position of the element.
	DP13: Support precise positioning of AR elements	To allow users to create spatially accurate AR instructions, employ an interface that enables precise placement of AR elements because the position of AR elements mainly encodes instruction information and is an essential feature of the app.
Tracking	**DP14:** Tracking accuracy is essential	For users to achieve a sense of reliability, employ accurate and reliable tracking mechanisms because with unstable tracking, users will become frustrated, and the usefulness of instructions cannot be guaranteed.
	DP15: Implement object tracking	For users to achieve a feeling of blending real and virtual contents, employ mechanisms to track dynamic real-world objects because then AR elements can be attached to moving real-world objects.

Table 1. Design Principles for AR Authoring Tools

DP1: Consider hardware requirements and limitations. For users to achieve a seamless and ergonomically comfortable experience using the app with a tablet, app design must avoid *(1)* holding the tablet in their hands continuously for too long while *(2)* not putting the tablet down too often because physical exhaustion due to the weight of the hardware and media breaks should be avoided.

Hardware choice influences the design of the app thoroughly. Using a tablet for AR applications requires the users to hold the tablet in their hands for a long time, which was reported as heavy and exhausting. Especially when the AR elements are placed in the 3D environment, the users must hold the tablet straight up in front of them to view the environment through the camera. Doing tasks requiring both hands is not possible. Doing one-handed tasks, e.g., taking a photo and showing something with the other hand simultaneously, is possible but causes quicker exhaustion. Some participants tended to put down the tablet to model the process or type longer texts. However, they also described these media breaks, i.e., putting the tablet down and lifting it again, as exhausting and annoying. Thus, the app design must avoid holding the tablet for extended periods, while on the other hand, laying it down in between should be minimized to prevent media breaks.

DP2: Manage and shape expectations for app use and outcomes in the beginning. To ensure that users do not develop false expectations and are satisfied with their outcomes, ensure expectation management in regard to *(1)* goals, tasks, and time required for app use, and *(2)* the possible outcome in the beginning because otherwise users will be disappointed and doubt their abilities.

Expectation management regarding the app and the outcome can prevent users from being frustrated. It is essential to clarify the AR authoring app's capabilities and limitations, the expected time required to create AR instructions, and the expected level of prior knowledge. Our results show that participants needed much more time than expected, which frustrated them. Also, regarding the outcome – i.e., the AR instruction app the users create, not all users were satisfied with their results because they had higher expectations. They did not see an example outcome before and had no example comparison. Managing and shaping these expectations might help users be more satisfied with their performance and abilities. One participant said that the created result would not be helpful for other people because it did not look like an AR app.

DP3: Limit free exploration to guide familiarization with the app. To allow users to use the app with little prior knowledge, employ mechanisms that provide guidance and assistance while leaving room for free and creative exploration because users struggle to understand the app without any guidance.

Finding the sweet spot of guidance with room for free and creative exploration must be explored. Without guidance, participants quickly felt lost during app use - especially those with low prior knowledge. On the other hand, with too much guidance, users might feel restricted in accomplishing their goals. For example, predefined templates for recurrent tasks may guide and assist the users in app usage.

DP4: Provide tutorials, tooltips, workflow patterns, and best practices. To allow users to acquire basic knowledge about the app, learn from existing knowledge, and get help when they encounter difficulties, employ *(1)* appropriate tutorials, *(2)* tooltips as help mechanisms, *(3)* example workflow patterns, and *(4)* best practices, otherwise users will feel helpless, frustrated, and doubt their abilities.

Our results show that users require tutorials and help functions in the app. Participants asked for initial tutorials describing the app's general functions and concepts. The app's concepts did not directly match their mental model. Not understanding the different screens (concept node editor, concept AR editor, concept scenario preview) and the navigation between them, participants reported feeling lost and not knowing how to start. An approach to overcome these challenges could be an introductory tutorial in the form of a video or text. Further, participants asked for help because they struggled with specific features. Not all buttons and functions are self-explanatory. It is not apparent how the different types of nodes work and which functionalities they come with. Some also struggled to find the toolbox to select different AR elements and how to interact with them. The differences between node images, media and the photo button were also unclear and took some time to explore and understand. Especially new concepts like the selection cursor or the hand-moving button might need special attention because users are unfamiliar with these concepts from other apps. Helping functions could be implemented using helping buttons that open overlays with short explanations.

Although participants struggled initially, they experienced a learning effect. The longer they used the app, the faster and more confident they were in handling the app. This is underlined by the fact that they needed less time for the second task of the study. The orientation phase, in the beginning, could be shortened by providing appropriate helping mechanisms.

To encourage users even more, workflow patterns and best practices could be supportive. Example patterns could suggest how to navigate and use the app efficiently. For example, some participants tried to improve the workflow by modeling the process in the node editor as a first step. Then, they edited the scenes in the AR editor sequentially, and last, they viewed the results in the AR scenario preview. In contrast, most participants added one node in the node editor and then edited this node in AR. Afterward, they returned to the node editor and added the next node. In between, they checked how their outcome looked in the AR scenario preview. This approach took longer because they had to jump back and forth between the different screens more often. Best practices could, for example, show which node combinations are useful for which tasks. Some participants added info nodes at the beginning and end to give basic information. In between, they used instruction nodes for the different tasks. Best practices could also describe which AR elements can be combined well. For example, tether elements were often used to navigate the way to a spot in the real world. The tether can be followed by other AR elements showing the task step at this place. Additionally, some attached an image of the end state to show what the result should look like. Another pattern we observed is to combine visual elements with textual descriptions in every scene.

DP5: Consider traditional interaction design and usability patterns. For the app to achieve usability, employ *(1)* traditional interaction design and usability patterns like feedback, different input mechanisms, assistance, and transparency, and *(2)* expect users to make mistakes or change their minds and therefore employ an undo and redo feature because otherwise low usability and user experience with the app leads to frustration and discomfort.

Use established interaction design and usability patterns to guarantee the general usefulness of the app. Provide feedback about what happens, e.g., if a scene or image is saved. Let users know when a new AR element or node is added and provide different input mechanisms. Because the tablet gets heavy to hold after some time and typing longer texts on a tablet screen is complicated, one participant said that they wished to have speech-to-text functionalities to be more efficient. This goes in line with DP1, which deals with hardware limitations. Assist whenever possible, e.g., enable hyphenation, spell check and auto-resize in text fields. Make transparent how mechanisms work. Explain why different functions are disabled and what is needed to activate them. For example, participants were confused about why the AR scenario preview button was disabled initially. They did not know that the preview only works when nodes are in the node editor.

Consider that users will make mistakes or change their minds. To ensure users feel safe using the app, provide functions to undo and redo steps. During the study, participants accidentally deleted nodes in the node editor and could not undo the deletion. They said this frustrated them because they had to add the node with all AR elements in the scene again. Further, participants asked for mechanisms that ask for confirmation before making major changes like deleting a node. Participants suggested having a prompt message asking for confirmation if they wanted to delete the complete node.

DP6: Don't let users do unnecessary tasks that can be automated. To allow users to have a seamless experience with the app, employ mechanisms that automate unnecessary tasks because the user does not know they need to do the task, or it annoys them to do it repeatedly.

Our results show that participants struggled with doing repetitive tasks. Especially when the tasks could be automated, it annoyed them. For example, when they created a new node, they had to connect it with the previous node. Some participants did not know they had to connect the nodes, which influenced the workflow. When the nodes are not connected, the AR scenario preview, for example, only shows one node because the sequence of steps is not defined. Even if the participants understood that they had to connect the nodes, they were annoyed by these repetitive tasks. Automating these steps could support the flow and user experience. Another example is the spawning of nodes in the node editor screen. In the prototype for the study, a new node always spawned at the same position. Every time participants added a new node, they had to scroll to the starting position to move the node to the end of their process chain. This task annoyed them because they sometimes did not see that the node was already created, and the scrolling took time. It is one of the most mentioned problems in the study. Automating these tasks could prevent users from not knowing that they must do this and running into further problems. Besides, it ensures a more seamless and effective experience with the app.

DP7: Empower power users. To allow experienced users to use the app efficiently and effectively, employ mechanisms that empower them to quickly complete their goals because otherwise, the app seems inconvenient.

The participants who got familiar with the app tried to optimize their workflow. In the study, we observed that participants acted more structured and goal-driven during the second task. One participant asked for a feature that allows them to place several AR elements of the same type in a row. It was inconvenient that they had to click the toolbox button and select the AR element repeatedly. For example, by enabling a power-placing mode for experienced users, they can be supported to work more efficiently. This design principle aligns with DP6, which aims to automate unnecessary steps.

DP8: Enable the modeling of complex processes. To enable users to model their processes as realistically as possible, enable them to *(1)* construct more complex processes with branches and conditions and not only linear processes and *(2)* combine different node types because otherwise, not all processes can be modeled correctly.

Most processes amenable to being assisted with an AR instruction app are linear (Bräker & Semmann, 2021). Although the tasks for the study were also linear processes, participants said they missed the opportunity to create more complex ones. They asked for functionality to add branches, conditions, and logical gates. Adding this option empowers users to model all kinds of processes. It gives more freedom to create processes as realistically as possible.

DP9: Enable textual descriptions to enrich AR elements. To allow users to deliver comprehensive information, employ mechanisms to create additional textual descriptions for scenes, AR elements, and nodes because, in some cases, simple geometric AR elements are not expressive enough.

Textual information allows users to give specific instructions and contextual information. Geometric AR elements are mainly used to give instructions, but often, they are not expressive enough to be intuitive without text. In AR scenes, participants applied textual instructions frequently to instruct and describe the task that had to be accomplished, e.g., "press power button to turn on coffee machine". Participants liked to add textual information to AR elements to give details about them. Following the previous example, the user would add an AR element to the power button and attach the description "power button" to the AR element. A short scene description would be added in the node editor, e.g., "turn on coffee machine". While the first two provide textual information to the end-user of the AR instruction app, the latter one serves as an orientation for the user of the AR authoring app. During the study, the prototype only allowed adding textual information to AR elements and scene descriptions of certain node types. Participants criticized this and explicitly wished to have the option everywhere.

DP10: Provide a variety of AR elements and style options. To allow users to deliver information in a diverse way, provide a variety of AR elements to choose from and allow individual styling options in terms of color, size, and rotation because the distinction by geometric shapes alone is not expressive enough.

Geometric AR elements are essential for AR authoring apps. They, e.g., aim to highlight specific areas or objects of interest in the real world and guide the way with tethers. In the AR authoring app, they are only distinct by geometrical shape. Styling options like color, size, and rotation of the AR elements are not customizable. Certain AR elements were only available in one node type but not in another. Our results show that participants did not want to be restricted in their selection range of AR elements. They wanted to customize the styling of elements according to their purpose. For example, a larger element could mean that it is more relevant. Color coding could indicate that red AR elements visualize warnings. Additionally, the styling might be influenced by the surroundings. In the study, participants placed a red POI element onto a red surface. Because neither the size nor the color of the AR element was customizable, the AR element was not visible.

DP11: Allow adding, editing, and enriching the scene with photos and videos. To allow users to deliver information in the form of real-world snapshots, employ a mechanism for adding photos and videos and enriching/editing them because photos and videos are a simple option to capture a current state or target state of real-world objects.

Images or videos can depict details about real-world objects' current or target state. During the study, participants could use different ways to add images. They could make a screenshot via the node image button and use it as an attached preview photo. Further, they could attach photos from the tablet's media library to a scene or take a photo directly. Users are required to have the possibility to add annotations and drawings to these images. Further, the AR elements were always visible in the scene. Participants wanted to decide if the AR elements should be shown because they sometimes occluded important real-world objects they wanted to capture. Overall, the photos were always attached to one scene. Participants missed the option to attach photos only to a specific AR element to depict a snapshot of an object's current or target state in the real world. They could not add videos during the study, which would enrich the information.

DP12: Facilitate the spatial perception of AR elements. To support users in the precise positioning of AR elements, employ mechanisms that *(1)* give feedback on surface detection and *(2)* support distance estimation because the placement of AR elements requires a clear idea of the spatial position of the element.

All participants had issues with spatial perception in AR. To position AR elements precisely, the users must understand how the tablet camera perceives the world. A visualization of the surface detection mechanism helps the user to have an idea of the spatial perception of the tablet's camera. For the study, surfaces were overlaid with orange-colored polygons. In this way, participants received feedback about the tablet's surface detection accuracy. Additionally, users can be supported in distance estimation, e.g., using visual cues. One participant mentioned that it was beneficial that objects get larger when they are closer. Without any facilitation, it is challenging for users to estimate depths correctly.

DP13: Support precise positioning of AR elements. To allow users to create spatially accurate AR instructions, employ an interface that enables precise placement of AR elements because the position of AR elements mainly encodes instruction information and is an essential feature of the app.

One of the users' biggest struggles during the study was the precise positioning of AR elements. Participants had to adjust the position of AR elements because the positioning was not working as precisely as expected. Participants mentioned that this frustrated them, and it took them longer to reset the scene or reposition the elements. The positioning was especially challenging when they aimed to place an AR element further away or on transparent or mirrored surfaces. This relates to DP12 because surface detection is unstable for transparent/mirroring surfaces, and distance estimation is challenging.

The positioning interface in the study was a static cursor in the center of the tablet screen. The cursor itself cannot be moved. The user must move the tablet to position objects so the cursor points at the aimed place.

Then, the user must tap on the screen to place the AR element. To move the AR element, the user must select it again with the cursor and then press and hold the hand-moving button to reposition it. This interface was neither intuitive nor very precise to control. Participants also had issues selecting small AR elements because they were hard to target. Therefore, the need arises to provide an interface that enables precise positioning. Some participants mentioned that they would prefer to adjust the position by moving AR elements along the x-, y-, or z-axis in 3D space. This could be implemented by giving numeric values or having slides to move the AR elements per axis. Another participant mentioned listing all AR elements in the scene would be helpful. With this list, the selection and manipulation of AR elements might also be possible and more accessible than with the cursor.

DP14: Tracking accuracy is essential. For users to achieve a sense of reliability, employ accurate and reliable tracking mechanisms because with unstable tracking, users will become frustrated, and the usefulness of instructions cannot be guaranteed.

Tracking instability was a major issue during the study. All participants experienced problems in tracking accuracy and reliability. Almost everyone mentioned that they were frustrated that their positioned AR elements suddenly were not in the correct place anymore, making the app unusable. Especially when they finished creating the AR instruction app and started the AR scenario preview, the tracking became unstable. It took a long time to reposition the AR elements again to be satisfied with the outcome. This drastically influenced the usability in a negative way. Participants adapted their behavior if the tracking was unstable. Instead of placing AR arrow elements, they tended to choose circles that highlight a larger area, and therefore, tracking inaccuracy is not that restrictive. Therefore, tracking accuracy is essential for the app to function.

DP15: Implement object tracking. For users to achieve a feeling of blending real and virtual contents, employ mechanisms to track dynamic real-world objects because then AR elements can be attached to moving real-world objects.

The AR authoring app for the study implemented world tracking. This means that the surrounding is tracked, as well as static real objects that stay in the same place. However, dynamic objects that are not always in the same place in the environment are not trackable. In the study, for example, the coffee beans were not placed in a defined place. For this reason, moving the coffee beans would make the tracking invalid. Participants said that they also want to track dynamic objects and that this should be possible as an essential app feature.

Discussion

In alignment with previous research, our results show that AR authoring tools are not yet fully mature (Hönemann et al., 2022). Our proposed design principles contribute to the understanding and concrete design knowledge of no-code AR authoring tools. The results of our user study reveal that the usability of the AR authoring tool is below average (SUS score of M = 40.4). This is also reflected in the design principles DP5-DP7. Furthermore, tracking problems occurred frequently in the study (DP14), which frustrated the users and could also be an explanation for the low usability score.

The results of the UEQ reveal that the hedonic quality was positively evaluated. Hedonic quality measures non-task-oriented aspects like innovativeness and originality, which are addressed with the AR authoring app. Our results show that users would prefer the AR authoring app over a desktop application. Furthermore, users would prefer the no-code solution over programming and rate the creation of AR instructions with the AR authoring app as faster. This may seem contradictory to the users' statements that users took longer than expected to complete the tasks in the study. However, all study participants estimated the time required for programming to be significantly longer than the average duration during the study. Thus, this is an indication that no-code authoring has the potential to create AR instructions without programming effort and long training periods. This is consistent with the advantages mentioned in previous research (Elshan et al., 2023).

There was no significant difference in workload between the "coffee machine" scenario and the "seminar room" scenario. Thus, we did not find any indication that the app is better suited for one particular scenario. A lower workload and a shorter completion time were observed in the second condition. This suggests that a learning effect occurs over time, as reflected in DP4. The learning time could be shortened with tutorials and help since the users do not start from scratch.

As described in the meta principle *Basic Interactions*, we encountered several usability issues. For instance, participants criticized the lack of an undo function and transparency about the application's workflow and current state (DP5). More advanced users wished for shortcuts to work more efficiently (DP7), and in general, users were frustrated by the need to perform repetitive tasks, such as manually connecting nodes (DP6). We noticed a strong overlap with usability heuristics, a concept from interaction design research that describes rules of thumb to ensure good usability. The most popular heuristics are the ten heuristics from Jakob Nielsen (Molich & Nielsen, 1990; Nielsen, 2005). For instance, the heuristic "User control and freedom" proposes that users should always be able to leave an unwanted system state and undo previous actions, which participants of this study also mentioned. The participants' request to use more efficient ways to spawn multiple AR elements for more experienced users resembles Nielsen's heuristic "Flexibility and efficiency of use", which suggests implementing different interactions for users with different experience levels to ensure easy and efficient use. The possibility for users to miss the connection of nodes and, thus, accidentally not create a connected graph should be avoided according to the heuristic "Error prevention". Furthermore, Nielson also suggests adding "Help and documentation", which we also derived as a design principle (DP4). Overall, considering this strong overlap of the derived design principles and Nielsen's usability heuristic, we assume that these heuristics also apply to AR authoring tools and should be considered when designing further applications. For instance, these heuristics can be used to perform a heuristic evaluation, a usability analysis in which 3-5 usability experts analyze an application based on heuristics to search for usability problems systematically (Nielsen & Molich, 1990).

Tracking problems were the most common issue in the study. The prototype does not use a marker to initialize tracking but relies on ARKit's world tracking feature. Our results show that world tracking, in our case, was too unstable. These inconsistencies frustrated users tremendously and, in the worst case, made the AR instruction unusable. DP14 emphasizes the importance of implementing accurate tracking mechanisms.

The design principles DP9-DP11 focus on the multifaceted enabling of information encoding. This includes both the variety of elements and their design options. This aligns with Nebeling and Speicher (2018) since they also mention that users are missing encompassing and diverse design options in AR authoring tools. These findings suggest that our distinction between exploration and instruction nodes may be obsolete. It is contrary to the users' demand to have multiple options for information encoding. In a further iteration of the prototype, only two node types could be provided: A 2D node – analogous to the info node – and a 3D node that combines the features of the exploration and instruction node. Nevertheless, the business use and the consideration of role concepts could raise the need for more differentiated nodes.

A recent study found that most AR processes are linear (Bräker & Semmann, 2021). Our results nevertheless show that users aim to model more complex processes and need more flexibility in process modeling (DP8). During our study, we observed this need, especially for optional process steps. Thus, further mechanisms for this issue could be explored in the future.

Conclusion

Within our study, we investigated how to design a no-code AR authoring tool that allows users a lightweight and seamless way to augment processes without requiring any programming skills. We derived design principles that cover a broad range of relevant facets and should guide future design and development of AR authoring tools. We contribute to understanding no-code tools in the context of AR authoring. From a methodological point of view, our study is an example of a mixed-method approach in which quantitative surveys enrich and validate qualitative data. Furthermore, this approach is user-centric and aims to empower users to identify and explore the potential of AR in an accessible and convenient way.

Despite the relatively large size of the sample and the systematic analysis, the study has some limitations. First, a broader and more diverse sample would enhance the validity of the results. Specifically, the perspective of practitioners with little or no programming experience would be interesting. This issue can be overcome with future research that aims for real-world application of the AR authoring tool and, likewise, real-world processes. Second, several issues raised by the participants deal with distinct a priori design decisions in the authoring tool itself. Further refinement and maturation should help in this regard.

Additional research avenues build on the outcome of the participants' treatments. We plan to do another study that evaluates the appropriateness of the process guidance designed in this study. Doing so enables us to understand better if, to what extent, and how AR is beneficial in such scenarios.

Acknowledgments

This research was partly sponsored by the German Federal Ministry for Education and Research and in the project WizARd under the reference 02K18D181. Further information can be found at: http://www.wizard.tu-dortmund.de/.

References

Ashtari, N., Bunt, A., McGrenere, J., Nebeling, M., & Chilana, P. K. (2020). Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3313831.3376722

Atkins, B. (2020, November 24). *The Most Disruptive Trend Of 2021: No Code / Low Code*. Forbes. https://www.forbes.com/sites/betsyatkins/2020/11/24/the-most-disruptive-trend-of-2021-no-code--low-code/

Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators & Virtual Environments*, *6*(4), 355–385.

Bock, A. C., & Frank, U. (2021). Low-Code Platform. *Business & Information Systems Engineering*, *63*(6), 733–740. https://doi.org/10.1007/s12599-021-00726-8

Bräker, J., Hertel, J., & Semmann, M. (2023). *Appendix: Clustering and Derivation of Design Principles* [dataset]. https://doi.org/10.25592/uhhfdm.13222

Bräker, J., Osterbrink, A., Semmann, M., & Wiesche, M. (2023). User-Centered Requirements for Augmented Reality as a Cognitive Assistant for Safety-Critical Services. *Business & Information Systems Engineering*, *65*, 161–178. https://doi.org/10.1007/s12599-022-00779-3

Bräker, J., & Semmann, M. (2021). How Does Business Process Modeling Reflect Augmented Reality-Based Processes? *Pacific Asia Conference on Information Systems (PACIS)*.

Bräker, J., & Semmann, M. (2022). Dividing Complexity to Conquer New Dimensions–Towards a Framework for Designing Augmented Reality Solutions. *Americas Conference on Information Systems (AMCIS)*.

Bräker, J., & Semmann, M. (2023). The Process Augmentability Canvas—How to Find the Sweet Spot for Augmented Reality. *European Conference on Information Systems (ECIS)*.

Cranmer, Eleanor. E., Urquhart, C., Claudia tom Dieck, M., & Jung, T. (2021). Developing augmented reality business models for SMEs in tourism. *Information & Management*, *58*(8), 103551. https://doi.org/10.1016/j.im.2021.103551

Dengel, A., Iqbal, M. Z., Grafe, S., & Mangina, E. (2022). A Review on Augmented Reality Authoring Toolkits for Education. *Frontiers in Virtual Reality*, *3*. https://www.frontiersin.org/articles/10.3389/frvir.2022.798032

Elshan, E., Germann, D., Dickhaut, E., & Li, M. (2023). Faster, Cheaper, Better? Analyzing How Low Code Development Platforms Drive Bottom-Up Innovation. *Thirty-First European Conference on Information Systems*.

Gregor, S., Kruse, L. C., & Seidel, S. (2020). Research Perspectives: The Anatomy of a Design Principle. *Journal of the Association for Information Systems*, *21*(6). https://doi.org/10.17705/1jais.00649

Hart, S. G. (1986). *NASA Task Load Index (TLX)*.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology*, *52*, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Hertel, J., Gabel, J., Kruse, L., Wollborn, M., & Steinicke, F. (2022). Co-Design of an Augmented Reality Maintenance Tool for Gas Pressure Regulation Stations. *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 720–724. https://doi.org/10.1109/ISMAR-Adjunct57072.2022.00152

Hönemann, K., Osterbrink, A., & Wiesche, M. (2022, December 12). Enabling Non-Professional Users To Bring Physical Processes Into The Industrial Metaverse. *SIGHCI 2022 Proceedings*. https://aisel.aisnet.org/sighci2022/6

Kammler, F., Brinker, J., Vogel, J., Hmaid, T., & Thomas, O. (2019). How Do We Support Technical Tasks in the Age of Augmented Reality? Some Evidence from Prototyping in Mechanical Engineering. *ICIS 2019 Proceedings*.

Klinker, K., Wiesche, M., & Krcmar, H. (2019). Digital transformation in health care: Augmented reality for hands-free service innovation. *Information Systems Frontiers*, *22*(6), 1419–1431. https://doi.org/10.1007/s10796-019-09937-7

Kohn, V., & Harborth, D. (2018). Augmented Reality – A Game Changing Technology for Manufacturing Processes? *ECIS 2018 Proceedings*.

Konopka, B., Hönemann, K., Brandt, P., & Wiesche, M. (2022, September). WizARd: A No-Code Tool for Business Process Guidance through the Use of Augmented Reality. *Proceedings of the Demonstration & Resources Track, Best BPM Dissertation Award, and Doctoral Consortium at BPM 2022 Co-Located with the 20th International Conference on Business Process Management*. BPM 2022, Münster, Germany.

Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction*, *34*(7), 577–590. https://doi.org/10.1080/10447318.2018.1455307

Masood, T., & Egger, J. (2019). Augmented reality in support of Industry 4.0—Implementation challenges and success factors. *Robotics and Computer-Integrated Manufacturing*, *58*, 181–195. https://doi.org/10.1016/j.rcim.2019.02.003

Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, *77*(12), 1321–1329.

Mohammadhossein, N., Richter, A., & Lukosch, S. (2022). Benefits of Using Augmented Reality in Learning Settings: A Systematic Literature Review. *Forty-Third International Conference on Information Systems*.

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM*, *33*(3), 338–348. https://doi.org/10.1145/77481.77486

Möller, F., Guggenberger, T. M., & Otto, B. (2020). Towards a Method for Design Principle Development in Information Systems. In S. Hofmann, O. Müller, & M. Rossi (Eds.), *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry* (pp. 208–220). https://doi.org/10.1007/978-3-030-64823-7_20

Nebeling, M., & Speicher, M. (2018). The Trouble with Augmented Reality/Virtual Reality Authoring Tools. *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 333–337. https://doi.org/10.1109/ISMAR-Adjunct.2018.00098

Nielsen, J. (2005). *Ten usability heuristics*.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 249–256. https://doi.org/10.1145/97243.97281

Osterbrink, A., Bräker, J., Semmann, M., & Wiesche, M. (2021). Requirements for Augmented Reality Solutions for Safety-Critical Services – The Case of Water Depth Management in a Maritime Logistics Hub. *International Conference on Wirtschaftsinformatik (WI)*.

Purao, S., Kruse, L. C., & Maedche, A. (2020). The Origins of Design Principles: Where do… they all come from? In S. Hofmann, O. Müller, & M. Rossi (Eds.), *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry* (pp. 183–194). https://doi.org/10.1007/978-3-030-64823-7_17

Schrepp, M. (2015). *User experience questionnaire handbook. All you need to know to apply the UEQ successfully in your project*.

Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017). Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence*, *4*(6), 103–108. https://doi.org/10.9781/ijimai.2017.09.001

Strauss, A., & Corbin, J. (1994). Grounded theory methodology: An overview. In *Handbook of qualitative research* (pp. 273–285). Sage Publications, Inc.

Van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The Think Aloud Method: A Practical Approach to Modelling Cognitive Processes*. AcademicPress.

Venkatesh, V., Brown, S., & Sullivan, Y. (2016). Guidelines for Conducting Mixed-methods Research: An Extension and Illustration. *Journal of the Association for Information Systems*, *17*(7), 435–494. https://doi.org/10.17705/1jais.00433

Wehking, C., Kaiser, L. F., Schenk, B., & Brocke, J. V. (2021). Orchestration of employees' creativity: A phased approach. *Hawaii International Conference on System Sciences 2021 (HICSS-54)*.