ImpersonatAR: Using Embodied Authoring and Evaluation to Prototype Multi-Scenario Use cases for Augmented Reality Applications

Meng-Han Wu
School of Electrical & Computer Engineering,
Purdue University, IN, USA
email: menghanwutw@gmail.com

Ananya Ipsita [1]
School of Mechanical Engineering,
Purdue University, IN, USA
email: aipsita@purdue.edu

Gaoping Huang
School of Electrical & Computer Engineering,
Purdue University, IN, USA
email: gavinphuang@tencent.com

Karthik Ramani
School of Mechanical Engineering,
Purdue University, IN, USA
email: ramani@purdue.edu

Alex Quinn
School of Electrical & Computer Engineering,
Purdue University, IN, USA
email: aq@purdue.edu

Prototyping use cases for Augmented Reality (AR) applications can be beneficial to elicit the functional requirements of the features early-on, to drive the subsequent development in a goal-oriented manner. Doing so would require designers to identify the goal-oriented interactions and map the associations between those interactions in a spatio-temporal context. Pertaining to the multiple scenarios that may result from the mapping, and the embodied nature of the interaction components, recent AR prototyping methods lack the support to adequately capture and communicate the intent of designers and stakeholders during this process. We present ImpersonatAR, a mobile-device based prototyping tool that utilizes embodied demonstrations in the augmented environment to support prototyping and evaluation of multi-scenario AR use cases. The approach uses: 1) capturing events or steps in form of embodied demonstrations using avatars and 3D animations, 2) organizing events and steps to compose multi-scenario experience, and finally 3) allowing stakeholders to explore the scenarios through interactive role-play with the prototypes. We conducted a user study with 10 participants to prototype use cases using ImpersonatAR from two different AR application features. Results validated that ImpersonatAR promotes exploration and evaluation of diverse design possibilities of multi-scenario AR use cases through embodied representations of the different scenarios.

Keywords: prototyping, augmented reality, use case, role-play, scenarios

1 Introduction

In the early stage of application development, use case prototypes help designers to refine and validate the functional requirements of the application from the end-user's perspective [1–3]. Not only does this facilitate a goal-focused sequence of events and steps that is easy for stakeholders and end users to follow, it helps design teams identify potential problems before they happen during the actual development of the application features. Despite their many benefits, Augmented Reality (AR) designers face distinct challenges in prototyping use cases for their applications [4,5]. Since the many aspects of AR are integrated with the user's physical presence in the direct environment, conventional illustrations in form of 2D sketches, sequence diagrams, or interactive wireframes fail to adequately capture the designer's intent during prototyping of the use cases, and later communicating those with the stakeholders and end users for feedback. Therefore, this creates a necessity for introducing prototyping tools to assist teams designing AR during authoring and evaluation of the use case prototypes for intended application features.

Prior research has produced several interface prototyping tools that have successfully gained the designers' attention to explore AR interactions through high- or low-fidelity artifacts and/or express those to the stakeholders, prior to actual implementation [6,7]. However, these tools primarily focus on exploring *what interactions users should associate with*, and tend to overlook *when and how to leverage those associations*, the latter being considered important to engage users with the AR experiences efficiently [8]. To overcome this issue, we introduce the concept of *BranchingNode* and *BranchingTimeline* to enable mapping of causal associations between the different interaction components of the targeted use cases in a spatio-temporal context.

AR applications have shown to immerse users in task-relevant sensorimotor experiences through incorporating several embodiment elements (gesture, object manipulation, and whole-body movement) [9,10]. While presenting relevant content that is spatially- and temporally- integrated with real physical objects, these experiences facilitate situated cognition in users enabling them to explicitly map relationships between new interactions and previous experiences to actively construct new experiences and knowledge [11,12]. In the same context, prior research has shown that *avatars* can effectively demonstrate body-coordinated interactions between the users and the environment [13,14]. Therefore, we leverage humanoid avatars to enable designers embody their presence and actions in an AR environment to facilitate the prototyping of the different interactions in the use case prototypes. This embodiment leads designers to closely visualize the different interactions from the end users' perspective and validate the viability, feasibility and desirability of their design considerations. Moreover, such avatar-based demonstrations can be persisted for sharing and future reflection, similar to video-based prototyping.

Furthermore, the development of use cases typically involves an interactive evaluation using stakeholders, the benefits of which are two-fold. First, stakeholders (and designers) can gain more authentic experience by trying out typical tasks of the application to be designed. Second, by watching stakeholders perform those tasks, designers can identify usability issues in the application features and collect comments from the stakeholders [6] to adopt relevant changes during the subsequent design iteration. To that end, existing tools have limitations in evaluating the prototypes interactively by stakeholders and even by the AR designers themselves, when re-

[1] Corresponding Author.

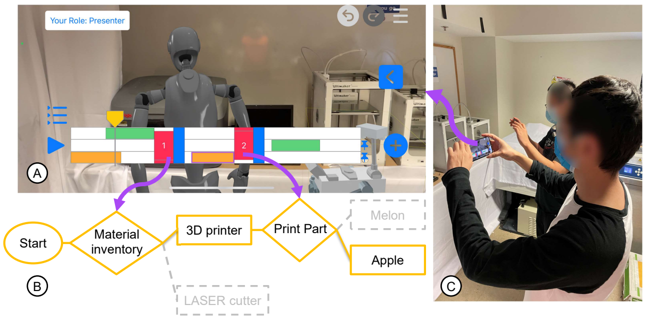


Fig. 1 ImpersonatAR allows designers of AR applications to author and share design prototypes of multi-scenario use cases with stakeholders. (A) A snapshot of ImpersonatAR in action, demonstrates an authored use case prototype of an AR application feature for Human robot collaboration in a factory setting. The timeline track at the bottom, referred to as the *BranchingTimeline* shows the various events and steps involved in the prototype. (B) The flow chart highlights the different events and steps associated with the multi-scenario use case. (C) Designers use the *BranchingTimeline* feature to control and choose the scenarios to demonstrate to the stakeholders.

lying heavily on *wizards* in Wizard of Oz (Woz) [15] or *facilitators* in paper prototyping [16,17]. Nonetheless, using Woz technique or paper prototyping introduces a heavy dependency on wizards or facilitators to simulate the interactive features, which hinders the ability to test and reflect about the design by stakeholders alone or by a single designer. Video prototyping, on the other hand, allows designers to persist, communicate, and reflect on the design [7,18]. However, video reduces the interactivity provided by Woz and loses the ability to test in AR environment [6].

Lastly, many other types of platforms have been explored for AR prototyping purposes beyond mobile devices. For example, desktop computer and Virtual Reality (VR) devices are used for high-fidelity content generation [19,20] and material organization [16,17,21]. Although those devices facilitate the prototype authoring, they hinder the ease of sharing with others, compared to mobile devices that are more accessible and movable.

Therefore, we present ***ImpersonatAR***, a mobile-based application for rapid authoring and interactive evaluation of use case prototypes for AR applications. The system functionalities are illustrated in Figure 1. First, designers can author various steps and events of the use cases using embodied avatar demonstrations and 3D animations of assets. After authoring the relevant events and steps, designers can arrange and situate those in a concise and sequential manner in a timeline track, referred to as the *BranchingTimeline* (Figure 1A). ImpersonatAR allows designers to navigate between multiple scenarios using *BranchingNode*, which represent the different conditions incurred in the use case (Figure 1B). Finally, ImpersonatAR allows designers and stakeholders to evaluate the authored prototypes using embodied demonstration and interactive role-play (Figure 1C). We conducted a two-session user study with 10 participants recruited as dyad groups, and with prior experience using AR/VR applications to test the ability of ImpersonatAR to facilitate embodied authoring and evaluation of the use case prototypes. During the first session, users were asked to individually author the prototypes using embodied interactions in an augmented environment. During the second session, each user dyad acted as designer and stakeholder to interactively evaluate the authored prototypes that were designed during the first session. Study results indicated that participants could successfully utilize the system features to prototype and evaluate multi-scenario use cases of different AR application features.

In summary, we highlight our key contributions as follows:

- An authoring workflow for prototyping of AR application use cases that contextualizes events and steps in form of the avatar recordings and 3D model animations using a mobile device and embodied demonstrations in the real world.
- The design of a *BranchingNode* and *BranchingTimeline* that allows AR designers to easily compose and manage associations of events and steps into scenario flows of AR application use cases to be designed.
- The interaction design and implementation that convert the scenario flows into semi-functional prototypes with which stakeholders can interactively explore in third-person and first-person view, requiring none or minimum help of Woz.

The article's structure is outlined as follows: In Section 2, relevant literature is presented. Section 3 introduces key terminology to facilitate comprehension of the work. Section 4 outlines a formative study conducted to uncover the design considerations adopted during the development of the system features. The system features are then detailed in Section 5, while proving examples from the first use case implementation, and Section 5.4 providing specific insights into the implementation details. Section 6 describes two additional use case prototype implementations. The user study procedure and its results are presented in Section 7. In Section 8, the study results are discussed in greater detail. Sections 9 and 10 respectively address the future potential for improvement and limitations. Finally, Section 11 presents the conclusion of the work.

2 Related Work

2.1 Early stage Design Prototyping and Evaluation of AR applications. ***AR Prototyping Tools:*** Despite the rising popularity of AR applications, authoring AR content remains challenging due to the steep learning curve. Mainstream AR authoring tools like Unity or Unreal necessitate expertise in 3D graphics and programming languages, excluding non-technical designers from the early design stage [21]. Although tools like SketchBox or PhotoShop do not require programming knowledge, they still demand a significant learning curve. Several digital tools have been developed to support AR authoring without programming skills [7,16,17,21,22]. Sharing a similar goal with these applications, ImpersonatAR is a mobile application that enables users to mockup scenarios within a physical space without needing prior programming knowledge.

In recent years, academic researchers have developed various tools that utilize traditional prototype methods for creating low-fidelity AR experiences during the early design stage. For instance, ProtoAR [17] allows rapid creation of 3D assets by capturing paper prototypes (i.e., sketches) and video snippets of the physical prototypes (e.g., clay models) to simulate a quasi-3D effect. ARcadia [22] enables quick prototyping of tangible user interfaces (UI) through a webcam, web browser, and paper. On the other hand, 360proto [16] transforms paper sketches into 360-degree representations of AR/VR environments, supporting live Woz for moving paper elements within the layers to simulate system responses. Although interaction mocked by paper prototypes are far away from an immersive AR/VR experience, they still provide flexibility for designers to express their ideas. Besides the 3D assets, ImpersonatAR also supports users to import their sketches in our system to support improvised creation. Another common prototyping tool is video prototyping [23], which refers to creating illustrations by augmented video through editing. Several AR prototyping tools [7,18,21,24] fall into this category. ImpersonatAR is not a video prototyping system, however, our UI is inspired by the video editing systems. Using ImpersonatAR, users could design the interactive behavior directly in the AR environment in real-time using virtual assets and avatars through a video editor-like interface.

Although many prototyping methods have been applied to support rapid AR prototyping [6,25], the support for creating interactive behavior in AR is limited in the context of goal-driven prototyping of AR use cases from end users' perspective [26]. More specifically, the techniques utilized by prior research are commonly found to be application-oriented and focus on the interactions that the target users are supposed to perform using the application. On the other hand, the system features of ImpersonatAR aim towards providing designers with a goal-oriented approach targeted towards the functional requirements of the system. Such an approach would enable designers to look closely from the end users' perspective as to how varied interactions and their associations would result in different scenario flows and user experience to achieve the desired goals in the environment.

Managing Multi-Scenarios and Alternatives: While existing AR rapid prototyping tools primarily support prototyping short-term interaction [7,16,17,27], the use cases pertaining to several AR application features often involve multi-scenarios [28–30], meaning that there are multiple possible outcomes of how users would interact with the system to achieve the desired goals. Traditionally, storyboards are frequently used to describe the complex usage scenarios [31] of the systems. Several storyboard-based design tools have been proposed to support the prototyping of complex system concepts. For example, DemoScript [32] defined cross-device interaction through storyboard-based illustrations. On the other hand, BrickRoad [33] and Toipary [34] use an interactive storyboard to simulate the interface for location-based systems. The storyboard-based design could also be found in commercial settings. For example, Xcode[2] uses storyboard to graphically lay out the workflow and transition between scenes.

Various methods have been used to manage the alternative pathways for storyboards. Specifically, version history and juxtaposition typically represent each alternative as a complete version of storyboard, which means that the same storyboard panels may be duplicated into different alternatives (e.g., Cal-ico [35]). On the other hand, a graph-based approach [36] saves alternatives as branches, which means that the same storyboard panel is a single node of the graph that is shared by related alternatives, instead of being duplicated. Such graph is typically a DAG, in which each path in the graph represents a storyboard. These methods inspire the design of ImpersonatAR. Given the constrained window size of a mobile device, we design a BranchingTimeline to show the multi-scenarios of the AR use case as a all-scenarios-in-one concise timeline.

Perhaps, the closest to our work is SketchStudio [37], which enables designers to prototype, share, and review an animated design scenario involving complex design subjects. However, the design and interaction of SketchStudio are different from our system. First, SketchStudio only supports one scenario while ImpersonatAR supports multiple scenarios. Second, SketchStudio is a sketch-based system and requires designers to create a node graph of the user's scenarios using a desktop environment, without information from a contextual scene. ImpersonatAR is a mobile application that enables the designer to author the scenarios within the contextual environment. Finally, SketchStudio users experience the sketched scenarios in a 2.5D virtual space without the ability to freely navigate between different scenarios. ImpersonatAR enables designers and stakeholders to directly experience the scenarios in a physical context with the animated digital assets.

Evaluating Prototypes: Paper prototypes and Woz are extensively used for testing AR prototypes. Paper prototyping requires in-person testing, while Woz serves as a rapid prototyping method for complex or technologically advanced systems, commonly used to simulate interactive behaviors in AR experiences [15,16,18,38]. The aim is to provide users with a sense of the prototype's functionality without actually building the entire system. However, Woz has limitations, such as wizard stress, fatigue, and delays between user actions and system responses [39]. Similarly, ImpersonatAR relies on demonstrations like Woz techniques but supports real-time simulation by sharing the AR content with stakeholders. Designers can smoothly control the player's progress bar on ImpersonatAR to showcase the designed interactions to stakeholders. Unlike 360proto [16], where the simulation uses paper, or PRONTO [7], where users passively observe the experience, ImpersonatAR allows stakeholders to directly experience the scenario and interact within the AR context through immersive role-play.

[2]https://developer.apple.com/xcode/

2.2 Embodiment in AR. ***Embodied nature of AR experiences:*** Leveraging the capabilities of AR media, the spatio-temporal content presented to users is contextualized through syncing with users' sense of presence and body ownership in real-time. This sync-up, facilitated by natural bodily interactions within the augmented environment, supports situated cognition and has proven highly effective in various applications, such as skill learning, training, hologram teleconferencing, manufacturing design, assembly, and entertainment and gaming industries [13,14]. In some cases, such experiences involve collaborative participation of multiple users performing tasks in the environment [40–42].

To ensure the development of user-friendly AR application features, designers need to rapidly explore design concepts and build a common understanding with stakeholders, especially in the early stages of development [43]. Moreover, such exploration, if relied directly upon the perceived experiences targeted for the end users, can enable designers to gauge the effectiveness of the application features in development of desired perception and skills in end users to achieve the target goals.

Prior research has demonstrated that in-situ authoring experiences in AR foster enhanced immersion, engagement, and motivation compared to 2D-based platforms like desktop environments and tablets [44]. Spatial affordances in the 3D authoring environment enable free and embodied interactions, allowing designers to directly experience the resulting application, a concept referred to as WYXIWYG: What You eXperience Is What You Get [44]. Such methods also permit testing content within the execution environment without switching between multiple platforms, enhancing the authoring experience. Building on these insights, ImpersonatAR utilizes designers' embodied interactions in AR to author and express targeted scenarios for achieving intended outcomes.

Authoring by Embodied Demonstration: Embodied demonstrations are extensively used for intuitive creation of digital content, leveraging users' body shape, positioning, and kinematics as spatial reference. They have found applications in complex art and 3D modeling [45], animation and story-making [46,47], realistic tutorials [13,48,49], human-robot interaction [50], and human-robot social interactions [51]. GhostAR enables visualizing, manipulating, and editing body gestures in AR for robot task authoring [29], while Porfirio et al. used human demonstration methods for human-robot social interactions [51]. Embodied demonstrations have also been utilized to author customized gestures and action detection algorithms. Lv et al. enabled end-users to design multi-touch gestures on tablets [52]. ACAPpella allowed interaction with multiple sensors [53]. Exemplar [54] and M.Gesture [55] supported rapid iteration and fine-tuning of gestures after demonstrations. MAGIC [56] enabled building classification algorithms through acting multiple gestures. Ye et al. employed such interactions to prototype proxemic and gestural experiences in IoT-enhanced spaces [27]. GesturAR created in-situ freehand AR applications using customized gesture inputs in spatial and temporal contexts [57]. CapturAR captured human actions in daily life to rapidly author context-aware applications [30].

To summarize, an embodied demonstration empowers rapid creation of complex and dynamic content through intuitive and straightforward bodily interactions. However, the exploration of such demonstrations in the context of prototyping use-cases for AR features still remains unexplored. ImpersonatAR provides a novel interaction technique to embody designers' presence and actions inside an augmented environment to impersonate the end user interactions with the system.

Avatar-based demonstration in AR: Virtual humanoid avatars are animated 3D models representing people's motions, gestures, and speech, widely used in AR and VR to understand, interpret, and visualize human actions. They have gained popularity as expressive tools for training and learning embodied skills. Chua et al. developed an asynchronous Tai Chi learning environment with a virtual instructor performing prerecorded movements [58]. YouMove utilized an AR mirror with a projected tutor avatar for full-body gesture comparison [59]. AvaTutor employed avatars to represent human tutor movements in machine task training scenarios [13]. These avatar-based tutors are used in two main ways: (1) Users directly observe demonstrations performed by avatars in third person to understand the topic. (2) The virtual instructor is superimposed in the AR view, allowing users to spatially align their bodies with the avatar, facilitating better comparison. Additionally, avatars can enable multi-user interactions involving multiple entities (users and/or embodied agents) collaborating on various tasks. Piumsomboon et al. used a miniature avatar for collaboration between a local AR user and a remote VR user [40,60]. Loki created a bi-directional mixed-reality telepresence system for teaching physical tasks using live and recorded remote instructions via avatars and RGBD point cloud [41].

These previous works reveal the virtual avatar's advantages in enhancing bodily-expressive human-human communication, for applications such as asynchronous learning, self-observing and training, teleconference, external tutor presence and MR remote collaboration. Nevertheless, the usability of the avatar as an agent for exploring and assessing end-user interaction design has not been systematically explored. This paper proposes to use avatars for representing the designer's spatial and bodily movements in the AR use case prototypes. ImpersonatAR utilizes this avatar based demonstrations to enable designers express the different intended interactions that they envision for their end users to perform the desired goals in the scenario use cases. Furthermore, these avatar based demonstrations facilitate the stakeholder understanding and interpretation of the proposed interactions to be able to critically assess them for refinement during further iterations.

3 Use case Definitions

Before proceeding further, we would like to define a few terms to aid in understanding the context of our work.

- ***Use case:*** A use case is a description of potential interactions between the system under consideration and its external agents, or actors in relation to a specific objective. Use cases create a goal-focused chain of events and steps by describing the functional needs of a system from the perspective of the end user [1].
- ***Actor:*** An actor could be a single person, a group of individuals, or even a computer program. The use case describes several groups of interactions that may take place between different external agents, or actors [2].
- ***Primary Actor:*** The primary actor is someone or something whose goals are achieved by the system in question. The objective of the primary actor is directly connected to the use case and it's fairly common for them to start the use case [3].
- ***Scenario:*** A scenario is an use case path containing possible sequence of interactions executed successfully to achieve the desired goal [1].
- ***Pre-condition:*** These are conditions or statements that need to be true for the use case to start [3].
- ***Post-condition:*** These are the system states after the use case is executed [3].
- ***Conditions:*** These are encountered during the execution of the use case that may change it's flow or the resulting scenario.
- ***Step***: A step is a discrete action or activity that a user takes in order to achieve a particular goal.
- ***Trigger:*** Triggers are events that may be used initiate the steps or other events of a scenario [2].

4 Design Considerations

The design of ImpersonatAR draws insights from the formative interviews with four expert AR application designers, as well as our synthesis of prior work and our direct experiences designing AR applications in a university lab where at least ten students are actively developing AR applications at any given time.

4.1 Formative interviews. We conducted interviews with four expert developers. One (P1) works at a large technology company with over 20,000 employees. Two (P2, P3) work at smaller local companies focused on technologies for IoT and manufacturing. One (P4) was a doctoral candidate in our lab who was not yet familiar with our project. We followed a semi-structured interview format with questions on the following themes:

(1) Experience developing AR applications.
(2) Discussions among team members and stakeholders about design of AR applications.
(3) Workflow for designing AR applications, from the initial spark to prototype.
(4) Tools used for developing embodied AR applications.
(5) Pain points and other unmet needs that arise when designing AR applications in a group.

Each interview lasted one hour. Participants were compensated with a $20 USD gift credit, and were ensured confidentiality of their applications. Each interview was recorded for later analysis.

Our interviews, together with our review of prior work, gave us insights on the needs of AR development teams designing with a variety of workflows and stakeholders. Synthesizing these insights led to three design considerations (DCs) to guide the development of ImpersonatAR. Below we summarize some of the insights we learned from the interviews. These are grouped into three themes, which roughly correspond to the three design considerations, which we will describe in the next section of this paper.

4.1.1 Difficulty expressing complex interactions of AR applications. Participants reported difficulty using existing tools to prototype AR experiences involving multi-scenario, body-level and/or multi-user interactions. P4 recounted the design of an AR storytelling application allowing children (multi-user) to tell stories (multi-scenario) by collaboratively controlling virtual objects. P1 described the design of an AR surgical application, in which virtual content was triggered by body motions (body-level interaction) performed by surgeons (multi-user). Recent publications illustrate more AR applications using body-level [29,30,61], multi-scenario [28], and multi-user interactions [40,41].

4.1.2 Rapid prototyping of AR applications. Although interviewees used a variety of tools to express design ideas within their teams, only P1 reported an interactive testing. P1's team created less robust applications using Unity to communicate proposed interactions. However, that incurred the cost and delay for application development, including UI structure, animations for 3D assets, and scripts for interactive behaviors. Non-programmers were unable to directly contribute, so prototyping new ideas was limited by the time availability of developers in the team [43].

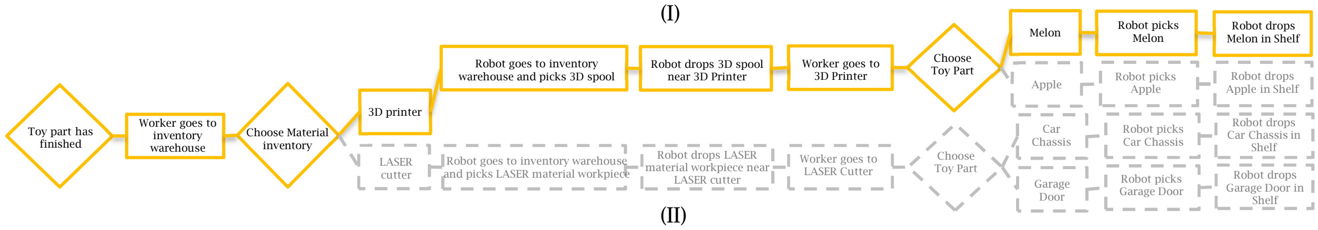
4.1.3 Collaboration in AR designing. Although all interviewees reported regular meetings with their stakeholders, the stakeholders could not directly experience proposed designs until they had been fully implemented. P1 came closest, with the Unity-based applications, but the cost and delay of development severely restricted the team's ability to share ideas on a regular basis. All interviewees reported that feedback was given verbally. Verbal feedback is spatially disassociated from elements in the interface [62]. These limitations suggested a need for design workflows that expose stakeholders and design team members to proposed designs directly in an AR experience, and facilities to associate feedback with interface elements [43].

4.2 Design considerations. The design of ImpersonatAR was guided by three key design considerations, which were arose from our interviews, in conjunction with our analysis of the prior work.

DC1: Expression of complex interactions (i.e., body-level, and/or multi-scenario, and/or multi-user interaction) of AR applications. The first DC is that prototypes created with the system

Use case 1:	AR application feature for Human Robot Collaboration
Description:	A toy manufacturing company wants to develop an AR application feature that enables workers to instruct robots for collecting and sorting finished toy parts.
Actors:	Factory worker, Robot
Primary Actor:	Factory worker
Pre-condition:	3D Printing of the toy part has finished.
Conditions:	Inventory warehouse material, Toy Part
Scenario 1:	After checking that 3D printing of "Melon" toy has finished, a factory worker instructs a robot to fetch 3D printing spool from warehouse. Then he instructs the robot to collect the toy part and store it in correct shelf.
Scenario 2:	After checking that 3D printing of "Apple" toy has finished, a factory worker instructs a robot to fetch 3D printing spool from warehouse. Then he instructs the robot to collect the toy part and store it in correct shelf.
Scenario 3:	After checking that LASER Cutting of "Toy Car Chassis" has finished, a factory worker instructs a robot to fetch LASER material workpiece from warehouse. Then he instructs the robot to collect the toy part and store it in correct shelf.
Scenario 4:	After checking that LASER Cutting of "Garage door" toy has finished, a factory worker instructs a robot to fetch LASER material workpiece from warehouse. Then, he instructs the robot to collect the toy part and store it in correct shelf.

(I)



(II)

Fig. 2 Use case 1: AR application feature for Human robot collaboration in a factory setting. (I) The flow chart representation shows the multiple scenarios of the use case. (II) Use case Description

must be able to express a wide range of embodied interaction types, including body-level interactions carried out by one user or multiple users within one scenario or multiple scenarios (non-linear interaction flow).

DC2: Rapid interactive testing of proposed designs. The second DC is that the system must allow rapid and joint testing of proposed designs by both designers and stakeholders. This is to allow more cycles of feedback and discussion. Also, rapid prototyping tends to be easier for novices to learn, allowing for a more inclusive design collaboration.

DC3: Sharing proposed designs and feedback elicitation in the AR environment. As was noticed from the formative interviews, with prior design methods, feedback was disassociated from the physical environment, making it difficult to discuss spatial aspects of the design problem. Thus, the third DC was that the system must allow design teams to share proposed designs and elicit feedback directly within the AR environment.

5 ImpersonatAR Design and Development

Aligned with the above design considerations, we designed and implemented ImpersonatAR, a mobile-based authoring, sharing, and evaluation system that enables AR designers to easily design and visualize their multi-scenario use cases, and then communicate those to stakeholders. ImpersonatAR consists of three modes: 1) Authoring Mode in which designers can create prototypes for the target use cases rapidly; 2) Demonstration Mode in which designers and stakeholders can watch the prototypes synchronously or asynchronously from a third-person view; 3) Role-play Mode in which designers or stakeholders can role-play an avatar and interact with the prototype lively from a first-person view. These modes are facilitated by incorporating the following functionalities into ImpersonatAR: (1) Authoring of steps using avatar representations and 3D animations of assets, (2) Authoring associations between steps to capture different scenario flows in the use case, and finally (3) Interactive testing with stakeholders using embodied demonstrations and role-play with the authored prototypes.

The workflow of capturing events and steps using body-coordinated interactions is straightforward: AR designers first demonstrate the local and/or spatial bodily interactions in front of a mobile device's camera, and then ImpersonatAR automatically converts those body-level interactions into avatar interactions via vision-based body tracking. Repeating this process, multi-user interactions can be represented as multiple avatars to facilitate prototyping use cases involving multi-user collaboration. This workflow is much simpler than similar systems that blend multiple 3D characters in the space (e.g., [37]). Besides, ImpersonatAR allows designers to add virtual assets and animations during the authoring mode to enhance the scope of interaction possibilities with the avatars. The start time and duration of the animations for 3D assets are scheduled by using avatars as a spatiotemporal point of reference [29].

To represent multiple scenarios of an AR use case, ImpersonatAR introduces a construct called *BranchingNode*. Every interaction that may lead to different pathways for the interaction flow can be represented as a BranchingNode. An example of the BranchingNode can be the action of operating a menu with three options that leads to three candidate branches. After choosing an option from the menu, the designer can navigate into the corresponding branch to further specify the subsequent interactions for each branch, comprising of new events, steps and even new BranchingNodes. More specifically, multiple scenarios utilized for the prototyping of an AR use case in ImpersonatAR can be represented as a directed acyclic graph (DAG). Given the limited screen size of a mobile device, we introduce a *BranchingTimeline* to situate the multiple events, steps and scenarios of the use case in a concise and sequential manner. Note that designers can still use ImpersonatAR to prototype single scenario use cases without using BranchingNode. However, being able to contextualize and navigate between multiple scenarios helps designers to prototype diverse and/or complex AR application features resulting from selection of different BranchingNodes. In contrast to the single scenario interaction prototyping, the above mechanism provides designers with a chance to explore and/or optimize the end-to-end interaction flow of authored prototypes for multi-scenario use cases.

To facilitate the evaluation of the authored prototypes, ImpersonatAR enables two modes: *demonstration* and *role-play*. In the demonstration mode, designers and stakeholders can present and/or observe the different avatar demonstrations of the prototypes in third-person view through their own mobile devices. In the role-play mode, however, the stakeholders can role-play the avatars and interact with the prototype directly in first-person view. For example, a role-player can freely decide upon the branch of a BranchingNode to proceed with by either interacting with the menu on the screen or enacting a gesture. This interactivity is achieved using the following: 1) During the authoring phase, AR designers are provided with tools to easily create interactive menus or gestures; 2) In the role-play mode, a second mobile device other than the role-player's is used to track his/her interaction which subsequently gets compared with the registered interactions and switches the branch once matched. This second device, also referred to as the *Monitor* can be either held by an observer, or be simply placed on a tripod if needed. Since the interactions during the role-playing can be performed easily by the stakeholders and do not need any manual inputs from wizards, the above mechanism thus minimizes the need of Woz.

We will explain the different modes and functionalities of our system using an example use case showing ***Human robot collab-***

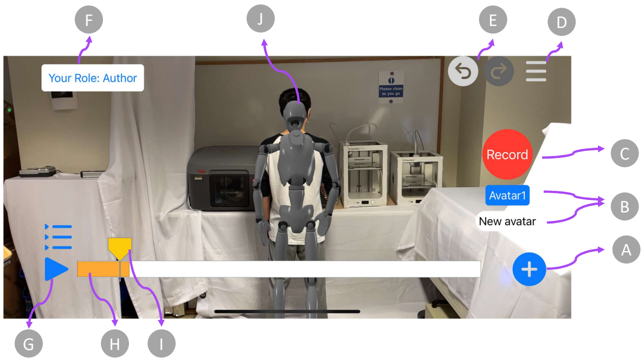


Fig. 3 ImpersonatAR's *Motion Recording* screen: (A) The *Add* button can show a menu to enter different screens, e.g., *Motion Recording*, *Adding Asset*, etc.; (B) the *Active Avatar* text displaying the current avatar's name and a *Add Avatar* button to add new avatar; (C) the *Record* button to start/stop recording motions; (D) the *Menu* button to switch between different modes (Authoring/Demonstration/Role-play) and save/load/share project; (E) *Undo* and *Redo*; (F) the *User Info* text displaying the current role of the user holding the device; (G) the *Play* button; (H) a *Timeline* track with a recorded *Avatar Segment*; (I) the *Cursor*; and (J) the avatar.

oration (HRC) in a factory setting. This is the use case for an AR tutorial feature to facilitate HRC in a toy manufacturing factory as shown in Figure 2. A factory worker can use this feature to instruct a robot to collect and sort finished toy parts in their respective shelves. The use case begins with the worker checking that the part manufacturing is completed. The worker then goes to the inventory warehouse and commands the robot to pick the material (3D spool if the manufacturing process involved 3D printing, or LASER material work-piece if the process involved LASER cutting) for replacement. The robot then collects the material from the inventory warehouse and drops it near the appropriate machine. The factory worker returns to the machine location and instructs the robot to collect the finished part by selecting the correct option - Apple or Melon for 3D Printing, or Car Chassis or Garage Door for LASER cutter. The robot then collects the selected toy part and moves to the appropriate shelf to place it.

The above use case comprises four distinct scenarios, each with unique events and outcomes. It can be observed that two conditions are utilized in the use case, i.e., "Inventory warehouse material" and "Toy Part". To illustrate the prototyping of the use case using ImpersonatAR, let's consider the example of instructing a robot to collect a 3D printed Apple toy part. Using ImpersonatAR, a designer can author the multi-scenario prototype by following these steps: (1) a recorded avatar demonstrates the worker's motion to the material inventory, (2) a BranchingNode with options related to "Inventory warehouse material" condition is incorporated, (3) an animation is created that demonstrates the robot following the instruction to fetch 3D printing spool, (4) another recorded avatar illustrates the worker's movement towards the 3D printer, (5) a BranchingNode with options related to "Toy Part" condition is incorporated, and (6) an animation is created that demonstrates the robot following the instruction to pick and place the printed Apple toy part. Using snapshots from the prototype that was developed for the above use case, we will next explain the different functionalities of our system.

5.1 Authoring of steps using avatar representations and 3D animations of assets. ImpersonatAR utilizes avatar demonstrations and 3D animation of assets to represent the various steps of the use case. The corresponding representations using avatars and animations are referred to as *Avatar Segment* and *Animation Segment* respectively.

5.1.1 Record avatars. *Avatar* is the key to demonstrate body-level user interactions. Recording avatars in ImpersonatAR is as easy as recording a video through a mobile device. First, AR designers enter the *Motion Recording* screen (Figure 3) by clicking the *Add* button (Figure 3A) and choosing *Motion Recording* from the menu. Then, designers click the *Record* button and start to demonstrate the spatial and bodily interactions in front of the device's camera. Once the recording is stopped, ImpersonatAR automatically converts the body motions into an avatar demonstration via vision-based body tracking. Meanwhile, an *Avatar Segment* in orange color is added into a *Timeline* track. Designers can drag the *Cursor* to view the avatar at any given point in time, or press the *Play* button to view the avatar demonstration like a 360° video. By default, the avatar is recorded for one user and denoted on the device screen as an *Active Avatar* text, e.g., "Avatar1" (Figure 3B). By pressing the *New Avatar* button (Figure 3B) and repeating the recording process, multiple avatars can be added in the same environment to illustrate multi-user interactions.

5.1.2 Insert and manipulate assets. To meet the diverse need of designers, ImpersonatAR provides four ways for preparing virtual assets: 1) import 3D assets via ImpersonatAR's *Gallery* screen which has been filled with 3D assets collected from online sources such as *Thinger - 3D Printing Models*[3] and *Google Poly*[4] , 2) scan a physical object with a third-party 3D scanning application (e.g., *3d Scanner App*[5]) and import the virtual counterpart into ImpersonatAR, 3) draw a sketch with pen and paper and take a photo with ImpersonatAR, and 4) draw a 3D stroke by sketching on the screen while moving the device (similar to *Just a Line* app[6]). These four methods vary from high fidelity to low fidelity in order and the typical time taken for preparing assets using each method is less than a minute.

Next, designers can insert an asset into the AR environment by tapping on a plane surface (e.g., floor or tabletop). Designers can scale, move, and rotate the asset using a standard set of multi-touch gestures on the device screen. However, the move operation cannot translate the asset out of its anchored plane. To support more flexible positioning (e.g., in mid-air), ImpersonatAR provides a *Hold and Move* feature. When holding the *Hold and Move* button, the selected asset moves relative to the device, and stays in the new position once the button is released.

5.1.3 Add animation. Designers can attach animation to the assets in three different ways. Firstly, they can draw a trajectory using an operation identical to the aforementioned 3D stroke drawing. The trajectory of the device then becomes the path for the animation. Secondly, designers can add position checkpoints in the environment to create a multi-stop linear movement for the desired animation. Specifically, the starting point of the animation is the current position of the asset. When designers tap the *Add Point* button after moving the device to the next desired position, the new position of the device becomes the next stop in the animation. Likewise, designers can add more checkpoints at different desired positions. Once all position checkpoints are captured, the asset navigates along the captured path by moving from one point to next in a linear fashion.

Thirdly, designers can create a special animation using *Move With* feature. During spatial and bodily interactions in AR applications, it is often the case that different assets need to move with respect to each other, and most frequently, they need to move with the human avatar. An example from the use case is as follows. To prototype the step where the robot is carrying the 3D printing spool from the inventory and following the factory worker, it seems necessary to make the 3D printing spool move with the robot while

[3]https://apps.apple.com/us/app/thinger-3d-printing-models/id1255097591
[4]https://poly.google.com/
[5]https://www.3dscannerapp.com/
[6]https://justaline.withgoogle.com/

the robot moves with the avatar. To that end, ImpersonatAR allows designers to set an asset to move with another asset or move with a part of the avatar (i.e., left hand, right hand, head, and hip).

When an animation is saved, an *Animation Segment* in green color is added into a different *Timeline* track (Figure 4B). Multiple animations of the same asset appear as multiple segments in the same track.

5.1.4 Edit avatars and animation. After adding avatar recordings and animations, designers can continue editing them to make further modifications to their appearance, such as adjusting their timing. Designers can tap on the screen to select an asset in the AR environment or select a segment in the timeline. The selected asset is highlighted with a bounding box, yellow in color (Figure 4C) while the selected segment is highlighted with a thick border, purple in color. Meanwhile, a *Context Menu* button (Figure 4E) is shown on click of which a context menu pops up containing the relevant editing commands specific to the selected segment or asset (Figure 4F). Using the *Context Menu* provides a consistent user experience for locating editing commands while also maintaining flexibility with the editing operations.

Several commands are common in the *Context Menu* for the *Avatar Segment* and the *Animation Segment*, such as *Play*, *Split*, *Move*, *Copy/Paste*, and *Delete*. As the names suggest, these commands provide the designers to perform the most common editing operations. For example, the *Move* command allows designers to drag the segment horizontally to adjust its start time. In the case of editing an animation, the avatar can serve as the spatio-temporal reference. The *Split* command can split a segment into two, considering the cursor position to be the common end point for the new segments. This command can be helpful for trimming undesired portions in the segments.

Additionally, a few specific operations are provided in the *Context Menu* for the *Avatar Segment* and the *Animation Segment*. For example, the *Context Menu* for only the *Animation Segment* contains the *Resize* option for changing the duration of the animation. This command is not needed for the *Avatar Segment* as its duration is fixed based on the actual demonstration. The *Context Menu* for the selected asset has the aforementioned *Add Animation* and *Hold and Move* buttons (Figure 4F). The *Context Menu* for the *Avatar Segment* has a *Trigger* operation to trigger events, which is discussed later in more detail.

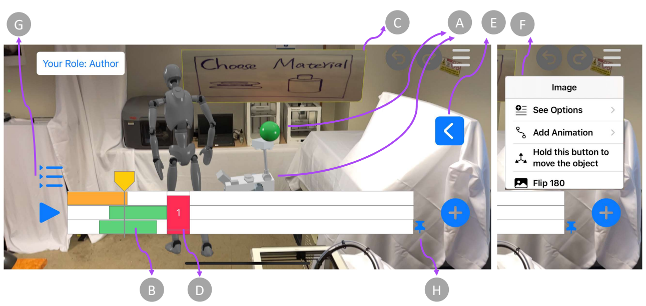


Fig. 4 ImpersonatAR's interface after adding two animations and a BranchingNode. (A) A virtual robot carrying a 3D printing spool; (B) two *Animation Segments* in green representing the animations for the robot and the 3D printing spool; (C) the 2D sketch showing a menu with two options for materials to choose from; (D) the BranchingNode corresponding to the menu with two options which shows that Option 1 is active; (E) the *Context Menu* button for the selected 2D sketch; (F) the *Context Menu*; (G) the *Track List* button to show a list of all tracks and also pin/unpin a track to manipulate its visibility in the timeline; (H) a *Pin* sign next to the pinned track in the timeline.

5.1.5 Manage tracks. Since multiple assets and avatars may be added into the scene, this can create multiple timeline tracks. Due to the limited screen size, ImpersonatAR allows only a few tracks to be visible on the timeline (e.g., 3 tracks for phone). The *Track List* button (Figure 4G) can be pressed to show a list of all tracks (currently visible and invisible). ImpersonatAR implements a Least Recent Used (LRU) mechanism to determine which track is visible on the timeline. When users select an asset in the scene or from the track list, its corresponding track appears on the timeline and evicts the least recent selected and unpinned track. Since an avatar track is frequently needed as a reference for an embodied interaction, ImpersonatAR allows designers to pin/unpin a track through the *Track List* for easy and frequent access of desired tracks.

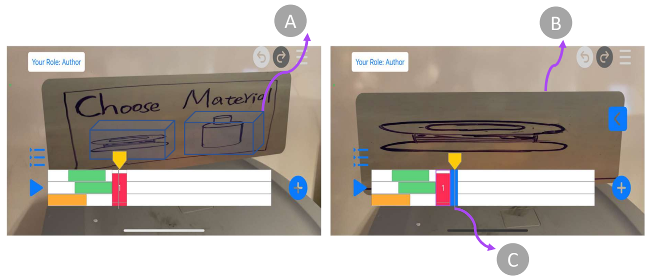


Fig. 5 Adding touch-based triggers and feedback. (A) Overlaying virtual colliders to menu options on a 2D sketch to make them touchable from screen or by hand; (B) a different sketch acting as the feedback if the left option (Option 1) is chosen by user; (C) a blue segment indicating a feedback is added for Option 1.

5.2 Authoring associations between steps and events to capture different scenario flows in the use case. Different steps in the use case may have certain dependencies and associations with one another and other events. A few examples from the HRC use case would be: (1) The robot goes to the 3D Printer only after the worker has chosen the corresponding machine. (2) The UI to choose the toy part shows up after the worker reaches the corresponding machine. Designers can author simple temporal dependencies by simply adjusting the start times of relevant segments on the timeline track. For complex dependencies requiring inputs from the designer, e.g., active choices from UI menus, location, gestures, etc., ImpersonatAR allows designers to author those using *BranchingNodes*, *Triggers* and *Feedback* which will be discussed next in more detail.

5.2.1 Add BranchingNode. *BranchingNode* is the key to represent multiple scenarios in the use case. The conditions that may lead to different course of events and steps in the environment can be represented as a BranchingNode. For example, Figure 4C shows a menu to choose a material from two given options: (1) 3D-printing spool (left) and (2) LASER work-piece material (right). Choosing different materials leads to different machines, steps and events. If the 3D printing spool is chosen, then the robot will carry it to a 3D printer; else if the LASER work-piece material is chosen, the robot will carry it to a LASER cutter. The BranchingNode is visualized as a vertical scroll wheel on the timeline with multiple options (Figure 4D). Users can rotate the scroll wheel to select or switch to a desired option. The traditional timeline enhanced by the BranchingNode is thus referred to as the *BranchingTimeline*.

Designers can convert an asset into a BranchingNode by adding options through its *Context Menu*. Typically, a designer can easily sketch a 2D menu or UI with multiple options using pen and paper that can serve as a BranchingNode. In addition, a 3D asset can also serve as a BranchingNode if the designer can trigger different events through different interactions, e.g., using different gestures to control a robot to perform different actions [29].

5.2.2 Add triggers with feedback. The prototype authored so far is fully animated, yet not interactive. In other words, the

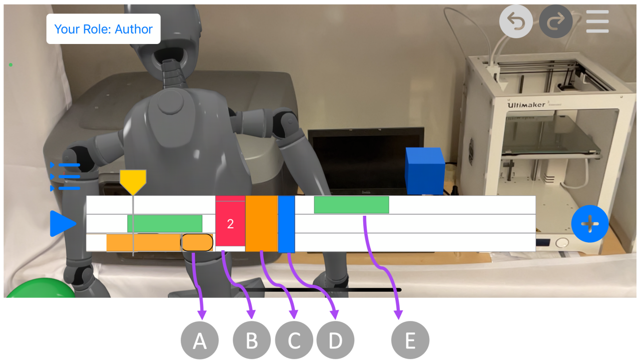


Fig. 6 Adding motion-based trigger for the appearance of asset or individual option of BranchingNode. Note that the BranchingNode is switched to Option 2, followed by a new animation on the timeline. (A) An *Avatar Segment* that is used as a motion trigger to make the BranchingNode pop up on the timeline; (B) a BranchingNode; (C) an *Avatar Segment* that is used to select the corresponding option from the BranchingNode; (D) a feedback shown on the timeline when the Option 2 is triggered through motion

prototype does not react to any inputs from the user. ImpersonatAR enables interactivity by supporting three types of *Triggers*: *location-based*, *touch-based*, and *motion-based*. This interactivity is required as in many cases, steps or events are triggered actively based on users' inputs or interactions.

Firstly, designers can add location-based triggers. In this kind of trigger mechanism, an asset can appear/disappear or an animation can start/pause when the potential user is in close proximity of the location anchored with a trigger. For example, the menu display in the inventory warehouse as shown in Figure 4C can appear when a user gets closer to the location of the inventory warehouse.

Secondly, designers can add touch-based triggers. As mentioned earlier, designers can select an asset by tapping on the device screen. To enable selection of individual options from a sketched menu or UI, ImpersonatAR allows designers to add virtual interactive colliders to different options (Figure 5A). Using this functionality, designers or stakeholders can interact with the BranchingNode through device screen or directly with hands, given that their hand positions are known through body tracking.

Thirdly, designers can add motion-based triggers. Since interactions in AR are inherently spatial and embodied [28–30,61], in ImpersonatAR, each *Avatar Segment* can be re-purposed as a motion trigger. Using this trigger mechanism, once a designer or stakeholder mimics the recorded avatar's motion, the corresponding event can be triggered. For example, Figure 6A is a part of the *Avatar Segment* that is used as a motion trigger to make the BranchingNode (Figure 6B) appear on the timeline. Likewise, Figure 6C is another *Avatar Segment* that is used to select the corresponding option (Figure 6B) from the BranchingNode. In such cases, different options of a BranchingNode need to register different motion-based triggers to avoid confusion.

After adding triggers for different options of the BranchingNode, ImpersonatAR provides visual *Feedback* to inform designers about the option which is triggered. For example, Figure 5B is a sketch that indicates the feedback in AR view when the Option 1 is triggered through touch. Meanwhile, Figure 5C shows a blue segment for the feedback right after the BranchingNode says Option 1 in the timeline. Similarly, Figure 6D indicates a feedback when the Option 2 is triggered through motion.

Note that the BranchingNode's option, and the corresponding motion-trigger and feedback (Figure 6B-D) are visualized vertically in the timeline. Adding BranchingNodes essentially partitions the entire timeline into *Sections*. For distinction, we denote a section containing horizontal avatar and animation segments as a *Normal Section*, while a section consisting of vertical BranchingNode's option, trigger and feedback segments is denoted as a *Branching Section*. All sections on the timeline following a Branching Section belong to the latter's selected option. Once a user rotates the vertical scroll wheel on the BranchingNode and switches to a different option, all sections belonging to the new option becomes visible on the timeline. In such cases, the entire timeline is always showing an end-to-end scenario for the use case. Users can easily navigate between different scenarios using this BranchingTimeline.

5.2.3 Manage the quick growth of scenarios. A potential concern that may arise during authoring is—if every BranchingNode appearing in the timeline needs to add triggers and feedback for its options, the authoring process may get tedious and the number of scenarios may grow too quickly. This concern is mitigated in two ways: (i) ImpersonatAR enables users to copy and paste segments or sections. This feature can be useful when the same menu or UI is used multiple times in one or many scenarios. (ii) We observe that not all BranchingNodes within a scenario may have similar diversity. In most scenarios, once the first few BranchingNodes are defined, the subsequent BranchingNodes only need to specify the relevant options. This is because each scenario depicts one typical task of the application to be designed. Once users have made decisions in the first few BranchingNodes, their goals tend to become more clear and focused, eventually leading them to choose the most-likely options and discard the irrelevant ones.

5.3 Interactive evaluation and role-play with designers and stakeholders.

5.3.1 Demonstration Mode. To communicate their design ideas and receive feedback, designers and stakeholders can enter the *Demonstration Mode* and watch the authored prototype collaboratively. Stakeholders can use their own mobile devices which are more accessible and familiar to use than other AR platforms such as HMD-based AR headset. Additionally, their devices can automatically connect with the designers' devices when they are under the same WiFi network. Thereafter, designers can choose to share the prototype with their peers.

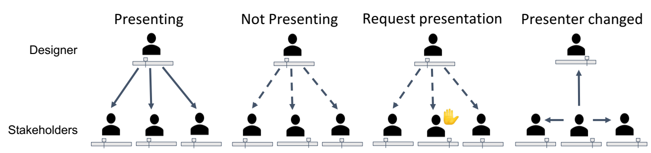
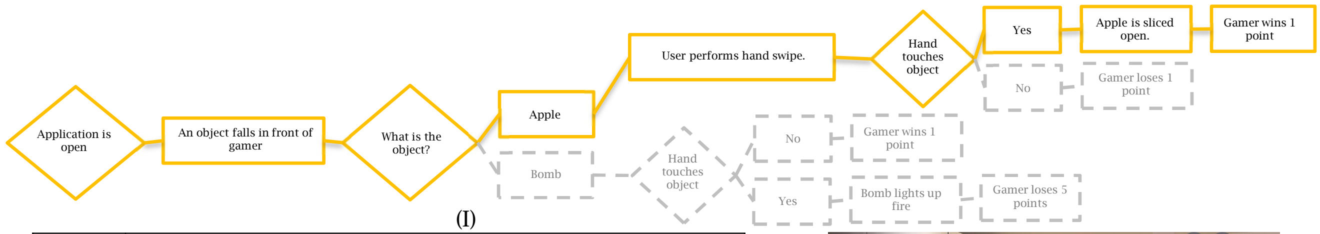


Fig. 7 In Demonstration Mode, everyone's play time is in sync with the presenter.

The workflow of the demonstration mode is shown in Figure 7. At the beginning, one designer becomes the *Presenter* while others are *Observers*. When the presenter starts presenting the prototype, the *cursor* on the timeline on observers' devices will be automatically synced with that of the presenter. This forced synchronization guarantees that the idea is demonstrated in the same branch and at the same time, without any inconsistencies. Next, when the presenter stops presenting, each observer can freely explore different branches at their own pace. An observer can also request to present if something needs clarification or attention. Once the current presenter approves the request, the observer becomes the new presenter and forces others to look at the same time of interest.

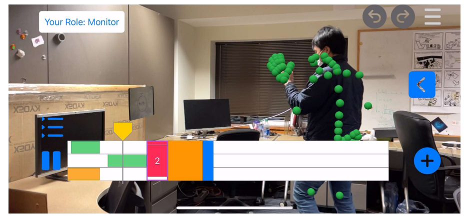
For the demonstration mode, we would like to clarify the use of ImpersonatAR rather than asking the designers to physically demonstrate their design ideas to the stakeholders when they are co-located. There are several reasons: 1) the usage of ImpersonatAR is assumed to reduce the presenter's load who may otherwise need to tediously demonstrate the desired events and steps and their associations in each presentation session; 2) the avatar-based demonstrations utilized in ImpersonatAR can be persisted for sharing and future reflection on the design ideas and prototypes; 3) it is challenging, if not impossible, for a single designer to demonstrate



(I)

Use case 2:	AR Fruit Ninja Game feature
Description:	A gaming company wants to develop an AR Fruit Ninja game that allows users to cut apples in mid-air using body swipe gesture while missing bombs.
Actors:	Gamer
Primary Actor:	Gamer
Pre-condition:	Gamer has opened the application.
Conditions:	The object is hit
Scenario 1:	An apple falls in front of the gamer to which she performs a hand swipe gesture. If the hand hits the apple, the apple slices open. Gamer receives a point.
Scenario 2:	An apple falls in front of the gamer to which she performs a hand swipe gesture. If the hand misses the apple, nothing happens. Gamer loses a point.
Scenario 3:	A bomb falls in front of the gamer. If the gamer misses the bomb, gamer receives a point.
Scenario 4:	A bomb falls in front of the gamer. If the hand hits the bomb, the bomb lights up fire. Gamer loses 5 points.

(II)



(III)

Fig. 8 Use case 2: AR Fruit Ninja Game Feature. (I) The flow chart representation shows the multiple scenarios. (II) Use case Description. (III) A use case prototype which shows a stakeholder role-playing with the scenario prototype of an AR Fruit Ninja game.

multi-user collaboration in real-time, which is however convenient and easily achievable with multiple avatars.

5.3.2 Role-play Mode. In the role-play mode, the stakeholders or designers can role-play an avatar and interact with the prototype in first-person view. The avatar is hidden by default and replaced with a dashed line, gray in color, from the user to the desired destination. First, the role-player can freely choose which option of the BranchingNode to proceed with. This is followed by the role-player performing the required action in the AR environment. The added triggers then enable the prototype to react to the role-player's action. The touch-based trigger with device screen or the location-based trigger can be performed with the role-player's device alone. On the other hand, the touch-based trigger with bare hand or the motion-based trigger need a second device to track the role player's body motion. In such cases, a second device (i.e., *Monitor*) could be either held by a person or placed on a tripod. This minimizes the need of Woz when it comes to common body-level interactions, but will fall back to Woz if the motion-based triggers require to perform complicated interactions.

In both demonstration and role-play modes, designers and stakeholders can walk freely in the environment to find a good view angle of ImpersonatAR in action. Notably, stakeholders may even switch their roles as authors and persist their comments and thoughts through embodied demonstrations. The demonstration and interactive exploration in the multiple scenarios can help both designers and stakeholders understand the big picture of the application feature and facilitate design optimization of the corresponding scenario workflow.

5.4 Implementation. ImpersonatAR is implemented on an iPhone 11 with iOS 14. However, the application can be run on any iOS devices with Core A9 processor or higher. We use Apple's ARKit[7] to track the environment and body motions (i.e., ARBodyTrackingConfiguration), and use RealityKit[8] to render the virtual assets, animations, and avatars. There is no stringent need of LIDAR. But we found better tracking possibilities with LIDAR-equipped devices such as iPhone 12 Pro and iPad Pro. The 2D user interface of ImpersonatAR is implemented using SwiftUI[9].

In demonstration and role-play mode, we use a zero-configuration network[10] to achieve multi-peer connection. Specifically, one device acts as a host while nearby peer devices can join the host automatically. Once connected, the host encodes the project data and sends it across to the peers. Any subsequent communication such as for time synchronization, request for presenting, etc. is also enabled by this connection procedure.

In role-play mode, in order to detect touch-based trigger with bare hand, the hand position is tracked using the ARKit's body tracking relative to the virtual colliders utilized in a BranchingNode. In order to detect motion-based triggers, we rely on positions of the user's head and both hands (nine degree-of-freedom inputs). We adopted a FastDTW implementation[11] to compare the real-time states of the user to all motion-based triggers belonging to the relevant section. The run time for FastDTW is $O(kN)$ where k is the number of triggers and N is the average duration of a trigger.

6 Example Demonstrations

With ImpersonatAR, users can rapidly prototype use cases for AR application features that involve one or multiple scenarios utilizing body-level and/or multi-user interactions. Here we demonstrate use cases from two other AR applications inspired from designs that we encountered during the formative interviews and review of prior work.

Use case prototype of an AR Fruit Ninja game feature. Leveraging the realistic visualization of human actions in AR, ImpersonatAR can prototype the body-level interactions for the AR application feature that involves human motions. For example, Fruit Ninja is an AR game where users perform body-level actions to cut virtual fruits. Using ImpersonatAR, a designer who wants to prototype a scenario of the user cutting a virtual apple can create the animation by adding three segments on the BranchingTimeline: (1) a virtual apple falling from top to bottom, (2) an avatar recording to visualize the "cutting action", and finally (3) the virtual content displaying whether the fruit is cut or missed. The description of the use case is shown in Figure 8-I, II, along with a snapshot from ImpersonatAR showing a stakeholder role-playing with the prototype in Figure 8-III.

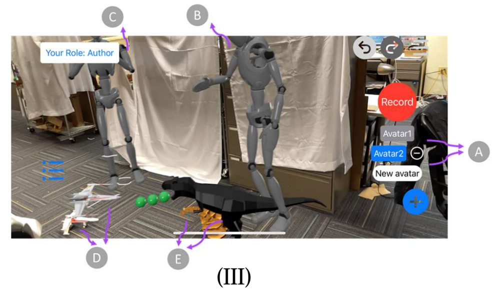
Use case prototype of a collaborative AR storytelling application feature. Figure 9-I, II describes the use case of a collaborative storytelling application feature where two kids A and B can collaboratively enact a scene by controlling two virtual characters in the story, a T-rex and a X-wing fighter. The kids would require to use their body motions to control the actions of the virtual characters. For example, kid A can trigger an attack from the T-rex by doing a boxing action. Once the attack is triggered, T-rex spits fire to attack its opponent, the X-wing fighter. As can be seen, this use case

[7] https://developer.apple.com/documentation/arkit
[8] https://developer.apple.com/documentation/realitykit
[9] https://developer.apple.com/documentation/swiftui/
[10] https://developer.apple.com/documentation/multipeerconnectivity
[11] https://github.com/melode11/FastDTW-x

(I)

Use case 3:	Collaborative AR Storytelling Application feature
Description:	A storyteller wants an AR application feature that allows kids to enact a fighting scene between two fantasy characters, a T-rex and a X-wing fighter.
Actors:	2 kids A and B. A controls T-rex and B controls X-wing fighter
Primary Actor:	Kid A
Pre-condition:	The attack mode of T-rex is ON.
Conditions:	The object is hit
Scenario 1:	Kid A enacts a boxing gesture to make T-rex attack by spitting fire balls. Kid B performs a cross-hand gesture to make X-wing fighter dodge the attack. X-wing fighter dodges the attack successfully.
Scenario 2:	Kid A enacts a boxing gesture to make T-rex attack by spitting fire balls. Kid B performs a cross-hand gesture to make X-wing fighter dodge the attack. X-wing fighter gets hit by the fire ball and loses its cannon.

(II)



(III)

Fig. 9 Use case 3: Collaborative AR storytelling application feature. (I) The flow chart representation shows the multiple scenarios of the use case. (II) Use case Description. (III) A use case prototype of a collaborative AR storytelling application feature. (A) Active Avatar texts showing the current avatars in the environment; (B and C) two avatars representing two users in the game; (D) a X-wing fighter and its virtual counterpart (top); (E) a cardboard T-Rex and its virtual counterpart.

can also involve two possible scenarios. For example, there can be two different outcomes following the attack from T-rex, resulting in the X-wing fighter to either successfully dodge or get hit. Once the X-wing fighter is able to dodge the attack, it could choose to fight back. Kid B can enable the X-wing fighter to dodge the attack from its opponent by performing a cross-hand gesture. However, if the X-wing fighter gets hit, one of its LASER cannons will be damaged. The complexity of the scenarios depends on how designers would like to design their application. ImpersonatAR supports the use case prototyping of this application feature by allowing designers to visualize multiple users as multiple avatars, and define triggers and animations to express the interactive behaviors, and steps in the story. Figure 9-III shows a snapshot of the use case prototype developed using ImpersonatAR.

7 User Study

We recruited 10 participants (9 male, 1 female) with ages between 18 and 30. Users were recruited as dyads so as to participate as designer and stakeholder during the study. All users had prior experiences in using AR/VR applications. The selection criteria for users focused on their familiarity with AR/VR technology, aiming to (1) ensure background knowledge necessary to design prototypes, within the limited study duration, and (2) minimize the need for extensive introduction to ImpersonatAR features. While technical expertise was not a requirement for user selection, it is worth mentioning that 8 out of 10 users had at least 3 months of experience in developing AR/VR applications. None of the participants had experience with ImpersonatAR before the user study. The study lasted for around two hours and each participant was compensated with a $40 USD Amazon gift credit.

Each study was conducted with two users and in two consecutive sessions. In the first session, each user independently used ImpersonatAR to author two use case prototypes related to different topics. In the second session, the prototypes were evaluated by the users in an alternating fashion. In each prototype evaluation, the designer of the prototype presented it to the other user, who played the role of a stakeholder throughout the process. During the study, we provided each user with an iPhone 12 to run the ImpersonatAR. After the study finished, we conducted a conversation-type interview with each participant to get subjective feedback. Participants were asked to complete a survey questionnaire based on 1-5 Likert-Scale (*1=Strongly Disagree, 5=Strongly Agree*) to rate their experience in using the features of ImpersonatAR.

7.1 Session 1: Authoring prototypes for multi-scenario use cases. In this session, each participant was given a topic for an AR application feature and asked to use ImpersonatAR to author the prototype for the use case. Note that for each dyad group, the users were not provided any information about each other's topics. Two topics from the three use cases described before were used for the study: 1) the AR tutorial application feature; 2) the Fruit Ninja game feature. Due to the constrained study duration, the prototype designs were restricted to contain two BranchingNodes, each containing 2-3 options, thus resulting in 4-6 different scenarios for a given topic. During this session, we evaluated how the authoring mode of ImpersonatAR enables users to prototype multi-scenario use case prototypes.

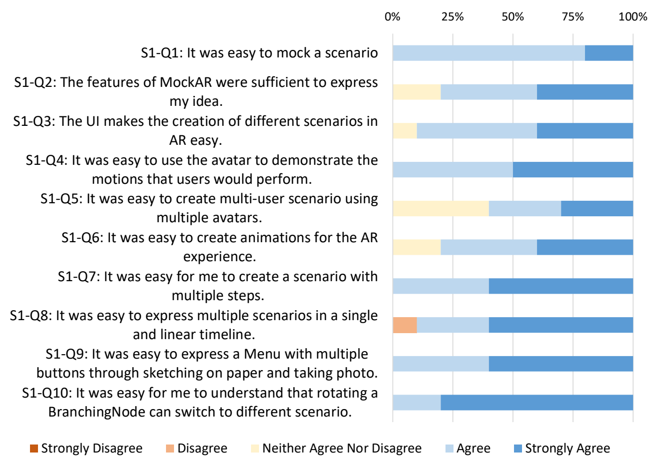


Fig. 10 Average participant scores from the survey questionnaire following Session 1 for authoring.

7.1.1 Results.. All 10 participants successfully completed the authoring tasks. The self-reported scores from the survey questionnaire following this session are shown in Figure 10. Overall, participants found ImpersonatAR easy to use to mock up a scenario (S1-Q1: AVG=4.20, SD=0.42). The system features were sufficient to express the design idea for the AR application (S1-Q2: AVG=4.20, SD=0.79). The UI features were easy to operate for creation of different scenarios (S1-Q3: AVG=4.3, SD=0.67). Specifically, participants found it easy to record avatar demonstrations for different body-level interactions that potential users would perform (S1-Q4: AVG=4.50, SD=0.53). They also found the avatar helpful in creating multiple scenarios for their prototypes (S1-Q5: AVG=3.9, SD=0.88). It was easy to create the animations they needed for their scenarios (S1-Q6: AVG=4.20, SD=0.79). It was easy to create a scenario with multiple steps (S1-Q7: AVG=4.60, SD=0.52) as well as express multiple scenarios in a single and linear timeline (S1-Q8: AVG=4.40, SD=0.97). Participants could easily sketch their menu ideas on paper and take a picture to upload into ImpersonatAR (S1-Q9: AVG=4.60, SD=0.52). Finally, the participants found it easy to use the BranchingTimeline to switch between different scenarios (S1-Q10: AVG=4.80, SD=0.42).

7.2 Session 2: Evaluating prototypes. In this session, we tested the evaluation features of ImpersonatAR in the Demonstration and Role-play modes. Each participant took turns in order to explain the prototype scenarios to the other participant using ImpersonatAR. When testing the demonstration mode, participant who acted as designer during Session 1 shared the project file with the other participant (i.e., stakeholder). Once both devices successfully finished the localization and mapping in the AR environment, the demonstration mode began. The process of testing during the role-play mode is similar to the demonstration mode, the only difference being that users interacted lively during the whole scenario in the role-play mode instead of passively watching the scenario as in the demonstration mode. During this session, we tested the usability of the system features in the demonstration mode and then explored how the role-play mode enhances the understanding of the scenarios.

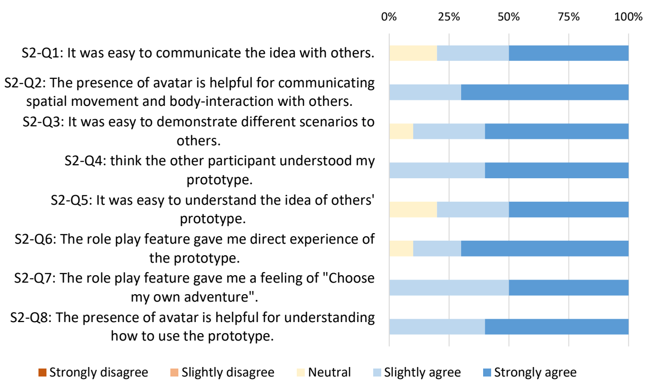


Fig. 11 Average participant scores from the survey questionnaire following Session 2 for demonstration + role-play.

7.2.1 Results.. All 10 participants successfully demonstrated and role-played the scenarios. The results from the survey questionnaire following the second session are summarized in Figure 11. During the demonstration mode, participants found it easy to demonstrate (S2-Q3: AVG=4.50, SD=0.71) their scenarios and communicate the ideas using ImpersonatAR (S2-Q1: AVG=4.30, SD=0.82). In addition, the presence of the avatar helped participants to easily explain the ideas behind the spatial and body-level interactions (S2-Q2: AVG=4.70, SD=0.48). Overall, they were able to make the stakeholders understand their use case prototypes for the AR application features (S2-Q4: AVG=4.6, SD=0.52).

During the role-playing of the scenarios, all participants agreed that the presence of avatar was helpful in understanding how to use the prototype (S2-Q8: AVG=4.60, SD=0.52). Besides, the role-play mode provided participants with direct experience with the prototype (S2-Q6: AVG=4.70, SD=0.70). Participants also felt like owning their adventure during the study (S2-Q8: AVG=4.5, SD=0.53). Overall, participants were able to understand the designers' concepts during the role-playing mode (S2-Q5: AVG=4.30, SD=0.82).

8 Discussion

The example demonstrations along with the study results affirms about the capabilities of ImpersonatAR to express a wide range of embodied interaction types, including body-level interactions carried out by users within multiple scenarios (non-linear interaction flow). In general, the study results indicate that participants found ImpersonatAR easy to use and its features sufficient to express the given topic. Here we will discuss the insights collected from the user study in more detail while providing examples from the prototype implementations to understand how the features of ImpersonatAR can help designers to effectively author and evaluate their use case prototypes.

BranchingNode and BranchingTimeline: During the authoring mode, participants provided positive ratings about the concept of BranchingNode to create multiple scenarios. After selecting the particular choice in the BranchingNode, designers can easily define scenarios by adding subsequent steps to the branch corresponding to the selected option. They can easily repeat the process for other options to define multiple scenarios for the use case. *"I can easily create multi-scenarios using BranchingNode." (P5).* Furthermore, ImpersonatAR allows flexible authoring of options and the corresponding triggers in the BranchingNodes. This can ultimately enable creation of customized scenarios targeted towards customized goals in the environment. For example, in the storytelling application feature, designers can add customized gestures to trigger interactions for different animations of different virtual characters, thereby enriching the storytelling capabilities by making room for numerous story elements. The flexibility in authoring choices of the BranchingNode can thereby enhance the diversity of the scenario space while facilitating the scope for creativity in the design space exploration.

Finally, the participants found the BranchingTimeline makes the management between different scenarios easy. Users can seamlessly switch between different scenarios by selecting different options from the BranchingNode, e.g., the context menu UI helps to navigate between the subsequent operations for the corresponding machine and the corresponding robot tasks. *"The BranchingTimeline was not intuitive to me at first, but it makes the switching between scenarios once I understand the concept." (P8)* Furthermore, by enabling reuse of components within the same or different scenarios of the use case, the BranchingTimeline can be considered as an effective way to enable faster authoring of use cases. Due to the limitations on the number of BranchingNodes during the study, the effectiveness of the feature in reduction of authoring time could not be further explored and is considered as a scope of future work.

During the demonstration mode, participants found it easy to demonstrate and explain the scenarios using ImpersonatAR. *"I can precisely refer to the step I would like to explain by moving the cursor to the correct timestamp, and switching between different scenarios" (P3).* From the conversational interview, participants also mentioned that they barely needed to manually control the role-play process as the system already switched to different branches based on the role-player's actions. This further indicates that ImpersonatAR feature minimizes the load of the wizard in Woz during the evaluation of the use case prototypes.

Embodied authoring and evaluation: The embodied nature of various features of ImpersonatAR allows designers to perform natural bodily motions for authoring the use case prototypes. During the process, designers can employ embodied skills from everyday life including naive physics, body awareness and environmental awareness to express their ideas related to the AR designs [8]. From the study results, embodied authoring using avatar recordings was considered as an easy and intuitive way to demonstrate the events and steps in the use cases by using simple body-level interactions. As evident from prior work, such embodied interactions can enhance the quality of authoring and user experience by allowing more freedom of control, and more emotional engagement during the process [63]. As pointed out by P2, *"Using the avatar to express the body motion is interesting and straightforward, it helps me explain my idea."* Furthermore, by employing the embodied features to express the dynamic relationship between the physical and the virtual representations, the designers can increase the prototype's ability for self-interpretation in an intuitive way. An example can be provided from the Fruit Ninja use case, where the swipe gesture can be easily used as an intuitive way to cut the falling fruit. Such facilitation of the physical-virtual interplay and interpretation in the contextual environment, can ultimately enable stakeholders to infer and enact the demonstration in real-time during evaluation.

Research in cognitive neuroscience has highlighted the importance of embodied demonstration as a learning tool in shaping cognition [64]. Simply put, the body and mind are closely interconnected, and cognition is significantly influenced by how bodies (self or others) move through space. Following in line with the ideas of embodied cognition, the avatar-based demonstrations can help advance stakeholders' understanding of the cognitive processes underlying the use case prototypes. This is noticed from the positive user ratings where the presence of avatar was found helpful in understanding how to use the prototype. As pointed out by P7, *"Avatar gives me a virtual clue of where to go and what I should do in role-play mode, which is really helpful."*

The role-play mode enabled stakeholders to enact the events and steps to get a first-hand experience of the targeted interactions. As pointed out by a user, *"It is interesting to interact with the virtual contents, the interaction helps me understand what another participant wants to explain."* According to research on the embodied perspective of interaction and the important role played by the physical body in how we experience, our understanding and interpretation of the world around us are rooted in our, often unconscious, experience of our bodies. While presenting relevant content that is spatially- and temporally- integrated with real physical objects, these experiences facilitate understanding through (1) lowering cognitive load by allowing interaction through natural motions, and (2) fostering generative processing when users explicitly map relationships between new ideas and analogous sensorimotor experiences [64]. Through the facilitation of situated and embodied cognition using direct experiences with the prototypes would not only enable the stakeholders to understand the designer's intent, but also execute their own critical judgement on the proposed interactions. This critical analysis of the proposed interactions can further result in providing insightful feedback to the designers to improve upon their design features in the subsequent iterations.

9 Future Scope of Improvement

During the conversational interview, users were asked to comment about any difficulties they faced during interacting with ImpersonatAR or suggestions for improving the system features. From the feedback that we received, we will briefly explain the challenges and recommendations along with our potential solutions for improvement.

Navigation between many scenarios. While all participants found the design of BranchingNode makes creation and management of multi-scenarios efficient, some participants raised a foreseeable issue of the system. 4 out of 10 users mentioned one way or another that, given the limited screen size, it would be hard to find the BranchingNodes on other branches if there are too many scenarios. This feedback emerged when the users were designing new scenarios but tried to find a BranchingNode of some other scenarios. Indeed, difficulties to navigate between different BranchingNodes would be an inevitable issue when the number of scenarios becomes too large. A quick fix could be naming each BranchingNode during authoring mode which can enable users to search them by name. Another potential solution involves adding more contextual filters, such as user's location, motions, and interacted assets, similar to [30]. For example, users can find all BranchingNodes that are close to their current location or those that can be triggered with a given motion. Notably, those contextual filters can be combined together for a more accurate search.

More flexible scenarios structure. Participants raised two suggestions: 1) support merging two scenarios at some point and sharing a common ending, and 2) support looping back to prior steps to make the scenarios more dynamic. Note that our current implementation assumes all the scenarios form a divergent tree. Conceptually, the first suggestion of merging scenarios can still be achieved with a DAG while the second suggestion requires the representation to be a directed *cyclic* graph. Although these suggestions sound reasonable, such complex structure is rarely needed. This is because the effects of the events are often *accumulated*. If the prior events of two scenarios are different, they would set the application in different states. In case two scenarios happen to reach the same BranchingNode (e.g., a shared menu) and choose the same option, it is less likely that they result in the exact same outcome (e.g., same animation). In other words, the two scenarios are unlikely to share the same ending. Here, we provide some insights to support the second suggestion (i.e., looping back) for ImpersonatAR. One way is to build the connection between two BranchingNodes by enabling designers to tag on BranchingNode. When a cursor hovers over a BranchingNode, the system would reveal links to other BranchingNodes with the same tag.

Free vs. guided role-play. During the user study, some participants brought up the idea of supporting free exploration of the scenarios. While this sounds exciting at the moment, we figure that it might be impossible to support without considering all the possible scenarios and programming it in advance. Although a role-player can freely choose which branch to proceed when confronted with a BranchingNode, the next step after selecting the choice is still predefined. In other words, stakeholders cannot move to an arbitrary step that is not defined in the chosen branch. However, the free exploration might be applicable for some applications that are spatially dependent and time independent. For example, museum tourism can be explored in any given order, the later scenarios do not depend on users' action history.

Co-located sharing. ImpersonatAR supports designers to share the AR content in the same physical space. Yet, some participants were curious about whether it is possible to remotely share the scenarios. ImpersonatAR leverages the contextual information from the physical environment (e.g., the physical machines in Figure 3), such that users could easily tell the story or create the scenarios by adding minimal virtual assets to the scene. Currently, ImpersonatAR does not support remote sharing. For future development, there are two possible ways. First, sharing the scenarios through VR. Designers could scan the physical environment using LIDAR-equipped devices (i.e., iPad Pro) and then send the whole project to remote stakeholders. The stakeholders could load the project in place and experience the scenarios with a VR headset. Second, export as video prototyping. Designers could share the live view of the same environment with remote stakeholders and demonstrate the scenarios. However, doing it in video will lose the ability of role-playing in a first-person view.

Managing visual hints. One common problem raised by the participants is that sometimes the physical environment is distracting and makes it hard for them to find the added virtual contents. A quick solution to the issue could be simply using a device with larger screen such as iPad. On the other hand, we could address the issue by adding more visual hints (e.g., arrows) to let users know where they should look at.

Managing alternatives. Although we do not explicitly support the creation of alternatives of a scenario, it could be accomplished by adding extra options to BranchingNode and using Copy/Paste features as introduced in Section 5.1.4. Given the scope of current study, we do not further explore the alternative management in AR rapid prototyping.

10 Limitations

System features and accessibility: While all participants successfully authored and demonstrated the multi-scenario use case prototypes, our result is constrained by the participant pool. First, since most participants have some AR development experiences, the learning curve of the system might be smoother than novice users. For example, people who have programming experience might find it easier to understand the BranchingNode. Second, the user selection criteria did not take into account the AR designing experience of users, primarily because identifying such individuals within a university setting would be challenging. Nevertheless, we acknowledge the value of including AR designers in testing the authoring features of ImpersonatAR as it would undoubtedly yield valuable insights and feedback. Third, the evaluation of prototypes during the study was performed by stakeholders having familiarity with AR/VR applications. Moreover, for each user dyad, users may already gain experience with ImpersonatAR during the authoring mode when asked to create a prototype, and/or during the first prototype evaluation when they experience other's prototype (either as an observer or role-player). These previous experiences might introduce potential biases when evaluating the system features due to the acquired knowledge and context. It is important to acknowledge that real-world scenarios could involve stakeholders with varying levels of technical expertise, including those with no prior experience in using our system. Therefore, it is important to note that the current evaluation may have limitations in terms of capturing the perspectives of users with different levels of prior experience. Another limitation is that the design goals of the system are derived from a small group of AR experts (e.g., manufacturing, IoT). Since the possibility of using AR is still at its exploration stage, our system might not be able to support the prototyping of all types of AR experience.

Despite the above limitations with the participant pool, we believe the formative and the user study results still provide valuable insights into eliciting the system requirements and analyzing the effect of the system features into prototyping AR experiences. Nevertheless, we acknowledge the benefits of future studies with larger sample size and more focused user selection criteria to further validate and extend the current findings, and hope that our work provides design insights to explore this area in greater detail.

Body Tracking Capabilities: One limitation of ImpersonatAR is related to the body tracking feature of the ARKit. ImpersonatAR already supports the prototyping of body-level interactions [47]. However, the ARKit has difficulty tracking delicate body motion such as hand gestures and eye gaze, which constrains the flexibility to define hand- or eye-based triggers to create interactive behavior. Although users can define these triggers by trimming the avatar clips, the triggers would not be recognized by the system. To tackle this problem in the future, we plan to utilize state-of-the-art hand gesture recognition approaches, such as probabilistic methods and deep neural networks. This will grant more freedom to the designer and enable more types of interactive behavior.

11 Conclusion

We presented ImpersonatAR, a mobile-based application for authoring and interactive evaluation of AR use case prototypes. We interviewed four AR expert designers and formulated a set of design considerations to address the current limitations of AR prototyping tools. Based on these considerations, we have demonstrated an end-to-end workflow and interaction technique to author, manage, and evaluate multi-scenario use case prototypes. Finally, we evaluated the system with 10 users and received positive feedback. Our work expands the area of HCI research in prototyping tools for AR use cases by introducing BranchingNode and avatars through embodied demonstration.

Acknowledgment

This work was partially supported by the National Science Foundation under grant FW-HTF 1839971. We also acknowledge the Feddersen ChairFunds. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

References

[1] Cockburn, A., 1997, "Structuring use cases with goals," Journal of object-oriented programming, **10**(5), pp. 56–62.

[2] Kulak, D. and Guiney, E., 2012, *Use cases: requirements in context*, Addison-Wesley.

[3] Schneider, G. and Winters, J. P., 2001, *Applying use cases: a practical guide*, Pearson Education.

[4] Duin, A. H., Armfield, D. M., and Pedersen, I., 2019, "Human-centered content design in augmented reality," *Content strategy in technical communication*, Routledge, pp. 89–116.

[5] Evans, K. and Koepfler, J., 2017, "The UX of AR: Toward a human-centered definition of augmented reality," User Experience Magazine, **17**(5).

[6] Freitas, G., Pinho, M. S., Silveira, M. S., and Maurer, F., 2020, "A Systematic Review of Rapid Prototyping Tools for Augmented Reality," *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pp. 199–209, doi: 10.1109/SVR51698.2020.00041.

[7] Leiva, G., Nguyen, C., Kazi, R. H., and Asente, P., 2020, "Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction," *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, doi: 10.1145/3313831.3376160.

[8] Xu, Y., Barba, E., Radu, I., Gandy, M., Shemaka, R., Schrank, B., MacIntyre, B., and Tseng, T., 2011, "Pre-patterns for designing embodied interactions in handheld augmented reality games," *2011 IEEE International Symposium on Mixed and Augmented Reality-Arts, Media, and Humanities*, IEEE, pp. 19–28, doi: 10.1109/ISMAR-AMH.2011.6093652.

[9] Webel, S., Bockholt, U., Engelke, T., Gavish, N., Olbrich, M., and Preusche, C., 2013, "An augmented reality training platform for assembly and maintenance skills," Robotics and autonomous systems, **61**(4), pp. 398–403.

[10] Johnson-Glenberg, M. C., Birchfield, D. A., Tolentino, L., and Koziupa, T., 2014, "Collaborative embodied learning in mixed reality motion-capture environments: Two science studies." Journal of educational psychology, **106**(1), p. 86.

[11] Shaghaghian, Z., Burte, H., Song, D., and Yan, W., 2022, "Design and Evaluation of an Augmented Reality App for Learning Spatial Transformations and their Mathematical Representations," *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, IEEE, pp. 608–609, doi: 10.1109/VRW55335.2022.00155.

[12] Shaghaghian, Z., Burte, H., Song, D., and Yan, W., 2022, "An Augmented Reality Application and User Study for Understanding and Learning Spatial Transformation Matrices," arXiv preprint arXiv:2212.00110.

[13] Cao, Y., Qian, X., Wang, T., Lee, R., Huo, K., and Ramani, K., 2020, "An exploratory study of augmented reality presence for tutoring machine tasks," *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–13.

[14] Genay, A., Lécuyer, A., and Hachet, M., 2021, "Being an avatar "for real": a survey on virtual embodiment in augmented reality," IEEE Transactions on Visualization and Computer Graphics, **28**(12), pp. 5071–5090.

[15] Alce, G., Hermodsson, K., and Wallergård, M., 2013, "WozARd: a wizard of oz tool for mobile AR," *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, Association for Computing Machinery, New York, NY, USA, pp. 600–605, doi: 10.1145/2493190.2494424.

[16] Nebeling, M. and Madier, K., 2019, "360proto: Making interactive virtual reality & augmented reality prototypes from paper," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, doi: 10.1145/3290605.3300826.

[17] Nebeling, M., Nebeling, J., Yu, A., and Rumble, R., 2018, "Protoar: Rapid physical-digital prototyping of mobile augmented reality applications," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, doi: 10.1145/3173574.3173927.

[18] Leiva, G. and Beaudouin-Lafon, M., 2018, "Montage: A Video Prototyping System to Reduce Re-Shooting and Increase Re-Usability," *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, Association for Computing Machinery, New York, NY, USA, pp. 675–682, doi: 10.1145/3242587.3242613.

[19] Hong, D., 2019, "Augmented Reality Prototyping Tools for Head-Mounted Displays," accessed 2021-04-05, https://medium.com/@dylanhongtech/augmented-reality-prototyping-tools-for-head-mounted-displays-c1ee8eaa0783

[20] Google, 2021, "Tilt Brush," accessed 2021-04-05, https://www.tiltbrush.com/

[21] MacIntyre, B., Gandy, M., Dow, S., and Bolter, J. D., 2004, "DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences," *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, Association for Computing Machinery, New York, NY, USA, p. 197–206, doi: 10.1145/1029632.1029669.

[22] Kelly, A., Shapiro, R. B., de Halleux, J., and Ball, T., 2018, "ARcadia: A rapid prototyping platform for real-time tangible interfaces," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–8, doi: 10.1145/3173574.3173983.

[23] Mackay, W. E., 2002, "Using video to support interaction design," DVD Tutorial, CHI, **2**(5).

[24] Leiva, G., Grønbæk, J. E., Klokmose, C. N., Nguyen, C., Kazi, R. H., and Asente, P., 2021, "Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration," *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 626–637, doi: 10.1145/3472749.3474774.

[25] Billinghurst, M. and Nebeling, M., 2022, "Rapid prototyping of XR experiences," *ACM SIGGRAPH 2022 Courses*, pp. 1–124.

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

[27] Ye, H. and Fu, H., 2022, "ProGesAR: Mobile AR Prototyping for Proxemic and Gestural Interactions with Real-world IoT Enhanced Spaces," *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, doi: 10.1145/3491102.3517689.

[28] Cao, Y., Xu, Z., Li, F., Zhong, W., Huo, K., and Ramani, K., 2019, "V.Ra: An In-Situ Visual Authoring System for Robot-IoT Task Planning with Augmented Reality," *Proceedings of the 2019 on Designing Interactive Systems Conference*, ACM, New York, NY, USA, pp. 1059–1070, doi: 10.1145/3322276.3322278, event-place: San Diego, CA, USA.

[29] Cao, Y., Wang, T., Qian, X., Rao, P. S., Wadhawan, M., Huo, K., and Ramani, K., 2019, "GhostAR: A Time-space Editor for Embodied Authoring of Human-Robot Collaborative Task with Augmented Reality," *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, pp. 521–534, doi: 10.1145/3332165.3347902, event-place: New Orleans, LA, USA.

[30] Wang, T., Qian, X., He, F., Hu, X., Huo, K., Cao, Y., and Ramani, K., 2020, "CAPturAR: An Augmented Reality Tool for Authoring Human-Involved Context-Aware Applications," *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, Association for Computing Machinery, New York, NY, USA, pp. 328–341, doi: 10.1145/3379337.3415815.

[31] Rosson, M. B. and Carroll, J. M., 2002, *Usability engineering: scenario-based development of human-computer interaction*, Morgan Kaufmann.

[32] Chi, P.-Y., Li, Y., and Hartmann, B., 2016, "Enhancing cross-device interaction scripting with interactive illustrations," *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 5482–5493, doi: 10.1145/2858036.2858382.

[33] Liu, A. L. and Li, Y., 2007, "BrickRoad: a light-weight tool for spontaneous design of location-enhanced applications," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 295–298, doi: 10.1145/1240624.1240673.

[34] Li, Y., Hong, J. I., and Landay, J. A., 2004, "Topiary: a tool for prototyping location-enhanced applications," *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pp. 217–226, doi: 10.1145/1029632.1029671.

[35] Mangano, N., LaToza, T. D., Petre, M., and van der Hoek, A., 2014, "Supporting informal design with interactive whiteboards," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 331–340, doi: 10.1145/2556288.2557411.

[36] Tharatipyakul, A., Lee, H., Zhao, S., and Davis, R. C., 2016, "Supporting the comparison of alternative stories," *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, ACM, pp. 266–270, doi: 10.1145/3010915.3010963.

[37] Kim, H.-J., Kim, C. M., and Nam, T.-J., 2018, "Sketchstudio: Experience prototyping with 2.5-dimensional animated design scenarios," *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 831–843, doi: 10.1145/3196709.3196736.

[38] Speicher, M., Lewis, K., and Nebeling, M., 2021, "Designers, the stage is yours! medium-fidelity prototyping of augmented & virtual reality interfaces with 360theater," Proceedings of the ACM on Human-Computer Interaction, **5**(EICS), pp. 1–25.

[39] Pettersson, J. S. and Wik, M., 2014, "Perspectives on Ozlab in the cloud: A literature review of tools supporting Wizard-of-Oz experimentation, including an historical overview of 1971-2013 and notes on methodological issues and supporting generic tools," .

[40] Piumsomboon, T., Lee, G. A., Hart, J. D., Ens, B., Lindeman, R. W., Thomas, B. H., and Billinghurst, M., 2018, "Mini-Me: An Adaptive Avatar for Mixed Reality Remote Collaboration," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Association for Computing Machinery, Montreal QC, Canada, pp. 1–13, doi: 10.1145/3173574.3173620.

[41] Thoravi Kumaravel, B., Anderson, F., Fitzmaurice, G., Hartmann, B., and Grossman, T., 2019, "Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence," *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, pp. 161–174, doi: 10.1145/3332165.3347872, event-place: New Orleans, LA, USA.

[42] Yu, K., Eck, U., Pankratz, F., Lazarovici, M., Wilhelm, D., and Navab, N., 2022, "Duplicated reality for co-located augmented reality collaboration," IEEE Transactions on Visualization and Computer Graphics, **28**(5), pp. 2190–2200.

[43] Maudet, N., Leiva, G., Beaudouin-Lafon, M., and Mackay, W., 2017, "Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems," *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, Association for Computing Machinery, New York, NY, USA, pp. 630–641, doi: 10.1145/2998181.2998190.

[44] Lee, G. A., Kim, G. J., and Billinghurst, M., 2005, "Immersive authoring: What you experience is what you get (wyxiwyg)," Communications of the ACM, **48**(7), pp. 76–81.

[45] Kim, Y. and Bae, S.-H., 2016, "SketchingWithHands: 3D sketching handheld products with first-person hand posture," *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 797–808, doi: 10.1145/2984511.2984567.

[46] Gupta, A., Agrawala, M., Curless, B., and Cohen, M., 2014, "Motionmontage: A system to annotate and combine motion takes for 3d animations," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2017–2026, doi: 10.1145/2556288.2557218.

[47] Saquib, N., Kazi, R. H., Wei, L.-Y., and Li, W., 2019, "Interactive body-driven graphics for augmented video performance," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, doi: 10.1145/3290605.3300852.

[48] Whitlock, M., Fitzmaurice, G., Grossman, T., and Matejka, J., 2020, "AuthAR: concurrent authoring of tutorials for AR assembly guidance," *Graphics Interface 2020*, doi: 10.20380/GI2020.43.

[49] Chidambaram, S., Huang, H., He, F., Qian, X., Villanueva, A. M., Redick, T. S., Stuerzlinger, W., and Ramani, K., 2021, "Processar: An augmented reality-based tool to create in-situ procedural 2d/3d ar instructions," *Designing Interactive Systems Conference 2021*, pp. 234–249, doi: 10.1145/3461778.3462126.

[50] Vogt, D., Stepputtis, S., Grehl, S., Jung, B., and Amor, H. B., 2017, "A system for learning continuous human-robot interactions from human-human demonstrations," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 2882–2889, doi: 10.1109/ICRA.2017.7989334.

[51] Porfirio, D., Fisher, E., Sauppé, A., Albarghouthi, A., and Mutlu, B., 2019, "Bodystorming human-robot interactions," *proceedings of the 32nd annual ACM symposium on user Interface software and technology*, pp. 479–491, doi: 10.1145/3332165.3347957.

[52] Lü, H. and Li, Y., 2013, "Gesture studio: authoring multi-touch interactions through demonstration and declaration," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 257–266, doi: 10.1145/2470654.2470690.

[53] Dey, A. K., Hamid, R., Beckmann, C., Li, I., and Hsu, D., 2004, "a CAPpella: programming by demonstration of context-aware applications," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 33–40, doi: 10.1145/985692.985697.

[54] Hartmann, B., Abdulla, L., Mittal, M., and Klemmer, S. R., 2007, "Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 145–154, doi: 10.1145/1240624.1240646.

[55] Kim, J.-W., Kim, H.-J., and Nam, T.-J., 2016, "M. gesture: an acceleration-based gesture authoring system on multiple handheld and wearable devices," *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2307–2318, doi: 10.1145/2858036.2858358.

[56] Ashbrook, D. and Starner, T., 2010, "MAGIC: a motion gesture design tool," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2159–2168, doi: 10.1145/1753326.1753653.

[57] Wang, T., Qian, X., He, F., Hu, X., Cao, Y., and Ramani, K., 2021, "Gesturar: An authoring system for creating freehand interactive augmented reality applications," *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 552–567, doi: 10.1145/3472749.3474769.

[58] Chua, P. T., Crivella, R., Daly, B., Hu, N., Schaaf, R., Ventura, D., Camill, T., Hodgins, J., and Pausch, R., 2003, "Training for physical tasks in virtual environments: Tai Chi," *IEEE Virtual Reality, 2003. Proceedings.*, IEEE, pp. 87–94, doi: 10.1109/VR.2003.1191125.

[59] Anderson, F., Grossman, T., Matejka, J., and Fitzmaurice, G., 2013, "YouMove: enhancing movement training with an augmented reality mirror," *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 311–320, doi: 10.1145/2501988.2502045.

[60] Piumsomboon, T., Lee, G. A., Irlitti, A., Ens, B., Thomas, B. H., and Billinghurst, M., 2019, "On the shoulder of the giant: A multi-scale mixed reality collaboration with 360 video sharing and tangible interaction," *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–17, doi: 10.1145/3290605.3300458.

[61] Xu, W., Liang, H.-N., Zhao, Y., Yu, D., and Monteiro, D., 2019, "DMove: Directional Motion-based Interaction for Augmented Reality Head-Mounted Displays," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Association for Computing Machinery, New York, NY, USA, pp. 1–14, doi: 10.1145/3290605.3300674.

[62] Edwards, R., Bybee, B. T., Frost, J. K., Harvey, A. J., and Navarro, M., 2017, "That's not what I meant: How misunderstanding is related to channel and perspective-taking," Journal of Language and Social Psychology, **36**(2), pp. 188–210.

[63] Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., and Zigelbaum, J., 2008, "Reality-based interaction: a framework for post-WIMP interfaces," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 201–210, doi: 10.1145/1357054.1357089.

[64] FIORELLA, L., 2022, "23 The Embodiment Principle in Multimedia Learning," .