



Pattern-Based Augmented Reality Authoring Using Different Degrees of Immersion: A Learning Nugget Approach

Linda Rau*, Dagny C. Döring, Robin Horst and Ralf Dörner

Design, Computer Science, Media, RheinMain University of Applied Sciences, Wiesbaden, Germany

OPEN ACCESS

Edited by:

George Papagiannakis,
University of Crete, Greece

Reviewed by:

Michele Fiorentino,
Politecnico di Bari, Italy
Eleni Kyza,
Cyprus University of Technology,
Cyprus

*Correspondence:

Linda Rau
linda.rau@hs-rm.de

Specialty section:

This article was submitted to
Technologies for VR,
a section of the journal
Frontiers in Virtual Reality

Received: 21 December 2021

Accepted: 03 March 2022

Published: 28 March 2022

Citation:

Rau L, Döring DC, Horst R and
Dörner R (2022) Pattern-Based
Augmented Reality Authoring Using
Different Degrees of Immersion: A
Learning Nugget Approach.
Front. Virtual Real. 3:841066.
doi: 10.3389/frvir.2022.841066

Creating Augmented Reality (AR) applications can be an arduous process. With most current authoring tools, authors must complete multiple authoring steps in a time-consuming process before they can try their AR application and get a first impression of it. Especially for laypersons, complex workflows set a high barrier to getting started with creating AR applications. This work presents a novel authoring approach for creating mobile AR applications. Our idea is to provide authors with small, ready-to-use AR applications that can be executed and tested directly as a starting point. Authors can then focus on customizing these AR applications to their needs without programming knowledge. We propose to use patterns from application domains to further facilitate the authoring process. Our idea is based on the learning nugget approach from the educational sciences, where a nugget is a small and self-contained learning unit. We transfer this approach to the field of AR authoring and introduce an AR nugget authoring tool. The authoring tool provides pattern-based self-contained AR applications, called AR nuggets. AR nuggets use simple geometric objects to give authors an impression of the AR application. By replacing these objects and further adaptations, authors can realize their AR applications. Our authoring tool draws from non-immersive desktop computers and AR devices. It synchronizes all changes to an AR nugget both to an AR device and a non-immersive device. This enables authors to use both devices, e.g., a desktop computer to type text and an AR device to place virtual objects in the 3D environment. We evaluate our proposed authoring approach and tool in a user study with 48 participants. Our users installed the AR nugget authoring tool on their own devices, worked with it for 3 weeks, and filled out a questionnaire. They were able to create AR applications and found the AR nugget approach supportive. The users mainly used the desktop computer for the authoring tasks but found the synchronization to the AR device helpful to experience the AR nuggets at any time. However, the users had difficulties with some interactions and rated the AR nugget authoring tool in a neutral field.

Keywords: augmented reality, nuggets, authoring, immersion, preview

1 INTRODUCTION

Recently, interest in Augmented Reality (AR) applications is increasing in several application fields. To build such AR applications, practical AR authoring tools are needed. However, several challenges and complex workflows in creating Augmented Reality (AR) applications hinder its adoption in the mainstream. Authors of AR applications typically run through a time-consuming process with multiple steps and a complex workflow before they can try their AR application to get a first impression of it. Additionally, finding the right tools and getting started can be challenging. If authors are not experienced with AR, these challenges are high barriers.

Our basic idea is to provide small, ready-to-use AR applications that we call AR nuggets. This work introduces an AR authoring tool, that is, based on AR nuggets. The idea of AR nuggets is based on Virtual Reality (VR) nuggets (Horst, 2021) and microlearning from the educational sciences. In microlearning, a nugget is a small and self-contained learning unit. Similarly, an AR nugget is a small stand-alone AR application (Rau et al., 2021). One AR nugget reflects a single pattern from the application domain, for example, adding virtual labels to an object or comparing a real-world object with a virtual one. An AR nugget authoring tool can provide multiple AR nuggets, one per pattern, to give authors a range of different application scenarios to choose from for their needs. As AR nuggets are always executable, they allow authors to try the pattern-based AR application before investing time in authoring steps. By this, the authors can get a first impression of their AR application and see if the authoring tool supports their idea. We strive to support especially authors with no or little experience in AR. Employing our tool, authors are faced only with a subset of tasks in the field of customization (instead of creation), e.g., replacing virtual objects or adapting default values. While authoring, the AR nugget remains executable and can always be tested. This work applies the concept of AR nuggets to an AR authoring tool and makes the following contributions:

- We introduce a novel authoring tool based on AR nuggets. The authoring tool is accessible for laypersons as it requires no programming knowledge. It reflects the concept of AR nuggets: small, stand-alone AR applications that are always executable.
- Additionally, we propose an authoring workflow that can be applied to the authoring tool.
- We explore authoring AR nuggets on non-immersive desktop PC technologies and AR devices simultaneously. Here, we introduce a synchronized preview function that offers a what-you-see-is-what-you-get experience.
- Although AR nuggets are stand-alone applications, it can be desired to integrate them into existing applications. For example, an existing e-learning application can be extended or updated with AR nuggets. To allow loading AR nuggets within another ready-built application, we first implement an export functionality within our AR nugget authoring tool that saves AR nuggets in an exchange format. Next, we implement a content delivery system and integrate it in a

tool-chain that delivers AR nuggets from the authoring tool to another ready-built application.

In the next section, we analyze related work. Next, we elaborate the authoring idea for AR nuggets from a conceptual point of view in **Section 3**. In **Section 4**, we present the implementation of the AR nugget authoring tool with three different types of AR nuggets. **Section 5** describes an evaluation in a user study with 48 authors and discusses the results. Finally, **Section 6** concludes this work and points to directions for future work.

2 RELATED WORK

Ashtari et al. (2020) analyzed challenges that authors of AR and VR applications face. They interviewed authors from different backgrounds from hobbyists to professionals that described similar challenges. For the group of domain experts, it was difficult to translate their domain-specific knowledge to AR or VR applications. One key barrier that they identified is that authors did not know how to get started and what tools and hardware suited their needs. For example, one of their participants found out that the chosen toolkit did not support what they planned to do with it after half of the implementation process. Furthermore, authors had no or little knowledge of how to design a good AR application due to a lack of guidelines and example projects. Challenges and opportunities especially for cross-device authoring were discussed by Speicher et al. (2018). They analyzed existing frameworks and survey challenges with current authoring tools. It was a challenging task for authors to be aware of different available options and frameworks and to know how to combine them. Some implementation strategies for AR authoring were investigated by Ledermann et al. (2007). They found fundamental aspects of AR were the co-presence of the real world with virtual content as well as the heterogeneous hardware and interaction techniques. Authors need to be aware of this and to be provided with strategies and concepts to target various hardware or devices. Furthermore, Ledermann et al. explained that there was a lack of both authoring tools and techniques to structure AR content. They concluded that conceptual models and corresponding workflows should be integrated into an AR authoring tool.

A wide range of AR authoring tools that target parts of those challenges has been developed in the last 2 decades, each focusing on different target authors and use cases. Work by Hampshire et al. (2006) introduced a taxonomy for different levels of abstraction in authoring AR applications. The authors differentiated between high-level and low-level tools with different abstraction levels of application interfaces and concepts. This divided authoring tools into high and low-level programming and content design tools. Furthermore, they proposed requirements for future AR authoring tools that targeted rapid application development. Content design authoring tools could help to make AR accessible not only for programming or AR experts but also for authors from other fields of knowledge or laypersons. Similarly, Roberto et al. (2016)

distinguished programming tools from content design tools. They focused on content design tools and identified two authoring paradigms and two deployment strategies. Based on this, they divided the content design tools into stand-alone or plug-in and platform-specific or platform-independent tools.

Wang et al. (2010) implemented an AR authoring tool with a graphical UI that supported teachers and students without programming knowledge in enhancing digital examination applications with AR. Besides the authoring tool, they also implemented an AR viewing interface and distinguished between authoring and viewing mode. Their tools exported and imported 3D model files as well as text, which was part of the AR application, in a text file to and from the authoring interface and to the viewing interface. This allowed easy and fast modification of the questions and 3D models. However, their authoring tool was limited to the creation of examination AR applications. Similarly, the content design framework SimpleAR (Apaza-Yllachura et al., 2019) distinguished between an editor and a viewer application that communicated through a Firebase real-time database. It targeted authors with basic computer skills in any application field. The editor application was based on components, e.g., “augment a marker.” It provided a 2D user interface (UI) on a web development platform, making the framework stand-alone and platform-independent. On the platform, users could add resources like 3D models as well as select and configure components. In a user study, the work showed that all users were able to complete the study’s task and were able to create an AR application in less than 5 min. However, their study also showed that the users had problems understanding the components’ concept. Another AR authoring tool, called ARTutor, was targeted to the field of education (Lytridis et al., 2018). The web-based authoring application allowed authors to create AR applications for textbooks that could be suitable for distance learning. It was accompanied by a mobile application that was used to access the authored educational AR applications. In a user study (Lytridis et al., 2018), the authors found that users were able to enhance textbooks with virtual augmentations using the authoring tool. Users were also able to view and use their authored AR application through a mobile device. While this authoring approach was suitable for persons without programming knowledge, it was limited to basic augmentations for textbooks. Gattullo et al. (2020) aimed to support technical writers of AR interfaces in conveying information to the interfaces’ users. Based on a study, the authors gathered information on which virtual assets were preferred for certain AR issues in an industrial context. This information could help other technical writers or laypersons in the field of AR to choose between various visual assets according to the information type. Work by Laviola et al. (2021) investigated an authoring approach called “minimal AR” that provided minimal information through AR visual assets to support users in accomplishing a task. In a user study, the authors found that users’ performance does not significantly increase when providing more than the minimal information, that is, required to accomplish the task. Based on these outcomes, authoring effort and difficulties could be reduced. Lee et al. (2004) applied immersive authoring to the

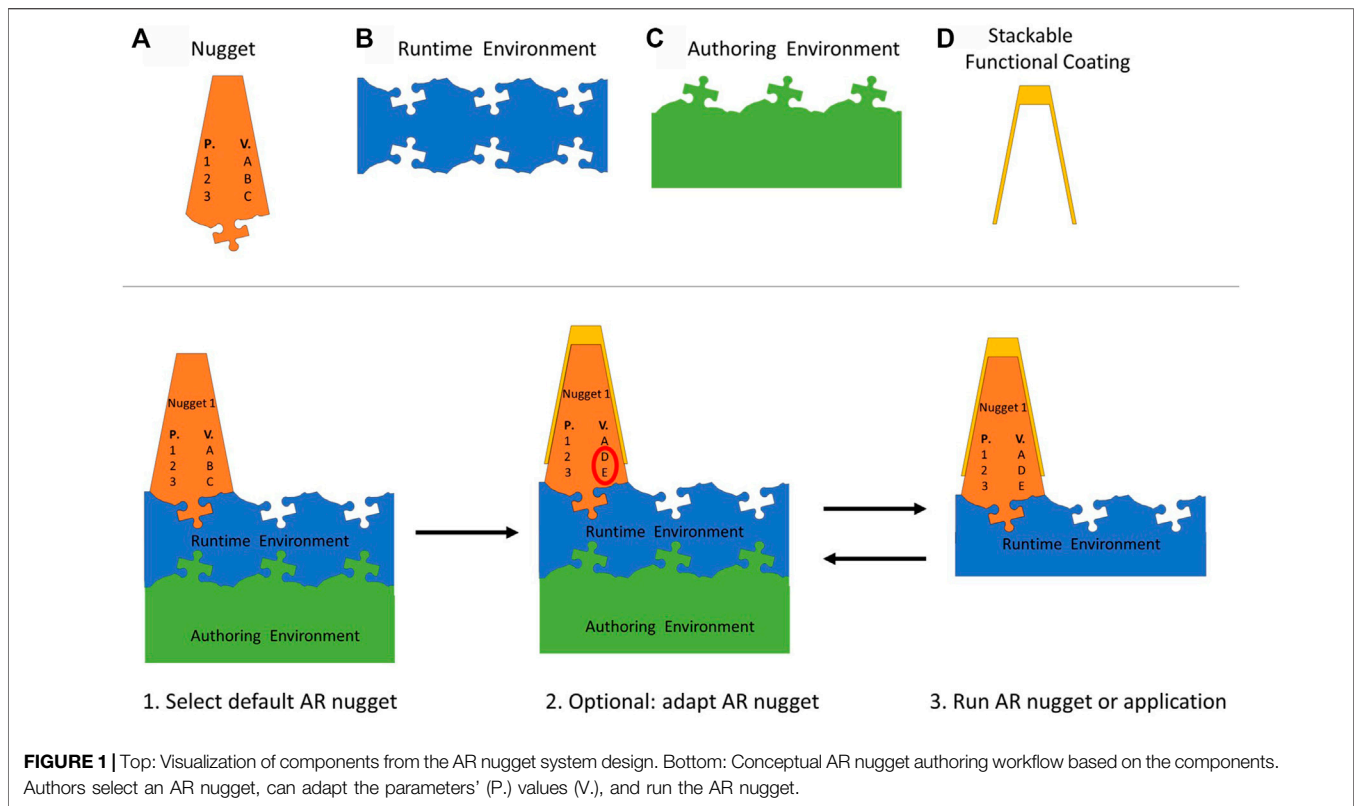
authoring of tangible AR applications, including the AR applications’ behaviors and interactions. They evaluated their authoring approach in a user study with users who are no experts in programming. All of their participants were able to create an AR scene during their user study, which indicated that their authoring process is suitable for persons without programming knowledge. While most of the users (42%) preferred the immersive interface over another non-immersive interface, a third of the users stated to appreciate a mixture of the immersive and a non-immersive interface where they could toggle between traditional input (over keyboard and mouse) and tangible AR input.

Authoring tools for VR by Horst and Dörner (2019), Horst et al. (2020) applied the idea of nuggets to VR within the concept of VR nuggets (Horst, 2021). The concept dovetailed the educational authoring of learning nuggets from the e-learning domain and transferred it to the area of VR authoring. VR nuggets were characterized by conveying relatively short VR experiences based on design patterns. The initial VR-nugget-based authoring tool, VR Forge (Horst and Dörner, 2019), provided components of a VR nugget with given parameters already filled with placeholder content so that a VR nugget remains in a usable state during the authoring process. Its UI was inspired by ones of established slideshow authoring software such as PowerPoint or Keynote. The second authoring tool, IN Tiles (Horst et al., 2020), provided authors with an alternative UI based on immersive technologies to create VR experiences based on VR nuggets. Both tools provided authors the possibility to preview and test the VR experiences throughout the process. However, authors of AR applications face additional challenges that authoring tools for VR applications cannot address. For example, where in the user’s environment should virtual content be placed? How should the system react if the user’s environment changes? How can users interact with real objects in their environment and with virtual objects?

3 AUGMENTED REALITY NUGGET BASED AUTHORING

AR nuggets (Rau et al., 2021) transfer the idea of VR nuggets (Horst, 2021) to AR. AR nuggets are small stand-alone AR applications based on patterns. Each AR nugget reflects one scenario, that is, recurring in various AR applications. An AR nugget always includes objects with parameters and a target where virtual content can be anchored in the real world. Authors can customize the AR nugget to create their own AR applications. We differentiate between default AR nuggets and adapted AR nuggets. A default AR nugget uses only default objects, parameters, and targets. Once an author made any change to a default AR nugget, e.g., replacing a default object with a custom one, we refer to the AR nugget as adapted AR nugget.

With an authoring tool for AR nuggets, we aim to support authors without programming knowledge in customizing AR nuggets to their own needs. One could also create and adapt AR nuggets using existing tools like game engines. However, working



with game engines is not suitable for laypersons and usually requires programming skills. Therefore, an authoring tool for AR nuggets is necessary. An AR nugget authoring tool should reflect the system design of AR nuggets. This section describes a system design for AR nuggets and how it can be applied to an AR nugget authoring tool.

We visualize an AR nugget as a component in our system design in **Figure 1** (top, A). In this figure, parameters (P.) are visualized with numbers and their values (V.) with letters. For example, a parameter could be a label and the text its value. Besides the AR nugget, our system design includes a runtime environment, an authoring environment, and optional functional coatings. These parts are visualized in **Figure 1** (top) and described in the following.

AR nuggets are connected to the runtime environment [**Figure 1** (top, B)] which executes the AR nuggets on a target AR device. We indicate this connection with the components' puzzle form. The runtime environment realizes general system functionalities such as memory management, data handling, or data processing. Additionally, it handles AR-specific tasks like detecting and tracking real-world anchors such as images, objects, or surfaces. Multiple AR nuggets can be connected to the same runtime environment to be executed successively within one application. In this case, the runtime environment manages that only one AR nugget is active at a time. Similar to nuggets in microlearning, multiple AR nuggets remain independent from each other. The independence allows to customize one AR nugget or to delete one AR nugget without changing anything in another nugget. Additionally, an AR

nugget can be copied to another application or use case without regard to other dependencies.

To customize AR nuggets, the runtime environment can be extended with an authoring environment [**Figure 1** (top, C)]. This is also indicated by the components' puzzle form. Using the authoring environment, authors can assign new values to the AR nugget's parameters or replace placeholder objects with custom 3D objects. While authors adapt the AR nugget, it remains executable.

Functional coatings [**Figure 1** (top, D)] can be used to add additional functionalities to AR nuggets. They are optional and do not change the AR nugget's goal or motivation. Each functional coating extends one virtual object in an AR nugget, e.g., a 3D model or a label, with one additional functionality. Functional coatings are stackable so that multiple of them can be applied to a single virtual object. For example, a functional coating can add a fade-in effect to a virtual object. When the AR nugget is started, this virtual object would not be visible from the start, but slowly become visible through a fade-in effect. Another example for a functional coating is an indicator that points towards a virtual object when the virtual object is not in the AR device's field of view. Both of these examples can be applied independently from each other to the same virtual object within one AR nugget.

When the authoring tool is started, authors can choose one default AR nugget that they want to experience or adapt to their own needs. Authors may also use the AR nugget authoring tool to create multiple AR nuggets that can be experienced successively in one application. For example, an author could choose to start

with a quiz AR nugget. The default quiz AR nugget is initialized with default objects and default values and therefore immediately executable. It could be initialized with two default cubes in different colors and a default quiz question like “Which cube is black?” Authors can tap or click on the cubes to answer the quiz question, then the tapped or clicked cube will blink in green color, if it was the right answer, or red color, if it was the wrong answer. While the authoring tool allows customizing, replacing, and in some cases adding objects, it should not allow deleting required objects. For example, a quiz AR nugget needs at least one 3D model that belongs to the category “wrong” and one for the category “right.” It should not be possible to delete all objects in both categories. Instead, one object per category needs to remain to keep the AR nugget executable. Authors would need to replace it instead of deleting it. This restriction supports authors because it ensures that the AR nugget remains executable. If authors intend to delete one of these objects and to use the AR nugget in another way, the AR nugget authoring tool could suggest another type of AR nugget. If no suitable pattern is available, a new default AR nugget needs to be created by programming experts.

3.1 Workflow

To adapt an existing default AR nugget, authors can replace its placeholder objects and adapt its parameters. In the following, we describe this process from an author’s point of view and visualize it in **Figure 1**.

As the first step, the authors select a default AR nugget. In **Figure 1**, the AR nugget has three default parameters (P.) with three values (V.) assigned to them. The figure visualizes one AR nugget connected to the runtime environment, but further AR nuggets can be added as indicated by the puzzle form.

In an optional second step, the authors can adapt the default parameters’ values to custom values and replace placeholder objects. Adapting an AR nugget can include replacing the target where virtual content is anchored in the real world, replacing default objects, moving, rotating, or scaling 3D objects, or adapting other values. For these actions, authors are not required to have programming skills or AR knowledge. However, they are required to use UI elements, e.g., buttons or other graphical elements, to perform these actions. In general, the adapting steps can be executed in any order. Still, it can be supportive to suggest an adapting workflow and order, especially for inexperienced authors. Our idea is to start with steps that make prominent changes and suggest starting with adapting the target where virtual content should be anchored. Next, the 3D model or models can be replaced and nugget-specific parameters, e.g., the quiz question or labels, should be adapted. Finally, virtual objects should be moved, rotated, and scaled before functional coatings are applied. **Figure 1** shows a red circle in step 2 where values were adapted. Additionally, functional coatings can be added. In **Figure 1**, we add one functional coating. AR nuggets can also be executed before authors adapt anything so that this second step is optional. It can be helpful to execute an AR nugget before adapting it to get an impression and to test it. Because all necessary scripts are included within the provided AR nuggets, adapting AR nuggets requires no coding.

Next, authors can export and test the AR nuggets in a runtime environment. From here, authors can always go back to step 2 to further adapt the AR nugget. Because AR nuggets always remain in an executable state, each AR nugget can be executed before and after each step. This authoring process applies to each AR nugget that an author selects and adapts.

3.2 Preview

To allow authors to experience their AR nuggets at any point during the development process, our idea is to provide a preview functionality. AR nuggets can be edited or viewed directly on the computer authoring device. By this, the interface for the computer already includes a preview. Additionally, we propose to incorporate a preview on an AR device. Therefore, we propose to implement one authoring tool for a desktop computer that can be extended by using the same authoring tool on an AR device. The tools need to synchronize changes to the AR nuggets between each other.

Because we see the AR device as optional, all authoring functionalities should be available on the desktop computer. Besides experiencing AR nuggets on the AR device, it could also be possible to edit them using the AR device. To make all authoring functionalities also available on the AR device, the AR device’s screen is likely too small to fit them all in a suitable size. For example, only a few buttons can be fitted on an AR device screen without occluding the AR content, that is, also rendered on the screen. However, some authoring functionalities can be easier for authors to carry out on an AR device than on a desktop computer. For AR nuggets, we propose to enable movement, rotation, and selection of objects on the AR device. For AR nuggets that support user input, it is important to separate authoring mode and edit mode. Otherwise, user input could be mistaken as authoring input or the other way round.

4 IMPLEMENTATION

Our work implements an authoring tool for AR nuggets and exemplary includes three default AR nuggets based on three application patterns. We describe the three application patterns and how we implemented authoring functionalities for each in the following subsections. This section describes parts of the implementation of our AR nugget authoring tool that are valid for all AR nuggets and patterns.

For the implementation of our default AR nuggets, we use the game engine Unity (2021). This allows utilization of core functionalities from Unity, e.g., a runtime environment for AR devices like smartphones or tablets. With this, our AR nuggets can be executed on different hardware devices. As tracking toolkit, we chose to use Vuforia Engine (PTC, accessed: 02.08.2021). To implement a default AR nugget based on a new pattern, we create a new Unity scene for the AR nugget. Based on the pattern, we define the AR nugget’s specific placeholder objects and default values for their parameters. Next, we place these placeholder objects in the Unity scene and set their parameters’ default values. If a pattern involves user interactions, for example, like in a quiz AR nugget, we define

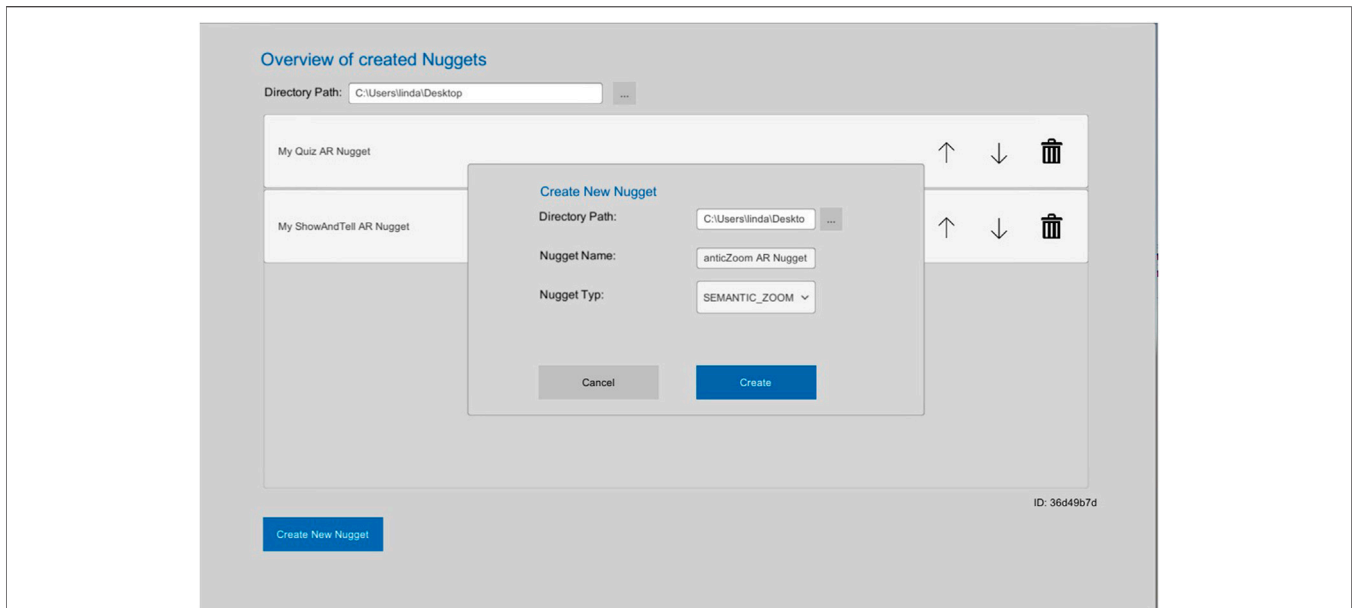


FIGURE 2 | Screenshot of the AR Nugget authoring tool on the desktop device. Authors can add new AR nuggets through the drop-down menu and select an AR nugget to edit from the list.

and implement these. All of the placeholder objects are saved as templates with Unity prefabs. This allows initializing additional placeholders by drag and drop. For example, additional virtual labels can be initialized based on their prefab. The prefabs contain scripts that aid to contribute to a sophisticated AR nugget. For example, all AR nuggets that include labels automatically adapt the labels' size to their texts.

To implement the AR nugget authoring tool, we also use the game engine Unity (Unity, accessed: 06.05.2021) and build the AR nugget authoring tool for a desktop computer as well as for an AR device. Authors install and start the authoring tool on both devices. The desktop computer screen displays a server ID and an AR nugget overview. The latter shows all created AR nuggets and allows to create a new one. On the starting screen on the AR device, authors are asked to type in the server ID to connect to the computer authoring device. They can connect the AR device at any time once the application on the desktop device is running. When connected, all changes to AR nuggets are synchronized between both devices using the multiplayer plugin Photon BOLT Photon (2022). The desktop computer hosts a BOLT session that the AR device connects to. The changes are then synchronized between clients using shared states and event-based communication. For our prototype, we limited the number of connections to one AR device. New AR nuggets can be created using the desktop authoring device (see **Figure 2**). For this, authors can choose a default AR nugget from a drop-down menu. The drop-down menu displays all available default AR nuggets, i.e., all available AR nugget types. Then, authors may enter a custom name for their AR nugget. The AR nugget is then added to a list with all AR nuggets that the author created. By clicking on the AR nugget from the list, authors can start it and immediately experience it. If an author creates multiple AR

nuggets, the authoring tool allows them to customize their order of succession by customizing their order in the list. However, each AR nugget remains its own stand-alone application.

The authoring tool's UI as used for the desktop authoring device is depicted in **Figure 3**. The figure shows screenshots of default AR nuggets for each AR nugget type. As an anchor to place virtual content in the real world, we use an image target, by default, an image target that shows some stones (PTC, 2021). The AR nugget authoring tool's UI suggests the workflow as proposed in **Section 3.1** by the order of the menus on the left side. As shown in **Figure 3**, the top left menu Target Selection allows to replace the default image target with a custom one. When authors click the replace button, a file explorer opens where authors can choose the custom image that they want as a replacement. For the quiz AR nugget (**Figure 3B**), the menu below allows editing the quiz question. Below the Target Selection and Quiz Question menu is the Model Selection menu that allows replacing default 3D objects. In case of the quiz AR nugget, it is divided in a menu for each category right and wrong. Similarly, for the semantic zoom AR nugget, the menu is divided in Model Selection and Semantic Zoom Objects. Akin to replacing 3D objects, authors click the replace button which opens a file explorer. Here, they can navigate to and choose custom 3D objects. Depending on the application pattern the AR nugget is based on, additional 3D models can be added and deleted again by clicking the buttons.

At the bottom of the UI are the menus for movement, rotation, and scale to fine-tune the adapted AR nuggets. In **Figure 3A**, the default virtual object, a cube, is selected. To select or deselect objects, authors can click them. Selected objects can be moved, rotated, and scaled using the buttons on the lower left side or the keys on the computer's keyboard. The buttons for movement,

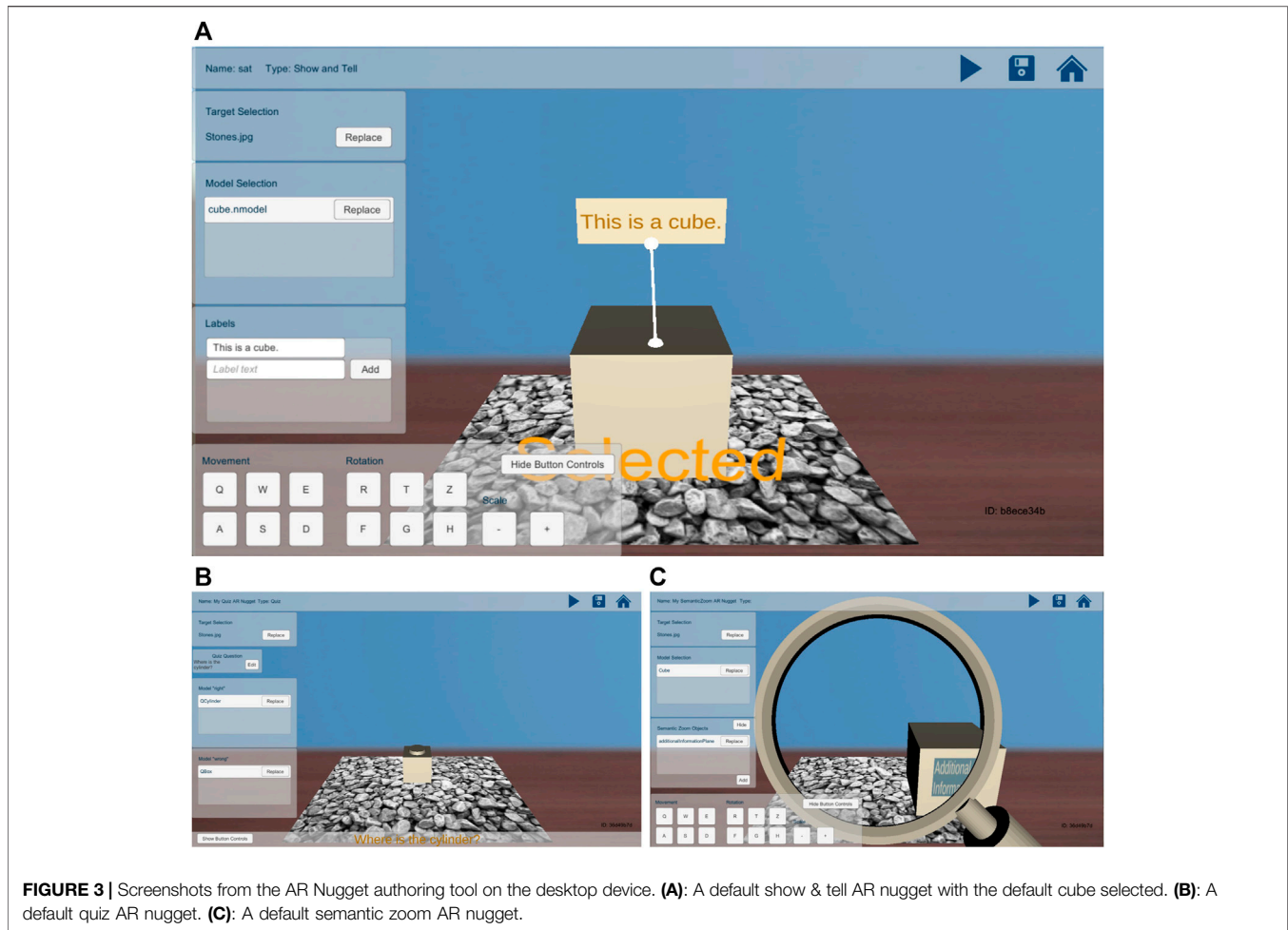


FIGURE 3 | Screenshots from the AR Nugget authoring tool on the desktop device. (A): A default show & tell AR nugget with the default cube selected. (B): A default quiz AR nugget. (C): A default semantic zoom AR nugget.

rotation, and scaling can be hidden from the UI as shown in the quiz AR nugget in **Figure 3B**.

Authors can switch to and back from the edit mode by clicking the play/pause button at the top right corner of the UI. The top right corner also provides buttons for saving and for going back to the AR nugget overview. The lower right corner shows the server ID, that is, required to connect the AR device with the desktop device.

For the AR device, we decided to exclude the UI elements that allow to replace, add, or delete custom 3D objects for the AR device's UI to be able to use the limited screen space for viewing the camera stream and augmentations better. Furthermore, custom 3D objects are more likely to be stored on a computer than on an AR device. However, moving and rotating objects is an interaction that may be easier on a device that allows direct interactions *via* touch-based gestures. Therefore, we included UI elements for movement and rotation on the AR device. Furthermore, we utilize LeanTouch (Wilkes, 2021) to make use of touch controls that are an established input method on mobile devices. When authors tap and hold an object on the AR device's screen, they can move it by moving their finger. Rotating and scaling is possible in a similar way, but using two fingers.

4.1 Show & Tell Augmented Reality Nugget

Per default, the show and tell AR nugget is initialized with a cube and one label that reads "This is a cube." The UI for it includes one Model Selection and a list with labels. The cube model is listed in the Model Selection. Next to it is a replace button. If authors click it, again a file explorer is opened to search for a custom 3D object. All labels appear in a list below the Model Selection. New labels can be added by typing the label text and clicking on the add button. The label text of existing ones can be edited through the text boxes in the label list. The text's size automatically adapts to the label's size. Labels can also be deleted, however, the system ensures that one label always remains to keep the AR nugget executable while sticking to the show & tell pattern. If the author deletes all except one label, there will be no delete button for the last label.

Similar to 3D objects, labels can be moved and rotated by clicking the buttons on the UI or using the keyboard. For this, a label needs to be selected. Selecting a label is possible by clicking it. Then, the selection is indicated with a blue color. To anchor a label to the 3D object, authors click or tap the label's anchor point and then click or tap on the 3D object where they want to draw the line to. After both clicks, a white line that connects one label and the 3D object appears. The label's anchor at the 3D object can

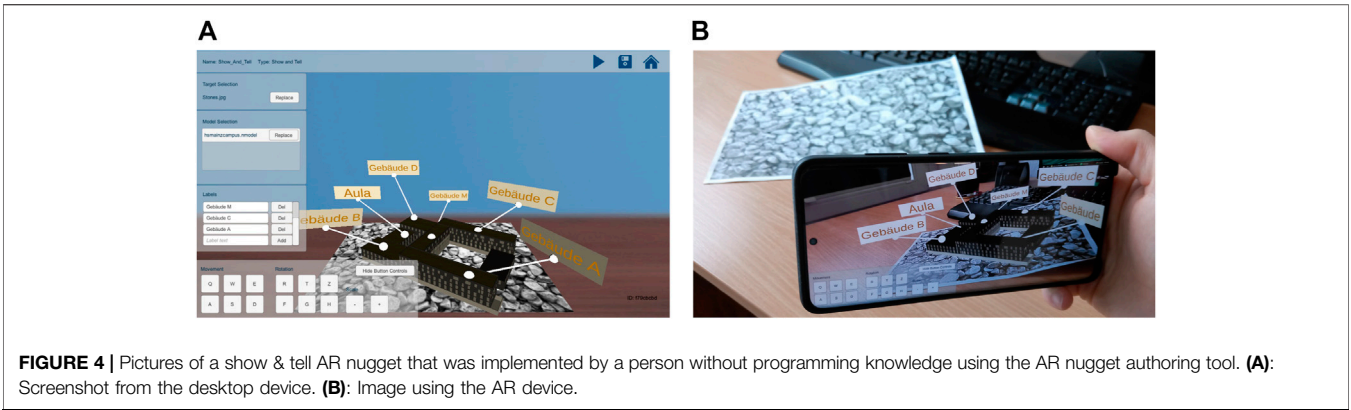


FIGURE 4 | Pictures of a show & tell AR nugget that was implemented by a person without programming knowledge using the AR nugget authoring tool. **(A):** Screenshot from the desktop device. **(B):** Image using the AR device.

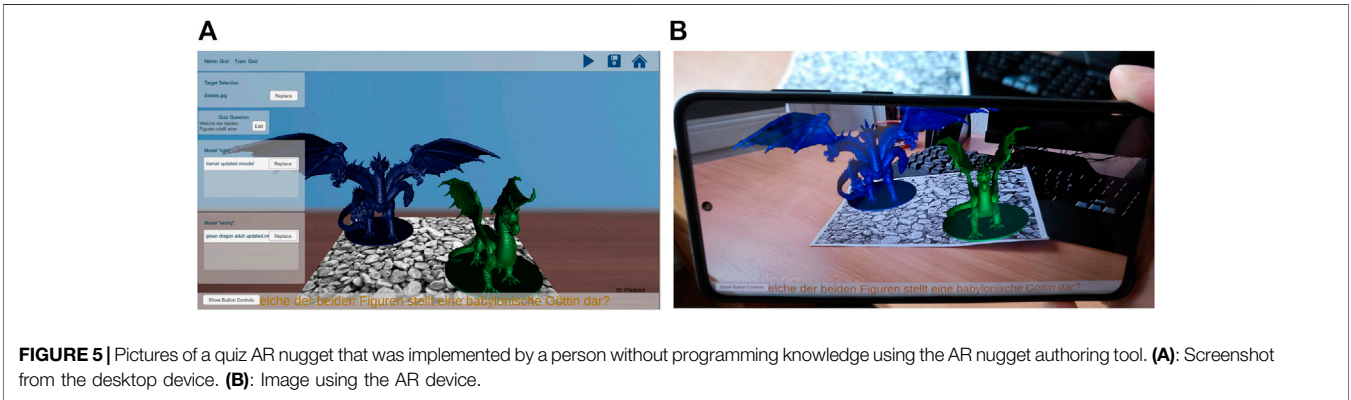


FIGURE 5 | Pictures of a quiz AR nugget that was implemented by a person without programming knowledge using the AR nugget authoring tool. **(A):** Screenshot from the desktop device. **(B):** Image using the AR device.

also be selected and moved. Placing these label anchors is possible on the desktop device as well as on the AR device.

Figure 4 shows one example of a show & tell AR nugget that was created from a person with no programming knowledge using the AR nugget authoring tool. The author kept the default image target and replaced the default cube with a 3D model of a building. The building was rotated and each of its locks was labeled, e.g., “block A,” “block B,” and “assembly hall.” All labels were moved and rotated.

4.2 Quiz Augmented Reality Nugget

The quiz AR nugget is initialized with a cube, a cylinder, and the quiz question “Where is the cylinder?” The cylinder belongs to the category right models while the cube belongs to the wrong models. So, when authors click the cylinder, it blinks in green color, while the cube blinks in red color when clicked or tapped.

Using the Model Selection menus, authors can replace the models for each of the two categories with the replace button. Similar to the labels in the show & tell AR nugget, additional models can be added and deleted. However, the system ensures that one model always remains to keep an executable quiz AR nugget. The quiz question can be edited by clicking the edit button and customizing the text. Models from both categories, “right” and “wrong,” can be moved, rotated, and scaled when selected.

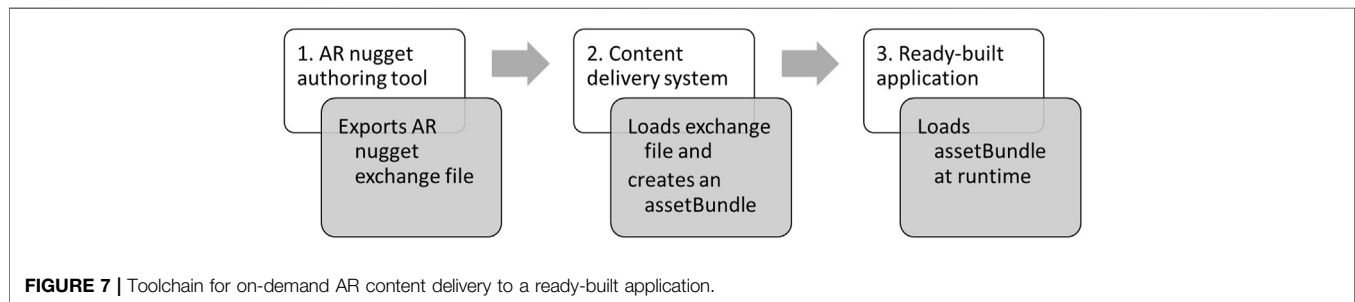
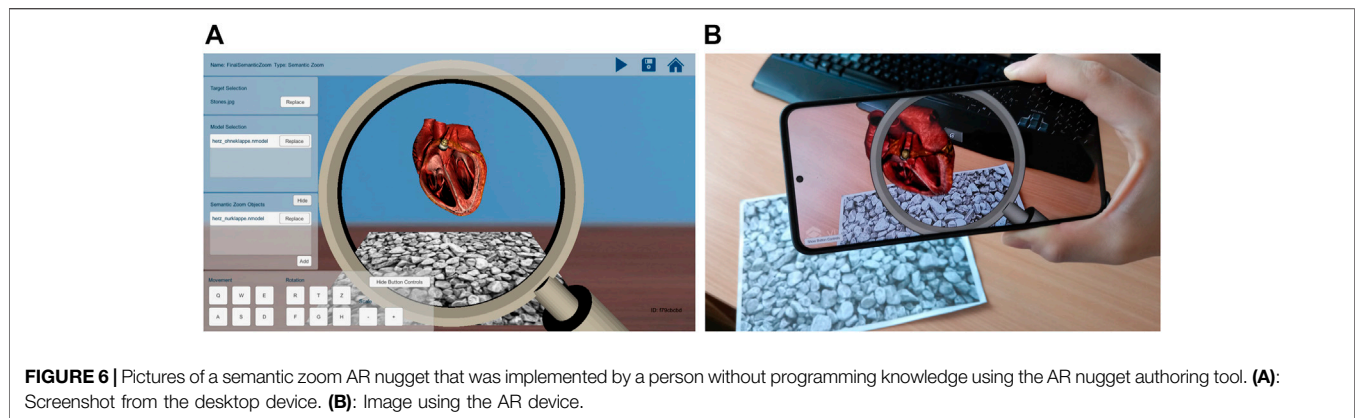
One example of an adapted quiz AR nugget is visualized in **Figure 5**. One person with no programming knowledge created

this AR nugget using the AR nugget authoring tool. The quiz AR nugget asks “Which of these two figures is a Babylonian goodness?” in German. If users click or tap the left figure, it will blink in green color because the author defined this as the right answer. Tapping or clicking the right figure will make it blink in red color.

4.3 Semantic Zoom Augmented Reality Nugget

The semantic zoom AR nugget is initialized with a cube and an image with the text “additional information” on it. The image belongs to the category semantic zoom objects and is only visible through the magnifying glass, which is also included in the AR nugget. The cube is visible outside of as well as through the magnifying glass. Models that are visible through the magnifying glass, as well as semantic zoom objects, can be replaced. Additional semantic zoom objects may be added, but when deleting them the system ensures that one semantic zoom object remains. To easier select the objects, authors can hide the magnifying glass by clicking a button.

Figure 6 shows an adapted semantic zoom AR nugget that was created by a person who was no programming expert using the AR nugget authoring tool. The cube was replaced with a 3D model of a human heart. As a semantic zoom object, a stent with an aortic valve was added. As visible on the figure’s right side, the



heart is visible through the magnifying glass as well as closer to the edges of the AR device's screen. In contrast, the stent is only visible through the magnifying glass.

4.4 Toolchain

AR nuggets include a runtime environment so that they can directly be deployed to AR devices. However, in some cases, it can be desired to integrate AR nuggets in an existing AR application. For example, AR nuggets can be used in an educational context and integrated into a learning app, that is, already familiar to the users. Also, it can be necessary to update AR nuggets within an existing application or to add further AR nuggets there. To support this scenario, we develop a toolchain, that is, visualized in **Figure 7**. The AR nugget authoring tool needs the functionality to save AR nuggets to an exchange format to deploy and integrate them. Our approach is to export each AR nugget in a separate file that includes a unique name, the type of pattern it is reflecting, its virtual objects, a path to the real-world anchor (for example, to the image target), and its nugget-specific parameters. For this, we develop an AR nugget exchange format and implement it within our AR nugget authoring tool (**Figure 7** left).

Furthermore, we develop an additional application that serves as a content delivery system. It can import AR nuggets that were created using the AR nugget authoring tool and rebuilds a Unity scene from the AR nugget exchange file (**Figure 7** middle). This allows authors with programming experience to further adapt AR nuggets using Unity if desired. From here, we can export the Unity scene with the final AR nugget to an

assetBundle. An assetBundle is a Unity-specific file format that can contain assets like 3D models and can be loaded at runtime. Finally, AR nuggets can be loaded into ready-built applications at runtime using on-demand content delivery via assetBundles (**Figure 7** right).

5 EVALUATION

In a user study, we evaluate if the AR nugget approach supports authors without programming experience in creating their own AR applications. Additionally, we evaluate the AR nugget authoring tool regarding pragmatic and hedonic qualities. The next subsection describes the user study's procedure and a questionnaire that the users were asked to fill out. Then, we present the study's results and discuss them in the following two subsections.

5.1 Methods

We evaluate our authoring approach and AR nugget authoring tool in a user study with 48 voluntary, unpaid participants. The participants are aged between 20 and 34 years, with \bar{x} 24.20 and SD 3.07 years, and are no programming experts. Most of the participants are students with knowledge in business administration and industrial economics. They rate their experience with AR in general \bar{x} 1.7083, SD 1.0793 and with creating AR applications \bar{x} 1.5417, SD 1.0198 on a Likert scale from 1 (no experience) to 7 (experienced, using AR at least once a week).

We offered the participants to show a demo of the AR nugget authoring tool so that they could decide if the authoring tool suits their needs. If a participant decided to participate in the user study, they were provided with the install files for desktop and AR devices, a selection of image targets and 3D models, as well as instructions on how to install the authoring tool. Additionally, the instruction included details about the user study's procedure, contact information, a link to an online questionnaire, and described the idea of AR nuggets shortly. We asked the participants to try the AR nugget authoring tool for 3 weeks and to create at least one AR nugget per type, i.e., three or more AR nuggets. After the 3 weeks, participants were asked to answer an online questionnaire and optionally provide their AR nuggets, images, screenshots, and a log file of their work with the AR nugget authoring tool. The log file included 1) for much time the authoring tool was used, 2) which actions (selecting, replacing, moving, rotating, or scaling objects) were performed, and 3) whether actions were performed on the computer or AR device. The questionnaire could be answered at any time to fit the participants' individual time schedules. Also, testing the AR nugget authoring tool and answering the questionnaire was not bounded to certain locations and could be fulfilled from home. However, the self-report nature of questionnaires also brings limitations. Users may answer with what they think is expected from them rather than truthfully, the wording of questions may be confusing, or users might tend to give only extreme or middle ratings to all questions. We mitigated these limitations by having the questionnaire answered anonymously to avoid people would feel pressured to give certain answers. The optional log file was stored in a text format and users were able to read it before uploading, so they could make sure they did not upload any personal data. Additionally, we checked if statements that our users made in the questionnaire fitted to the uploaded log files and AR nuggets and found no discrepancies. However, some users did not upload the optional files or uploaded incomplete files that could not or only partly be included in our analysis.

Our questionnaire starts with the 16 items from the AttrakDiff's short version (Hassenzahl et al., 2003). Next, it asks the questions listed below. If not indicated otherwise, questions were asked on a 7-point Likert scale. While Q1–Q7 refer to all AR nugget types, the following questions Q8–Q17 were asked individually for each AR nugget type. We list Q8–Q17 as they were asked for the show and tell AR nugget type (Q8.sat–Q17.sat). These Questions were asked similarly for the quiz AR nugget type (Q8.q–Q17.q) and semantic zoom AR nugget type (Q8.sz–Q17.sz). Finally, participants were asked if they can imagine further use cases from their daily lives where they would like to use the AR nugget authoring tool (Q18), experience with AR applications (Q19), experience in creating AR applications (Q20), age and gender.

(Q₁): Which device did you prefer for the following actions? (computer or AR device)

- 1) moving labels
- 2) rotating labels
- 3) scaling labels
- 4) move 3D models

- 5) rotate 3D models
- 6) scale 3D models
- 7) look at the preview

(Q₂): Switching my focus from computer to AR device was ... (complicated 1–7 straightforward)

(Q₃): Switching my focus from AR device to computer was ... (complicated 1–7 straightforward)

(Q₄): If I had only the computer but no AR device available, I would have needed ... time to create the AR nuggets (less 1–7 more)

(Q₅): Which additional edit functionalities would you like on the computer? (free text)

(Q₆): Which additional edit functionalities would you like on the AR device? (free text)

(Q₇): Placing and moving objects (3D models, labels) in the 3D room was ... (easy 1–7 complicated)

- 1) by pressing keys on the computer device

- 2) by clicking the buttons on the computer's UI

- 3) by tapping the buttons on the AR device's UI

- 4) by tapping and holding the object on the AR device

(Q_{8.sat}): How do you rate the workload for one show & tell AR nugget? (high 1–7 low)

(Q_{9.sat}): How satisfied are you with your final show & tell AR nuggets? (not satisfied 1–7 satisfied)

(Q_{10.sat}): How do you rate the workflow for show & tell AR nuggets? (complicated 1–7 straightforward)

(Q_{11.sat}): What contributed to a complicated workflow? (free text)

(Q_{12.sat}): What contributed to a straightforward workflow? (free text)

(Q_{13.sat}): When you tried the show & tell AR nugget with the default objects, did you get an impression of how your own AR nugget could look like? (No 1–7 Yes)

(Q_{14.sat}): Why were you able to get this impression? (free text)

(Q_{15.sat}): What did hinder you to get this impression? (free text)

(Q_{16.sat}): How important was it to be able to try out the show & tell AR nuggets directly on the AR device in the first step and at any time as it progressed? (not important 1–7 important)

(Q_{17.sat}): What did you especially like or dislike about the show & tell AR nuggets? (free text)

5.2 Results

We visualize the outcomes from the AttrakDiff questionnaire in **Figure 8**. The AttrakDiff portfolio presentation (**Figure 8B**) places the AR nugget authoring tool in the neutral area for hedonic as well as pragmatic qualities. The confidence intervals are similar small for hedonic and pragmatic qualities.

Figure 8A visualizes the mean scores for the AttrakDiff's word pairs. On the AttrakDiff's scale from 0 to 6, a value of 3 equals a neutral rating. Except for "unimaginative—creative," all word pairs are rated close to the neutral value of 3, in a range between 2 and 4. The word pair "impractical—practical" is rated slightly positively, while the other word pairs for pragmatic quality are rated slightly negatively. For the hedonic quality, the word pairs are rated closer to the neutral value of 3, but still slightly negatively, except for the creativity, which is rated the most positive from all word pairs.

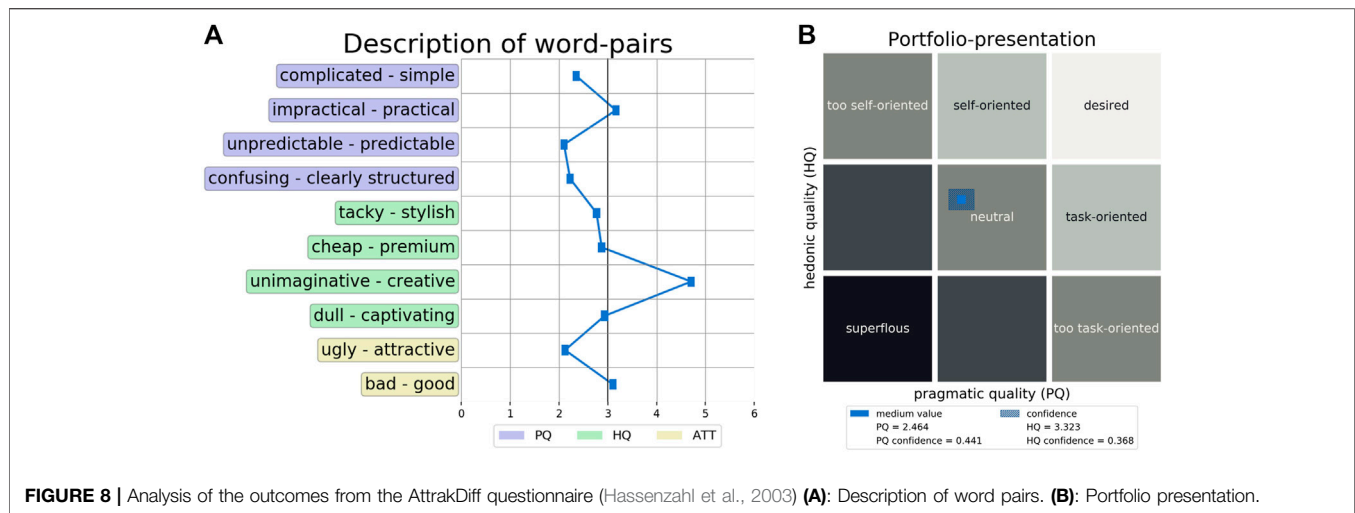


TABLE 1 | Outcomes from the questionnaire for questions that were asked for each AR nugget type (show & tell (.sat), quiz (.q), semantic zoom (.sz)) on a scale from 1 to 7.

Q	.sat Ø	.sat SD	.q Ø	.q SD	.sz Ø	.sz SD
Q8 (workload)	4.2083	1.6196	4.3958	1.7409	3.6042	1.5909
Q9 (satisfaction)	4.1667	1.6750	4.1458	1.8142	3.2708	1.7882
Q10 (workflow)	3.8958	1.5443	4.9583	1.3534	3.5417	1.5540
Q13 (first impression)	5.1042	1.4753	5.3542	1.4214	4.0208	1.9093
Q16 (try anytime)	4.8958	1.9283	4.2083	1.8815	4.8333	1.7717

From our users, 70.8333% (34 persons), preferred to preview AR nuggets using the AR device over the desktop device (Q1) g. In contrast, for the other actions (Q1) a to (Q1) f, only 4–8 users preferred the AR device, and most users (40–44) prefer using the desktop device.

Switching focus from desktop to the AR device was rated mostly straightforward (Q2, Ø5.1042, SD 2.1138) and similar straightforward for switching from AR device back to desktop (Q3, Ø5.4375, SD 1.7429). Anyway, our users think that without an AR device, creating the AR nuggets would have been neither slower nor faster (Q4, Ø4.1250, SD 1.3938).

Additionally to the provided authoring functionalities, users would like to have a 3D navigation control on the computer, similar to established 3D view and edit tools (Q5). They also mentioned some usability-related opportunities for improvement: displaying and scanning a QR code instead of the server ID, allowing drag and drop for 3D models, allowing scaling using the mouse wheel, selecting objects from a list instead of the 3D view. Five users would wish that the AR device supports touch input to edit the 3D models (Q6). The AR nugget authoring tool provides four different interactions to place and move virtual objects in the 3D environment. Here, clicking buttons was rated as most complicated (Q7c, Ø 5.8519, SD 1.3798, and Q7b Ø 5.4146, SD 1.6376). Using the keys on the computer’s keyboard was easier (Q7a, Ø 4.7391, SD 1.7623), and using touch input on the AR device was easiest (Q7d, Ø 4.5000, SD 2.1016). However, on our scale from 1 to 7 a neutral value equals four so that all interactions were rated rather complicated than easy. Half of our

users (24 persons) did not use the touch functionality and therefore were excluded from the calculation for (Q7d).

Table 1 shows the results of questions that were answered on a 7-point Likert scale for each AR nugget type. The workload was rated neutrally for the show & tell and quiz AR nugget and higher for the semantic zoom AR nugget (Q8). The example AR nuggets that we show in Figures 4, 6 were created by our participants in 11 min and the example quiz AR nugget shown in Figure 5 in 5 min (see Section 4). The users’ satisfaction with the AR nuggets was on a neutral level (Q9). For both, the semantic zoom AR nugget was rated lowest while show & tell and quiz AR nugget were rated with values close to each other.

Similarly, the workflow was rated neutral (Q10). Here, the quiz AR nugget’s workflow was rated significantly more positive than the workflow of show & tell or semantic zoom AR nuggets (Friedmann test with a *p*-value of 0.00005). For the show & tell AR nuggets, the interaction to connect labels with 3D models via two mouse clicks or by tapping was criticized (Q11.sat) by 13 users. Mostly for the show & tell AR nugget, but also for the other two types of AR nuggets, several users describe unexpected behavior (bugs) from the authoring tool (Q11.sat, q, sz). Three users had problems with the menu’s arrangement because in their case menus occluded other menu elements (Q11.q). For all nugget types, users explain that the order of the menus was contributing to a straightforward workflow (Q12.sat, q, sz). Furthermore, users explicitly state that starting with a default AR nugget instead of from scratch was helpful (Q12.sat, q, sz).

For the show & tell and quiz AR nugget, users were able to get a good impression of how their own AR nugget could look like (Q13). The users state that the default AR nugget is a good example to understand what this application type is about (Q14.sat, q). Users found it more difficult to get this impression for the semantic zoom AR nugget and rated it lowest for (Q13). This differs significantly from the other two AR nugget types (Q13.sat, q), (Friedmann test with a p -value of 0.02303). Users explain that the magnifying glass was scaled to their AR device in such a way that it took the whole AR device's screen (Q15.sz). Therefore, they were not able to see that semantic zoom objects were visible only through the magnifying glass. Our users rated it important to be able to experience the AR nuggets at any time (Q16).

About the show & tell AR nuggets, users liked especially the simplicity of the default AR nugget and the option to create an AR application with complex information from it (Q17.sat). They disliked the interaction to connect labels to the virtual object with two clicks (Q17.sat). Five users describe here that this interaction did not work for them at all. Furthermore, they express that they would like to add more than one 3D object to the AR nugget. The AR nugget's simplicity was also perceived positively for the quiz AR nugget (Q17.q). Furthermore, users liked the interactivity and the visualization of correct and wrong answers through blinking. Nine users disliked that the collider box for their 3D model was not set correctly, so that correct answers were detected as wrong. About the semantic zoom AR nugget, users liked the option to include multiple semantic zoom objects (Q17.sz). Additionally, they liked the idea of using a magnifying glass and pointed out that this can help to make a complex 3D model more clearly. The users disliked that the magnifying glass caused some usability issues. One user would like to combine this with labels as in the show & tell AR nugget. Some use cases where our users would like to use an AR nugget authoring tool in the future are tabletop and board games, education, navigation, guided tours, and care at home (Q18).

5.3 Discussion

Overall, our users rate the authoring tool with a small tendency to negatively and explain that with unclear user interactions and some unexpected behaviors (bugs). In contrast, the AR nugget approach and starting with a default AR nugget as an example application was perceived as helpful to get started with the AR application. The default objects within our default AR nuggets are simple objects like cubes and cylinders. However, these fairly simple objects are sufficient to give authors an idea of the AR application. The simplicity of the default AR nuggets allows authors to develop their own creative ideas and has no negative impact on the authoring tool's creativity (see **Figure 8A**). Finally, being able to try the AR nuggets at any time was rated important for all AR nugget types and suggests that the approach of starting with default objects and replacing them is supportive for our authors.

Although the AR nugget authoring tool provided the functionality to use touch input on the AR device for authoring activities, some authors listed this as a feature they would wish for (Q6). This indicates that touch input did either

not work for them or that they were not aware of it. These issues could also be one reason why most users used the desktop device for most authoring functionalities and only switched to the AR device to experience the AR nugget (Q1). Such usability issues have a negative impact on pragmatic qualities, which is also reflected by the outcomes from the AttrakDiff questionnaire (**Figure 8A**), where the pragmatic quality is rated between 2.1 and 3.2 points. Of course, one could provide a manual or tutorials to the authors that explain the authoring tool's functionalities. However, AR nuggets could also include smart assistant functionalities that can support authors pro-actively.

Furthermore, our authors had some issues that were related to scaling on different screen sizes (Q11). This has a negative impact on hedonic qualities but also to pragmatic qualities if users are unable to click or see occluded buttons. Scaling issues also negatively affected the experience with the semantic zoom AR nugget so that the semantic zoom AR nugget is rated slightly lower than the other two AR nugget types for (Q8–Q13). To improve this, AR nuggets as well as our AR nugget authoring tool should be further adapted for different hardware. The AR nuggets' runtime environment should be implemented and tested for multiple devices and screen resolutions.

Some authors in our user study had issues with the virtual models (Q11) because the models' colliders were not fitting the 3D model's meshes. Then, clicking or tapping the virtual objects is not detected as expected from users, making the authoring tool more unpredictable (see **Figure 8A**). Creating and preparing virtual 3D models is not addressed in our AR nugget authoring tool because content creation is another challenging task on its own. However, the AR nugget authoring tool could be extended with a functionality that checks if the 3D models are suitable when importing them. This can contribute to making the authoring tool more predictable.

Some of our users had ideas that they wanted to implement but that were not possible within the AR nugget's pattern. Others wanted to combine two patterns with each other. As our authoring tool was implemented exemplarily with three types of AR nuggets, the selection of patterns and AR nuggets was limited. Here, a wider range of various patterns should be implemented to support more than three types of AR nuggets. A greater selection can help to cope with a wider range of different application scenarios. However, complex application scenarios could be depicted by complex patterns, resulting in complex AR nugget types for these scenarios. While complex AR nuggets also make use of simple geometric objects as default objects, their complexity could still increase with increasing functionalities and complex patterns. Furthermore, providing a greater range of available AR nuggets could also make it challenging for authors to get an overview of available AR nuggets and to select the most suitable AR nugget. In this case, it could be useful to categorize AR nuggets to support authors in their selection process. Additionally, it can be supportive that authors can experience their AR nugget before making adaptations and at any time during the authoring process, especially for these scenarios. While adapting AR nuggets using the AR nugget authoring tool is suitable for persons without programming knowledge, programming knowledge is required to create the default AR nuggets. Therefore, it is up to

programming experts to extend the current state of the AR nugget authoring tool with additional AR nugget types. When no suitable AR nugget is available, a new suitable pattern must be identified and a new default AR nugget must be created and programmed. In this case, authors of AR applications would be reliant on programming experts and interdisciplinary communication might be required to fit the pattern and AR nugget to the author's need. However, once the new default AR nugget is created and included in the AR nugget authoring tool, authors would be able to re-use it in multiple application scenarios. Our AR nugget authoring tool is currently restricted to anchor virtual content on real-world images that are deposited as Vuforia image targets. However, it could also be useful to augment objects or surfaces. The AR nugget authoring tool could be extended with these functionalities by programming experts. Functionalities that aim to support authors can also be esteemed as an unwanted restriction. For example, we implemented that the text size for the show & tell labels was automatically adjusted to fit the labels' size. One user stated he would have liked to be able to adjust the text size on his own. One option could be to allow more experienced authors to disable these support functionalities.

6 CONCLUSION AND FUTURE WORK

This work introduced an authoring tool for pattern-based AR applications that we call AR nuggets. Similar to nuggets in microlearning or VR nuggets (Horst, 2021), AR nuggets are small self-contained units. Additionally, an AR nugget is a ready-to-use AR application that authors can customize to their own needs. Because AR nuggets are based on patterns, they can contribute to the reusability of AR applications and therefore to an efficient workflow.

In our user study, authors found the approach of AR nuggets helpful for the authoring process. The default AR nuggets use simple geometric default objects to give authors an impression of the AR application. Based on this, the authors can develop their own creative ideas for adapting the AR nugget. The order of the menus in the AR nugget authoring tool's UI was perceived as a helpful suggestion for a workflow. Using the AR nugget authoring tool, authors without programming knowledge were able to create AR applications. This work showed three exemplary AR applications that were created in this way in a relatively short time. Although authors in our user study described the AR nugget approach as supportive, they rated most aspects from the authoring tool in a neutral range and described that there was some unexpected behavior (bugs). The AR nugget authoring tool incorporates different degrees of immersion using a desktop computer along with an AR device. Our users authored mostly with the desktop computer but used the AR device to view and experience the AR nuggets. Here, being able to experience the AR

nuggets on an AR device at any time was helpful. Furthermore, this work proposed a toolchain that allows the integration of AR nuggets in another ready-built application. Although AR nuggets are stand-alone applications that include a runtime environment, it can be desired to load them in another ready-built application to contribute to standardized ways of updating applications. With our toolchain, single AR nuggets within an application can be updated or added without requiring a user to re-install the whole application.

In future work, the AR nugget authoring tool's usability can be improved based on the outcomes from the user study. Additionally, further default AR nuggets should be developed and implemented to provide a wider range of patterns to the authors. Here, one could incorporate different author roles: Authors with programming knowledge could create additional default AR nuggets based on ideas from those authors who customize AR nuggets to their use cases using the AR nugget authoring tool.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required for this study in accordance with the national legislation and the institutional requirements.

AUTHOR CONTRIBUTIONS

LR transferred the concept of VR nugget-based authoring to AR and developed the concept and workflow for the proposed authoring tool. Additionally, she realized and analyzed the user study. DD worked on the implementation of the authoring tool and added some conceptual ideas. RH and RD contributed to the conceptual ideas and the realization of the user study.

FUNDING

This work has been funded by the German Federal Ministry of Education and Research (BMBF), funding program Forschung an Fachhochschulen, contract number 13FH181PX8.

REFERENCES

Apaza-Yllachura, Y., Paz-Valderrama, A., and Corrales-Delgado, C. (2019). SimpleAR: Augmented Reality High-Level Content Design Framework

Using Visual Programming. In 2019 38th International Conference of the Chilean Computer Science Society (SCCC). IEEE, 1–7. doi:10.1109/SCCC49216.2019.8966427

Ashtari, N., Bunt, A., McGrenere, J., Nebeling, M., and Chilana, P. K. (2020). Creating Augmented and Virtual Reality Applications: Current

- Practices, Challenges, and Opportunities. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. New York, NY, USA: Association for Computing Machinery, 1–13. doi:10.1145/3313831.3376722
- Gattullo, M., Dammacco, L., Ruospo, F., Evangelista, A., Fiorentino, M., Schmitt, J., et al. (2020). Design Preferences on Industrial Augmented Reality: a Survey with Potential Technical Writers. In 2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE, 172–177. doi:10.1109/ISMAR-Adjunct51615.2020.00054
- Hampshire, A., Seichter, H., Grasset, R., and Billinghurst, M. (2006). “Augmented Reality Authoring”. In Proceedings of the 18th Australia conference on Computer-Human Interaction Design Activities, Artefacts and Environments. Editor T. Robertson (New York, NY: ACM), 409. doi:10.1145/1228175.1228259
- Hassenzahl, M., Burmester, M., and Koller, F. (2003). *AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität*. Wiesbaden: Vieweg, 187–196. doi:10.1007/978-3-322-80058-9_19
- Horst, R., and Dörner, R. (2019). Virtual Reality Forge: Pattern-Oriented Authoring of Virtual Reality Nuggets. In 25th ACM Symposium on Virtual Reality Software and Technology. 1–12. doi:10.1145/3359996.3364261
- Horst, R., Naraghi-Taghi-Off, R., Rau, L., and Dörner, R. (2020). Bite-sized Virtual Reality Learning Applications: A Pattern-Based Immersive Authoring Environment. *jucs* 26, 947–971. doi:10.3897/jucs.2020.051
- Horst, R. (2021). *Virtual Reality Nuggets-Authoring and Usage of Concise and Pattern-Based Educational Virtual Reality Experiences*. Ph.D. thesis. Wiesbaden: Hochschule RheinMain.
- Laviola, E., Gattullo, M., Manghisi, V. M., Fiorentino, M., and Uva, A. E. (2021). Minimal Ar: Visual Asset Optimization for the Authoring of Augmented Reality Work Instructions in Manufacturing. *Int. J. Adv. Manuf. Technol.* 119, 1769–1784. doi:10.1007/s00170-021-08449-6
- Ledermann, F., Barakonyi, I., and Schmalstieg, D. (2007). “Abstraction and Implementation Strategies for Augmented Reality Authoring,” in *Emerging Technologies of Augmented Reality*. Editors M. Haller, M. Billinghurst, and B. Thomas (Hershey: IGI Global), 138–159. doi:10.4018/978-1-59904-066-0.ch007
- Lee, G. A., Nelles, C., Billinghurst, M., and Kim, G. J. (2004). “Immersive Authoring of Tangible Augmented Reality Applications,” in *Third IEEE and ACM International Symposium on Mixed and Augmented Reality (IEEE)*, 172–181.
- Lytridis, C., Tsinakos, A., and Kazanidis, I. (2018). ARTutor-An Augmented Reality Platform for Interactive Distance Learning. *Edu. Sci.* 8, 6. doi:10.3390/educsci8010006
- Photon (2022). Bolt Overview — Photon Engine. Available at: <https://doc.photonengine.com/en-us/bolt/current/getting-started/overview>.
- PTC (2021). Vuforia enterprise Augmented Reality (Ar) Software. Available at: <https://www.ptc.com/en/products/vuforia>.
- Rau, L., Horst, R., Liu, Y., and Dorner, R. (2021). A Nugget-Based Concept for Creating Augmented Reality. In 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE, 212–217. doi:10.1109/ISMAR-Adjunct54149.2021.00051
- Roberto, R. A., Lima, J. P., Mota, R. C., and Teichrieb, V. (2016). “Authoring Tools for Augmented Reality: An Analysis and Classification of Content Design Tools,” in *Design, User Experience, and Usability: Technological Contexts*. Editor A. Marcus, 237–248. doi:10.1007/978-3-319-40406-6_22
- Speicher, M., Hall, B. D., Yu, A., Zhang, B., Zhang, H., Nebeling, J., et al. (2018). XD-AR. *Proc. ACM Hum.-Comput. Interact.* 2, 1–24. doi:10.1145/3229089
- Unity (2021). The Platform of Choice for Multiplayer Hits. Available at: <https://unity.com/>.
- Wang, M.-J., Tseng, C.-H., and Shen, C.-Y. (2010). “An Easy to Use Augmented Reality Authoring Tool for Use in Examination Purpose,” in *Human-Computer Interaction*. Editor P. Forbrig (Berlin, Heidelberg: IFIP International Federation for Information Processing), 285–288. doi:10.1007/978-3-642-15231-3_31
- Wilkes, C. (2021). Lean Touch. Available at: <https://carloswilkes.com/Documentation/LeanTouch>.
- Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.
- Copyright © 2022 Rau, Döring, Horst and Dörner. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.