MISSIONE 4
ISTRUZIONE
RICERCA

EUD4XR

WP 1 - END-USER DEVELOPMENT METHODOLOGY

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

LOGO
ente beneficiario

MISSIONE 4
ISTRUZIONE
RICERCA

Sommario

1. EUD process, roles and requirements

Recent advancements in technology, particularly in miniaturization, have contributed to the increasing prevalence of connected smart devices performing various functions across different domains, like home automation, healthcare, wellness, transportation and public safety. These devices are not only able to interact with each other by exchanging information but also to react autonomously to events in the physical world and influence it by performing processes or actions. In parallel, this surge in technology, in conjunction with investments by large companies, has renewed interest in XR applications and devices, e.g. Microsoft HoloLens, Oculus Rift or Meta Quest, from both consumers and researchers. Initially seen as tools for entertainment, these devices have matured, finding applications in various sectors, such as healthcare, aerospace, automotive, retail, and manufacturing. While these technologies have opened up new perspectives, such applications need to effectively address the needs of each potential end-user, or class of end-users, in order to be further confirmed and developed.

Depending on the context, managing the needs of users at the design level could be inefficient, and lead to overtime and cost. One possible solution, as happened for example in the publication of web content, is to actively involve end users, intended as domain experts without a programming background, in the development and editing process. By providing easy-to-use and usable tools, we can empower them to customize mixed-reality environments creating interactions that involve real and virtual objects.

Currently, both research and commercial products have proposed several End-User Development approaches that support end users in configuring XR applications [1], [2], [3], [4], in defining interactions between virtual objects in VR scenarios [5], [6], i. e. responding to the interaction with a user or other objects, or that leverage XR environments to help them in physical device automation [7], [8], [9]. The methodology proposed in this deliverable aims to enhance these solutions, taking the configuration of XR environments a step further by supporting the end user in customizing the behaviours of virtual and physical objects within them. Our solution will exploit a taxonomy for defining virtual objects and real devices having high-level actions for specifying event-condition-action rules readable as natural language sentences. Inspired by the meta-design approach [1], [2], [3], we will translate this functionality to XR environments, where:

- Professional developers, in our method called Element Builders (EB), create and deploy complex, shared XR environments via web repositories. Leveraging our taxonomy, developers can convert their implementations of virtual or real interactive objects into customizable elements within the XR environment, enabling the execution of high-level actions.
- End User Developers (EUDs), can customize pre-built XR elements provided by professional developers to create the final product. They define object behaviours within the environment using a rule-based language. To minimize any confusion when defining interactions, an intelligent conversational agent included in the authoring system will help the end-user.
- End-user consumers, which represent the final consumer of the created XR applications.

Finally, aiming to reduce the potential for user errors, or misunderstandings, our methodology will involve involves an intelligent conversational agent alongside the user.

1.1 EUD XR State of the Art

According to Lieberman et al. [4], End-User development (EUD) includes all methods, tools, and techniques aimed at lowering the knowledge for creating or modifying software systems, enabling users without programming skills to develop new applications or adapting existing ones to their needs. Involving end-users directly in the development process can accelerate the maintenance cycle, reducing costs and accelerating the process.

Similar efforts have also been made in the fields of VR, AR, and MR, where the barrier to entry for XR development is shrinking thanks to various commercial and research authoring tools. Generally, authoring tools can be divided into two categories:

- Desktop, where users create content on a standard computer screen.
- Immersive, which allows users to develop XR experiences while they are actually inside the virtual environment.

We find different commercial/open-source desktop authoring tools for XR environments. Hubs by Mozilla [5] is a multi-user virtual platform that allows people to interact with others in VR environments and create personalized virtual rooms. Although its main focus is on gathering people, users also have the option to customize the rooms they create by including multimedia elements or adjusting their appearance. However, the environment's behaviour remains a secondary aspect. An integrated editor 3D for Hubs, called Spoke[6], supports users on creating static representations of virtual scenes and composing 3D models. While this editor utilizes several mature techniques for end-user VR modelling in a desktop setting, it lacks robust support for defining the behaviour behind animation. Another example of a web-based application is Ottifox[7], designed to help users on creating VR/AR environments without writing code by using a set of rules, but its user target are VR designers who want to create rapid prototypes, and it works only with animation. Fungus[8] is an open-source tool for Unity, that supports novice developers to create interactive storytelling for both 2D and 3D games in Unity3D by using flowcharts. However, while the flowcharts approach helps users to define complex behaviours, it requires a huge amount of screen restraining end-user developers' understanding [9].

Focusing on the customization of more complex mechanics and behaviours within virtual environments, the literature proposes different tools aimed at providing the end-user with tools that enable him to customize interactions with virtual objects. Torres et al. [10] propose an editor where an end-user can add interactive content in the form of information panels, quizzes, pictures and 3D models. However, it is limited to using pre-capture 360° videos. Blecic et al. [11] propose a more powerful behaviour desktop authoring tool for creating desktop and immersive point-and-click games designed specifically for users without computer programming knowledge. It supports the user in defining the game behaviour, riddles and puzzles, by using event-condition-action (ECA) rules expressed in natural language, but also this work is limited to using pre-captured 360° photos or videos for the environment.

Generally, one limitation of desktop authoring tools, especially in the VR environment, is connected to the “build-test-fix” cycle, i.e., the need to compile the entire environment to test it and verify its operation and result. This iterative process can slow down the development process, creating an additional barrier for EUDevs and making it more difficult for them to create and modify their own applications. The ability to manipulate the environment directly within an immersive experience makes development more intuitive and streamlined, allowing developers to check interaction usability and user perception directly in the environment [12], [13].

The research community proposes different approaches for immersive authoring tools [12], [13], [14], [15], [16], [17] showing the potential of this paradigm for creating and changing the environment, its interactions, and behaviours. However, their targets are developers. RUIS[18] is an immersive authoring toolkit that lowers the barrier for creating and modifying VR environments using the metaphor of building blocks to facilitate and speed up the creation of VR experiences. While it supports a non-professional public, i.e. hobbyists and students, they require technical knowledge to assemble the building blocks through dedicated code. VR GREP[19] targets end-users, providing a tool where EUD can create VR environments. However, this tool does not support defining interactions or tasks, limiting it to navigation and reactions to click buttons. It is designed for novice developers, lowering the entry barrier but still requiring some technical understanding of 3D and programming concepts.

Industry and research proposed different approaches to define EUD systems able to define IoT home automations, e.g. IFTTT (https://ifttt.com/explore), EFESTO [60], TAREME [61], ImAtHome [62]. Such systems are built upon a visual paradigm which tend to be static, abstract and too large, and these issues can increase users' difficulties in defining the rules since such systems are detached from users' context.

Augmented reality can play a useful role by allowing users to direct interact with real objects, monitoring and exploring their state and dynamically discover/modify existing automations and create new ones that involve smart objects while moving around the home environment. In particular, AR-based solutions that exploit smartphones as interaction device can result especially convenient for users in that smartphones are devices that people already have, thereby there is no need to leverage on dedicated AR devices such as headsets or glasses. With particular reference to the home domain, previous work [58] indicated that users are eager to benefit from on-demand information, assistance, enhanced sensory perception, and play offered by AR across many locations at home.

One of the first contributions exploring the possibility of using AR not only to control but also to connect the behaviour of different IoT objects is Reality Editor [59]. Using AR, Reality Editor maps graphical elements directly on top of tangible interfaces associated with physical objects, such as push buttons or knobs. By connecting tags of different objects (by drawing a line between them), the user can program multi-object functionality. Thus, its goal is to provide additional possibilities for user interaction with the functionalities of available objects and devices, while our main goal is to support users in flexibly specifying automations involving multiple such objects. HoloFlows [56] is aimed at making simpler the configuration and modelling of IoT workflows through a no-code Augmented/Mixed Reality approach. It exploits concepts from the Business Process Modelling (BPM) domain and allows the definition of automated tasks involving one or more IoT devices. End users can use "virtual wires" to connect physical IoT devices and create processes involving them. It adopted the optical see-through approach to display holographic images on the glasses and to position them on the scene. The authors put forward an extended contribution [57] where the HoloFlows modelling approach is compared with a classical BPM approach (implemented in the

Camunda tool) and with a flow-based approach (NodeRED). HoloFlows requires dedicated hardware, and it is more oriented toward the "wiring" of objects physically nearby.

MagicHand [20] is an AR tool that allows end-users to control and monitor a restricted selection of IoT device behaviours using hand gestures. In particular, it first detects and localises available target devices from a depth camera mounted on the HoloLens; then, through a virtual control panel, it is able to detect and recognise hand gestures, and performs the corresponding action through a touchless interaction. However, a main limitation of this system is that it mainly aimed at controlling single devices.

Similar to MagicHand, HoloHome [21] targets non-developers with its AR framework for Hololens devices, facing similar limitations in terms of interaction complexity (i.e. controlling single devices). The main purpose of HoloHome is to support inhabitants to control real devices to complete their activities of daily living. However, the potentialities of HoloHome have been shown only through basic use cases in Ambient Assisted Living scenarios (e.g. users turning on/off a light, users looking for a specific object): thus, its possibilities to program the behaviour of various devices, sensors and appliances in a coordinated manner seems limited.

Instead, Stefanidi et al. propose a collaborative platform called BricklAyeR [22], an AR-based system that exploits a 3D building blocks metaphor to help non-technical users personalizing the behaviours of the mapped IoT devices. BricklAyeR allows users to define trigger-action rules through their tablets, defining how the available smart objects will respond to contextual events. Every rule is composed of triggers and actions represented as virtual blocks (i.e. 3D bricks) that can be connected like puzzle pieces: a user can direct the camera towards an object, and a 'digital twin' in the form of a 3D block will be created. However, it seems that this concept was not yet mature for testing with end-users, as only a cognitive walkthrough expert-based evaluation was conducted to gather feedback. The evolution of BricklAyeR has been MagiPlay [23], an augmented reality-based serious game for children, whose main goal is to provide young learners with an engaging way to program their surroundings. The provided user interface aims at enabling players to combine the 3D bricks as a part of the rule-based creation process. However, MagiPlay is a computational thinking learning game targeting children and mainly aiming at familiarizing them with problem-solving and computational thinking-related aspects: as such, it does not support the possibility to monitor and modify existing automations involving the objects encountered by mobile users.

The robot programming community explores also AR environment solutions to help end-users, for example Ikeda and Szafir propose PRogramAR[24], an AR trigger-action robot programming system designed for non-technical users.

1.2 Rule languages for End-User Development

One of the goals of the methodology we want to implement is to support the end-user in customizing environments and interactions with virtual and real objects they are working on. Over the past decade, the research community has proposed several programming styles for enabling these operations. These styles include rule languages and the literature explores different rule programming styles: while all allow to set at least one action, they differ in their ability to set conditions for when or under what circumstances those actions should be performed. The most adopted rule style is the trigger-action-programming (TAP), which has this pattern:

"*IF <a trigger occurs>, THEN <an action is executed>*"

Where:

- the trigger represents the event that fires the rule, it involves the state of an object or a change of context within the environment.
- the action describes the operation to execute, and can relate to a command to the object that fired the event or to another object in the environment.

For example, consider the rule *"WHEN someone enters the room, THEN turns on the light"*, entering the room will fire an event, the trigger, which results in the light being turned on, the action. This action may trigger another rule, leading to a chaining effect, and allowing to create more complex behaviours. TA rules find wide use in different fields, especially in IoT domains, both research systems [25], [22], [23], [24], [26], [27], [28], [29] and commercial platforms [30], [31]

However, while TA rules are intuitive and easy to use, they can lead to problems when the environment has many devices [32] and can be vulnerable to reasoning errors [33], [34].
Indeed, a potential source of ambiguity in this paradigm comes from the dual nature of a trigger, which can be either an event or a state. Events are instantaneous signals that indicate the occurrence of a change in the context, generated by an object, a device, by a user or by a change in the environment at a specific time, e.g., "*the temperature drops below 20°*", "*the user enters the room*", "*it starts raining*". Conversely, states are boolean conditions that can be true or false at any time, e.g., "*the temperature is below 20°*" or "*the user is inside the room*", *"it is raining"*.
Therefore, using states as triggers for a TA rule could cause inconsistencies or confusion, since the timing of rule execution remains unspecified.

Other works [35], [36], [37] explore a variant of the TA scheme, named if-then-else, that is obtained by including the else block to empower end-users to create more complex behaviours. In such a variant, the else block represents the action to perform when the trigger isn't raised, e.g., "*IF someone enters the room, THEN turns on the light ELSE turns off the light*".

Another variant of the TA rule, called event-condition-action (ECA), leverages both events and states to create a rule by enabling a third component, the condition, that stands between the trigger and the action: when an event occurs, the system checks the condition. If the condition is met, the associated action is performed. The condition can reflect the current state of an object or its environment. Compared with the pattern of a TA rule, ECA rules incorporate the adverb "WHEN" within their structure to differentiate between the trigger and the condition: "*WHEN <a trigger occurs> IF <condition> THEN <an action is executed>*", e.g, "*WHEN someone enters the room, IF the light is off, THEN turns on the light*". Although this paradigm can support defining multiple conditions, e.g. through boolean operators, many ECA languages for end-users limit the condition to only one predicate, reflecting the tendency of end-users to create simple rules with single properties checks [35].

ECA rules find use in several areas of end-user development. Barricelli et al. [38] apply them in a wellness domain to help coaches and trainers monitor and react to data coming from sport trackers to improve the quality of life and promote wellness awareness. In its interactive visual system, each rule represents a behaviour while the conditions involve the data received from the sensors. Manca et al. [39] leverage this paradigm in the ambient-assisted living domain, proposing a tool to support end-user TA rules debugging. In this tool, events and states are clearly distinguished using different keywords when editing the rules, *IF* and *IS* for the state and *WHEN* and *BECOMES* for the events. In Artizzu et al. [40], end-users uses ECA rules, expressed in natural language, to personalize the behaviour of specific objects. In the literature, this paradigm is leveraged also in home automation domain [32], [41], [42], [43], or on creating point-and-click games to personalize logic and behaviours [11].

While these systems differ in the application domain, the way the rules are created, or the authoring interface, they all share a common principle: states alone cannot trigger rules. The event remains the primary trigger, while states function as filters, determining when the action is executed based on the context following the event. One of the problems with using states as triggers for a rule relies on the timing of rule execution remains unspecified. Desolda et al.[44] proposes an approach based on 5W model aiming to give more context, especially temporal and spatial context, in the definition of the rule. In order to create a rule, the user is asked to answer these five questions: (1) WHICH service or device is involved, (2) WHAT indicates the rule's trigger and the actions to perform, (3) WHEN and (4) WHERE define, respectively, the time and location for triggering the rule and performing actions, (5) WHY is used to describe the rule behaviour to a human reader.

Actions can also be divided into different groups. Huang and Cakmak[34] differentiate actions in three ways:

- instantaneous actions, e.g. "*Send an alert*".
- actions that are extended in time and completed within a certain period of time, involving a change of state at the beginning and end of the action, e.g. "*Brew the coffee*".
- sustained actions, where the state reset may not occur, e.g. "Turn on *the light*".

For example, consider an automata (Figure 1). Instantaneous actions would be seen as an instantaneous transition from the current state to itself, e.g. $State\ 0\ - (sending\ the\ alert, 0) \rightarrow\ State\ 0$. Extended actions are not instantaneous, but they need a finite amount of time to be performed. In an automata, this action type represents a timed transition from the current state to another state where the system begins and ends in the same state after a finite time period, e.g. $State\ 0\ - (start\ brew\ the\ coffee,\ 0) \rightarrow State\ 1 - (brewing, t) \rightarrow State\ 0$. Finally, unlike extended actions, sustained actions, transition the system to another state without automatically resetting it to the initial state, e.g. $State\ 0\ - (turning\ the\ light\ on,\ 0) \rightarrow State\ 1$ and $State\ 1\ - (turning\ the\ light\ off,\ 0) \rightarrow State\ 0$.

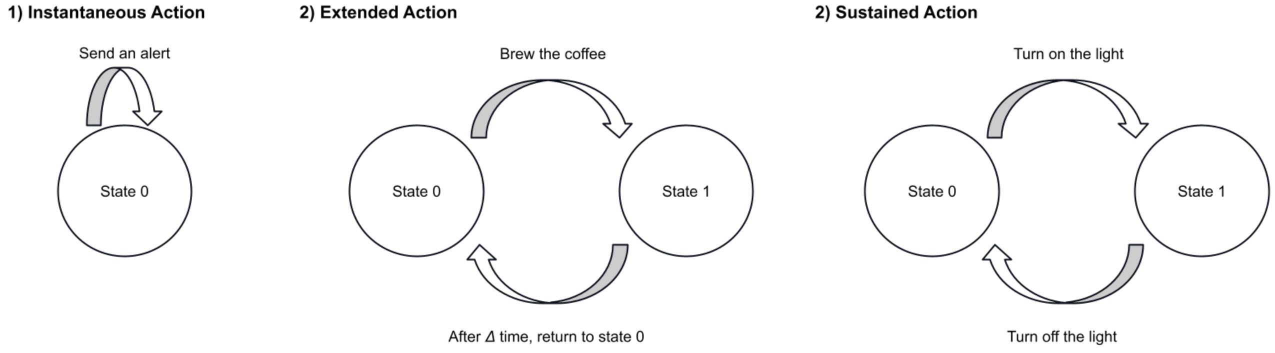


Figure 1 - Examples of different actions in a finite automata configuration.

In this methodology we will adopt the ECA rules format, expanding the methodology leveraged in [29], [41] for creating interactions with virtual objects or real devices and supporting.

In addition, aiming to expand ECARules4All to support the use of states as triggers, in this project, we aim to leverage the action categorization proposed by Huang and Cakmak [34] to enable the indirect use of environment, virtual or physical object states to create rules. For example, let us consider the action "Brew the coffee" of a coffee machine or the action "Turn on the light" of a lamp: this can be seen both as an action in a rule and as the event that produces a temporary state (which in turn can activate other rules). The rule engine of our project will need to provide developers, both end-user and professionals, with tools that allow them to recognize and exploit these categories of actions/events to use states as triggers.

2. Roles and Requirements

In the proposed methodology, we will use a meta-design model that actively involves end-users in the system development process, aiming to support them in customizing the XR environment and behaviours according to their needs. This model identifies three hierarchy levels:

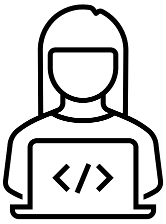
- The meta-design level. It represents the initial stage where the experts, e.g. software developers, engineers and graphic designers, create the system and its basic components.
- The design level. It is the level that involves the owners of the problem in hand, or the domain experts, in the development process: leveraging expert-built tools, they customize the system according to their needs.
- The use level. It focuses on the interaction between the consumer and the software.

Considering the actors involved in the various levels of this model, our method distinguishes three roles:

- **Element Builder**, or EB, this role typically belongs to professional developers and designers confident in creating and modelling objects and scenarios for XR applications, who create materials open to end-user configurations. Within the meta-design model, this role represents the experts in digital/software technologies.
- **End User Developer**, or EUDev, represents the domain experts without skills in programming, but with a moderate level of familiarity with XR environments and computer use. In our method, they need to create or modify XR content for professional or personal purposes but cannot build entire XR environments from scratch.
- **XR Consumer**, who is the final user of the XR contents, either created by EBs or by EUDevs. They represent the final user of the use level.

2.1 Personas

Based on the identified roles, we define 6 personas. Each persona describes a plausible user of our project, with specific demographic, technical skills, goals, and needs.



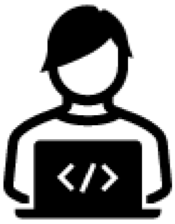
Francesca, the AR/VR professional developer (EB)

- **Name**: Francesca.
- **Age**: 31
- **Gender**: Female.
- **Profession**: AR app developer with 5 years of experience.
- **Location**: Milan area.
- **Education**: Degree in Computer Science or related field, with a Master in Human-Computer Interaction (HCI) or similar.

- **Experience**: Started as a freelance software developer, moved into AR development 5 years ago. She worked in several international projects, focusing on AR-based applications for supporting training and maintenance tasks in industry, AR applications for the real estate sector, and for the e-commerce.
- **Programming Skills**: Expert in AR software development tools (e.g., ARKit, ARCore, Unity 3D); she has also strong programming skills in C#, JavaScript, and Python and she is familiar with building 3D models. She has experience with UX/UI design, especially for AR environments, and also with user testing methodologies.
- **Modelling Skills:** Francesca is proficient in creating high-quality 3D models of objects, characters, and environments for use in her AR apps. Since AR overlays digital elements onto the real world, 3D models are the building blocks for these digital elements. As such, Francesca is well-aware of the importance of high-quality, optimized 3D models that should be detailed enough to look good, but also lightweight enough for smooth performance for real-time rendering on mobile devices, and which should also be able to realistically respond to user manipulation.
- **Familiarity with web repositories**: Francesca is highly familiar with web repositories and utilize them in several ways for AR app development. For instance, she uses them for 3D asset acquisition, as several online repositories offer high-quality, royalty-free 3D models that Francesca leverages in her AR apps. Using these web repositories can allow her to save time, allowing her to focus on customizing and integrating existing models rather than building everything from scratch. In addition, many web repositories offer access to AR development tools, plugins, and SDKs that can streamline the development process by providing pre-built functionalities, and also code snippets that Francesca can integrate into her AR apps (e.g. GitHub). Also, Francesca is familiar with online repositories hosting forums and communities dedicated to AR development, where Francesca actively participates in, to stay updated on the latest trends and share experience with other AR developers.
- **Goal**: One of the main goals of Francesca is to develop AR apps that are both innovative and user-friendly. To do that, Francesca is always eager to explore innovative and creative uses of AR technology to make its use increasingly widespread.
- **Needs**: She needs to have access to cutting-edge software and hardware that can support complex AR experiences. This includes e.g. development kits like ARKit and ARCore, as well as powerful computing platforms capable of handling intensive graphics and real-time data processing.

Caterina, the Unity game developer (EB)

- **Name**: Caterina.
- **Age**: 28.
- **Gender**: Female.
- **Profession**: Mid-level game developer.
- **Location**: Monza.
- **Education**: She holds a bachelor's degree in information and communication technologies from the Polytechnic University of Milan. Next, she spent two years at a master's in visual design and 3D illustration from the IT University of Copenhagen.
- **Experience**: She has 6 years of experience in video game development, working on both AR and 3D environments to design and implement gameplay mechanics and, often, 3D models.
- **Programming Skills**: Caterina is proficient in using Unity [45] and some of its main tools to develop video games. Initially, her primary focus was on 2D sceneries, but over the past few years, she has developed vast expertise in both MRTK1 and MRTK2 SDKs [46]. She capitalizes on these tools to design and realize immersive and interactive AR applications.
- **Modelling Skills:** Because of her educational background, Caterina is well versed in 3D modelling and animation, which she uses to create detailed 3D models, when necessary, for characters and objects in the projects she works on. In particular, she is familiar with the 3D modelling and animation Blender tool [47] and knows how to integrate models within Unity projects.
- **Familiarity with web repositories**: The company where Caterina works extensively uses version control systems, such as GitHub [48] and Bitbucket [37]. Now, she has a good understanding of the Git versioning tool. In the last past months, she has also begun uploading the 3D models she creates to practice on sharing platforms like Sketchfab [49].
- **Goal**: Caterina's primary object is using feedback and users' requests to increase her 3D modelling and design skills. She has started to accept small-maintenance works upon demands.
- **Needs**:
    - Reach a larger community interested in her work more efficiently.
    - Receive feedback from users regarding her shared materials.

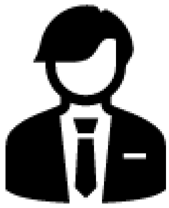


Frederick, the 3D artist (EB)

- **Name**: Frederick.
- **Age**: 39.
- **Gender**: Male.
- **Profession**: 3D modeller and VR digital artist.
- **Location**: Stuttgart.
- **Education**: He holds a three-year degree from the Academy of Fine Arts in Stuttgart in 2007. He attended different master's programs dedicated to 3D modelling and virtual reality development.
- **Experience**: Frederick has over a decade of experience in 3D modelling and digital art. He has collaborated as a remote freelance 3D modeler artist for various clients around the world, contributing to different productions. Recently, he has developed a strong interest in VR immersive experience development.
- **Programming Skills**: He has a solid base understanding of object-oriented programming principles. He created several immersive experiences, both VR and AR, using platforms such as Unity and Unreal Engine [50].
- **Modelling Skills:** He is well-known for his expertise in 3D modelling software, including Maya [51] and Cinema 4D [52], and his proficiency in texture painting software such as Substance Painter [53]. He has shown mastery of lighting and rendering techniques to achieve realistic and immersive visual results.
- **Familiarity with web repositories**: Having worked with clients from different parts of the world, Frederick has become very confident in using web repositories. He frequently sells 3D models and sceneries in web stores such as Sketchfab or Unity Asset Store [54].
- **Goals**: Frederick aims to expand his online sales of 3D models and templates, with the dual purpose of diversifying his income streams and spreading the maintenance over numerous consumers.
- **Needs**:
  - A wider audience to which to offer his creations.
  - Attracting the attention of potential new clients and collaborators.

Maurizio the lawyer, (EUDev)

- **Name**: Maurizio
- **Age**: 41
- **Gender**: Male
- **Location**: Rome
- **Education**: He has a degree in Law
- **Occupation**: Lawyer
- **Attitude to technology**: Maurizio is a fervent technology enthusiast, constantly exploring the latest advancements in tech, despite lacking formal training in areas like computer programming. Eager to embrace the potential of smart home technology, he has outfitted his residence with cutting-edge IoT sensors, devices and services, to make automatic some of his everyday routines, and to have more control on his house-related preferences. He discovered the AR technology from his daughter because she used to play Pokemon Go on her smartphone.
- **Tasks to complete**: He would like to define some home automations to e.g. increase security in his house, better manage the electric consumption in it, and ensure a healthy environment at home.
- **Needs**:
  - Get the data from sensors and check the status of devices installed within the house through the AR
  - Discover and modify the existing automations involving such sensors and devices
  - Create new automations
  - Test and debug: Maurizio wants to find out why some of the automations he defined are never executed. A tool that allows him to test the environment while he is editing the automations would help him find any errors.
  - Authoring support: Considering that Maurizio lacks programming skills, an intelligent conversational agent can help him create and update new automations and verify their behaviour.



Gianluca, the technical operator (EUDev)

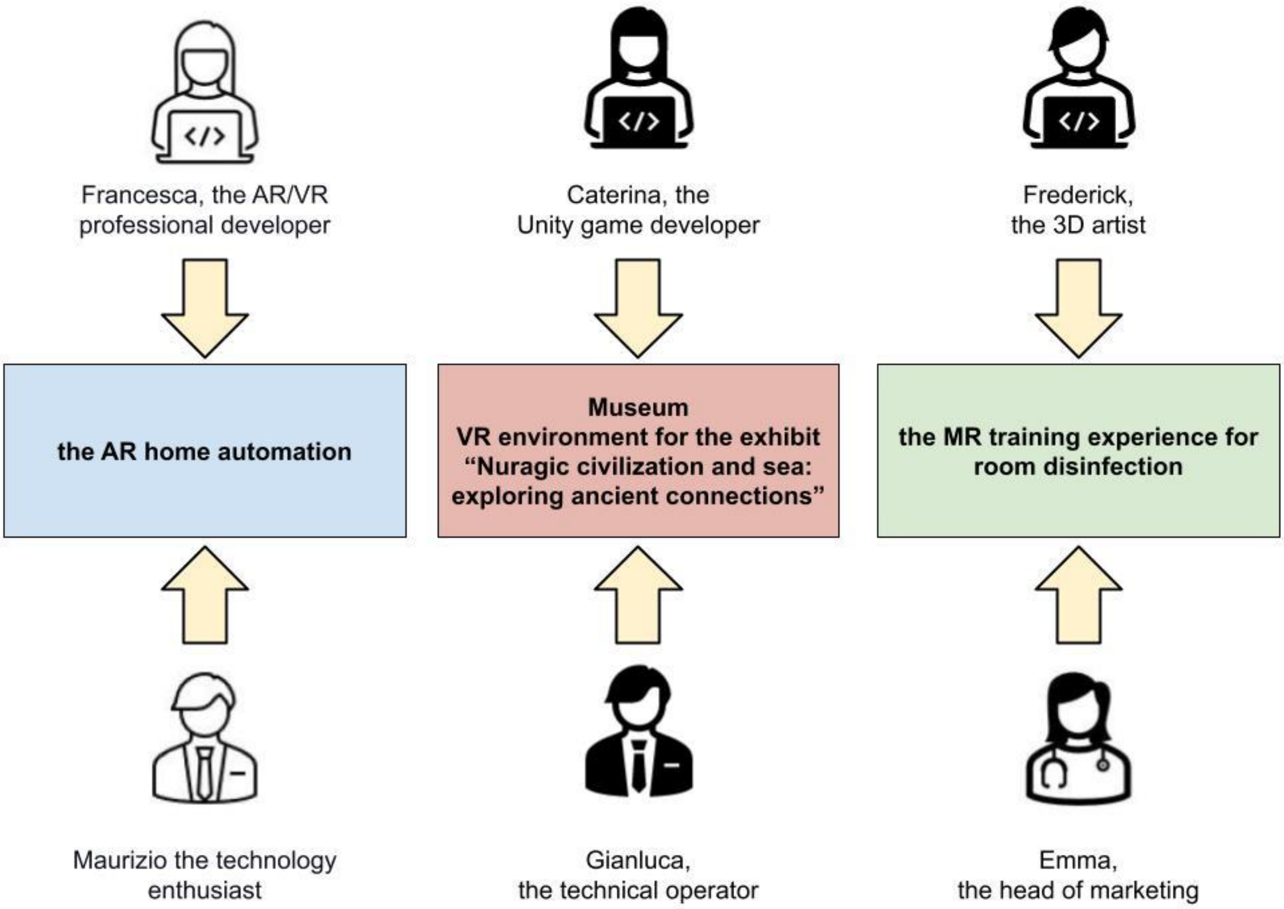
- **Name**: Gianluca.
- **Age**: 46.
- **Gender**: Male.
- **Location**: Cagliari.
- **Education**: He holds a bachelor's degree in electronic engineering. Two years ago, he attended a course on VR and AR, sponsored by the local government.
- **Occupation**: Technical operator at the engineering office of the National Museum of Archaeology in Cagliari.
- **Attitude to technology**: He has a general knowledge of standard office PC programs and is curious about new technologies. Some years ago, his daughter bought a Meta Quest 2 VR [55] headset, pushing him to explore this technology through games and immersive experiences.
- **Tasks to complete**: Recently, a VR tour has been installed in conjunction with a newly opened exhibition. Because this exhibition is relatively new, it is common for archaeological findings to arrive, requiring to update of the VR tour. Gianluca's supervisor assigned him the following tasks:
  - Update the VR tour's environment when a new archaeological finding arrives, including adding:
    - Their 3D representation in the VR environment.
    - Their detailed descriptive video within the guided tour.
    - The behaviours that will allow users to interact with their 3D representation (rotation, scaling and zooming).
  - Check the VR tour's operating status.
- **Needs**:
  - Update environment: to update the VR scenery that describes the exhibition, he needs a simple and intuitive tool where he can:
    - Add new items into the environment.
    - Add and customize behaviours within the environment.
    - Update existing rules.
  - Test and debug: Gianluca has to verify the changes he made to the environment. A real-time tool that allows him to test the environment while he is editing would help him find any errors.
  - Authoring support: Considering Gianluca lacks programming skills, an intelligent conversational agent can help Gianluca create and update new rules, and verify their behaviour.

Emma, the ward assistant (EUDev)

- **Name**: Emma.
- **Age**: 23.
- **Gender**: Female.
- **Location**: Vallensbæk.
- **Education**: She graduated from high school and completed a qualification course to become a ward assistant. Throughout her career, she has taken several training courses on the proper disposal of infectious waste, disinfection of machines and rooms, and hospital hygiene.
- **Occupation**: She is a ward assistant with specialized skills in disinfection at Rigshospitalet Hospital in Glostrup.
- **Attitude to technology**: She is passionate about technology and tech wearable gadgets. She is open to adopting new technologies that ease her work and provide her with opportunities for learning new skills.
- **Tasks to complete**: The Rigshospitalet Glostrup Hospital has organized an MR training course for the disinfection of rooms, machines and instruments. Considering Emma's background and experience in this field, she was assigned the responsibility for the course. Her tasks are:
  - To identify the learning objectives.
  - To keep updated the training course:
    - By changing the shown disinfection procedures, best practices and safety guidelines or adding new ones based on the latest available control procedures.
    - By synchronizing the MR environment content with the latest course updates. This could involve introducing new physical devices into the scenario or revising the lesson content, including descriptions and steps.
- **Needs**:
  - Update environment: she needs a tool where:
    - She can add or remove models from the VR environment.
    - Assign interactions to available clothing.
  - Test and debug: She would find very useful a tool that allows her to check the interactions assigned to each procedure shown in the course.
  - Authoring support: Due to the extensive number of procedures, an intelligent conversational agent could assist Emma in accurately associating interactions with the appropriate tools.

2.2 Scenarios and use cases

This section shows three scenarios describing how our solution supports users to meet their goals. Figure 1 summarizes the interactions among personas in our considered cases.



Scenario 1: the AR home automation

- **Context**: Maurizio is a lawyer who recently moved to a new house with his family. Being a technology enthusiast, he decided to equip his new home with the latest IoT sensors, devices and services (such as thermostat, air conditioner, mechanisms for opening windows/doors, etc.) for ensuring to his family a comfortable environment where to live. However, he is disappointed by the fact that often such devices can be managed only through the specific app developed by each manufacturing company, whereas he would like to flexibly set their behaviour in an orchestrated manner, and not using multiple apps. In addition, Maurizio would like to have better awareness and easy control of the automations that are currently active in the house and be able to analyse and modify them serendipitously and more directly and more direct manners smartphone while moving around the home.
- **Characters:**
  - **Francesca:** a developer, with several years of experience in developing AR/VR applications. Francesca agreed to help Maurizio in providing him with the possibility to customise the behaviour of the devices available in his home in a more intuitive manner, by also using some digital content augmenting the real devices available at home.
  - **Maurizio:** a technology enthusiast, eager to put in place some automations involving the devices available at home and aimed to better manage the life of his family in the new house, or modifying the existing ones from time to time according to current family needs with the general goal of facilitating daily routines, with special consideration to aspects such as security, electric consumption, and, more in general, comfort/wellbeing.
- **Story**:

  Maurizio recently benefitted of the Black Friday to buy a number of devices and objects with which outfit his new house. Thus, he contacted Francesca asking her to support him in having an intuitive way with which it is possible to create automations involving such devices, while moving within the house. In particular, Maurizio told Francesca that is tired of manually configure each smart object: he would like to have a system that allows him to define automatic actions involving various devices once some events occur in the environment regardless of the specific ecosystem to which each device belong. Also, the system should be flexible (as his family has needs that can vary over time) and intuitive to use, and it should help him to not only to know the current state of each device, but also to manage (e.g. to edit/add/remove). the automations that are currently active in the house and involving the smart objects.

  Francesca then needs to make available in the system a digital counterpart of the various devices that are in the home of Maurizio, with their main capabilities/functionalities, so that they can be managed by various automations. In particular, Francesca has to create suitable digital representation to use for visualising each device within the app, in such a way that its rendering would be effective when dynamically presented in an AR-based app (even using the small screen of a smartphone), as soon as the final user frames the object with the smartphone. The AR-based app should be also equipped with various functionalities for helping Maurizio in getting a better awareness and control/handling of the automations created and currently active in the various rooms of the house.

Maurizio, using the system proposed in our project, is able to configure the automations that are currently of most interest for him

- **Use case**:
  - Francesca is contacted by Maurizio who informs her that he moved in a new house with his family and therefore he wants to be able to create automations involving such devices, so that the family routines are facilitated.
    Francesca creates the digital models/representations of the various devices existing in the house of Maurizio, according to information provided by him. Once created, Francesca configures the AR-based app in such a way that the digital representation of the various devices will be available in it, and ready to be used by Maurizio for possible customisations involving them.
    For instance, one of such devices is a purifier: Maurizio recently discovered to be allergic to dust and decided to take the management of this condition seriously: indeed Maurizio placed it in the living room so it is possible to benefit from it when reading, studying, or watching TV.
  - By leveraging the taxonomy provided by our methodology, Francesca assigns to the model of the purifier device the category “Item” and the following behaviours “Remove dust particles”, “Detect current air quality”.
  - Then, Maurizio creates an automation that states that between h16 and h23 the purifier should be switched on. However, to avoid that the purifier is switched on also when there is nobody in the living room, Maurizio modifies the previous automation in such a way that the purifier is switched on only when there is someone in the living room during these hours.
    Also, Maurizio often spends the entire day outside, returning only when it's already dark. He noticed that when coming back late, the light outside the main entrance of his house is off, and this does not make Maurizio feel safe. Therefore, Maurizio would like to ensure that when returning late, the light illuminating that door is always on.
    Maurizio then wants to be sure that all such automations work in the right manner, thereby, using the system proposed in the project, he simulates the behaviour so defined and test it to be sure that everything work properly.
    When Maurizio is happy with the specification of the various automations, he save them and set them as ‘active’, so that they will start to actually work in the house.

Scenario 2: the Museum VR experience

- **Context**: An exhibition focused on the connections between the Nuragic civilization and other Mediterranean civilizations, titled "Nuragic Civilization and the Sea: Exploring Ancient Connections" is available at the National Museum of Archaeology in Cagliari. The exhibition's curator commissioned a VR environment to enhance the visitor experience through a virtual tour. In this virtual scenario, the visitor walks through a long room that contains all the objects within the exhibition. During the experience, the visitor can select one of the shown archaeologic finds and interact with it by rotating and zooming to view the item from different angles. The visitor can also access detailed educational materials available for that find, which provide information about the object's history, significance, and other relevant details.

- **Characters**:
  - **Caterina**: a Unity game developer and 3D modeller. She agreed to help the museum to create new VR content. Caterina's goal is to improve her 3D modelling skills by creating interactive and engaging content.
  - **Gianluca**: a technical operator at the National Museum of Archaeology in Cagliari, his main task is to update the VR environment according to the exhibition's contents. Gianluca is responsible for incorporating content created by Caterina into the VR environment, updating the behaviours within the scenario, and ensuring the system works properly.

- **Story**: One of the Egyptian figurines recently found in a Nuragic tomb was handed over this morning after its restoration work. The exhibition curator, Silvia, opted to include it immediately in the VR tour.

  Two weeks ago, Silvia contacted Caterina to request a virtual 3D model of the Egyptian statuette based on a set of photos. Using her skills in 3D modelling and animation, Caterina strives to make the model as accurate and detailed as possible, taking into account the dimensions, colours and details of the original.

  Gianluca needs to update the virtual tour to integrate Caterina's 3D model of the Egyptian figurine and define user interactions with the new object: selection, rotation, zoom, and show the educational video content related to the new artifact prepared by the media department.

  Gianluca leverages the methodology proposed in our project to integrate Caterina's 3D model of the statuette into the virtual tour. He configures the planned interaction available: rotation, zoom and playback of the educational video content, allowing visitors to explore the object in detail during the virtual experience.

- **Use case**:

- Caterina received an email from Silvia, the curator of an exhibition at the National Archaeological Museum in Cagliari, for which she has previously created 3D representations of some bronze statues and objects. In the email, Silvia asks if she can make a 3D model of a recently found Egyptian statuette that they want to include in the virtual exhibition. Similar to the previously created models, visitors should be able to interact with the 3D representation by selecting, rotating, and enlarging it.
- Caterina creates the 3D model of the Egyptian statuette using a 3D modelling software program such as Blender. Leveraging the taxonomy provided by our methodology, Caterina assigns to the model the category “Item” and the following behaviours “Selectable”, “Rotationable”, and “Scalable”.
- Caterina uploads the 3D model made to the remote repository, which she and the museum use to share the commissioned material.
- The exhibition curator informs Gianluca that the 3D model of the new find is ready to be included in the VR environment. Gianluca then downloads the model of find from the remote repository.
- Gianluca starts the 3D development environment, such as Unity. He then loads the VR environment in immersive authoring mode. Within the environment, he adds a new virtual display element, using the same 3D model previously employed, positioning the newly downloaded model on the display.
- Gianluca defines the interactions for the newly added figure. Using the rule language defined in our methodology, and with the assistance of the ICA, he specifies the following behaviours:
    1. When the user selects the archaeological find, and no other object is selected, the application brings the item into the foreground by zooming in.
    2. When a find turns out to be selected, the application starts an educational video content prepared by the museum's media department within a designated area of the interface.
    3. When the user applies a rotation, and if the figurine is selected, it rotates following the user’s movement.
    4. When the user applies a zoom, in or out, while the figure is selected, the system zooms on the virtual object accordingly.
- Gianluca, leveraging the immersive authoring tool, tests the defined behaviours. He verifies that the defined interactions, including selection focus, education content triggering, synchronized rotation, and zoom functionality, all operate as intended.
- Gianluca saves the changes and starts the application build.

Scenario 3: the MR training experience for room disinfection

- **Setting:** The Rigshospitalet Glostrup Hospital has organized a training course focused on the correct procedures to disinfect sanitary rooms, devices, and instruments. To improve participants' preparedness, the organizer commissioned an MR system that allows participants to safely test procedures.

- **Characters:**
  - **Frederick:** a Unity game developer and 3D modeller. She agreed to help the museum to create new VR content. Caterina's goal is to improve her 3D modelling skills by creating interactive and engaging content.
  - **Emma:** Emma is one of the most skilled Rigshospitalet ward assistants in such field. Her supervisor decided to assign her to lead this training course. Her tasks are:
    1. prepare the procedures to be shown within the lessons that make up the course;
    2. configure the MR environment to reflect the steps explained in the lessons.

- **Story:** The hospital executive asked Emma to add to the course a procedure for proper disposal of contaminated used syringes. This procedure requires a detailed sequence of steps to ensure the safety and effective disinfection.

  Emma begins to modify the XR environment to add the new procedure. Using our proposed methodology, Emma adds a step-by-step sequence of actions required to handle the contaminated syringes: putting on the appropriate gloves, requesting a special container, applying the correct solvent in the environment, and proceeding with the disposal. This ensures the procedure is simulated accurately and realistically in the XR environment.

  To complete the modification, Emma wants to add the 3D model that will represent the syringes to the XR environment. After searching a specialized website, she finds a suitable model, uploaded by Frederick. Once she obtains permission from her supervisor to purchase such a model, she integrates it into the XR environment, making sure that the model is positioned correctly.

- **Use case**:
  - Emma receives a request from the hospital executive to incorporate a new procedure into the training course for the proper disposal of contaminated used syringes. The procedure must be accurately simulated in the XR environment to ensure participants try the steps involved in safely handling contaminated materials.
  - Emma outlines the step-by-step sequence of actions required to handle contaminated syringes. She ensures that the procedure is detailed and aligns with the best practices and national safety guidelines.

- Emma opens a 3D development application to update the XR environment to accurately reflect each step of the disposal procedure. Leveraging on the rule language defined in our methodology, and with the assistance of the ICA, she defines the interactive elements and behaviours required for participants to navigate through the procedure. The XR environment drives participants to each step by showing, in the upper right corner of the interface, a text block that describes in detail the current step. This procedure consists of the following steps:
    1. The user wears the proper gloves.
    2. The user can apply the correct solvent to the affected surface if they wear the gloves.
    3. When the surface has dried up, the user can take the disposal container.
    4. The user can store the contaminated syringe inside the appropriate container.
- Emma searches for a suitable 3D model of a contaminated syringe to integrate into the XR environment. She explores a specialized website and finds a suitable model uploaded by Frederick, a 3D modeller and digital artist.
- After obtaining permission from her supervisor, Emma purchases the 3D model from Frederick and integrates it into the XR environment. She ensures that the model is positioned accurately within the environment to represent the contaminated syringe.
- Emma tests the XR environment, aiming to verify that the newly added procedure is accurately simulated and that participants can interact with the 3D model of the contaminated syringe as intended.
- Emma finalizes the modifications to the XR environment.

3. EUD objects and action representation

Our methodology aims to support EUDevs to customize XR experiences by configuring the behaviour of physical and virtual objects. To achieve this, EUDevs require access to a tool that lists and shows, in a clear and simple way, what actions are associated with virtual and real objects in the environment. Conversely, an EB needs a mechanism by which it can specify what an object can do in an environment. Therefore, such a tool must:

1. avoid technical and jargon concepts, allowing the EUDev can understand and manipulate the actions.
2. support different domains, real and virtual objects. In the first case, two instances of the same device predict the same actions, with virtual objects this may not be true, so the tool must be able to support both domains.

While a comprehensive dictionary mapping all high-level actions to every virtual or physical object within a template might be considered, this approach would be impractical. Our proposed solution leverages a taxonomy of reusable, high-level types for constructing XR templates. This taxonomy focuses on virtual and physical objects commonly used in XR environments, and aims to achieve a balance between flexibility and ease of extension through domain-specific additions.

Our taxonomy represents common XR objects through a predefined set of categories, and the EB then assigns a category to each virtual and real object that users can configure. The following subsections explain the taxonomy's organization, detailing how it categorizes objects usable within XR templates and the associated actions they can perform.

3.1 EUD objects

The taxonomy distinguishes between two different types of elements: objects and behaviours. The first describes the category that a user associates with entities based on their perceivable characteristics, e.g. their shape or the type of device to which it belongs. The second represents, on the other hand, the interactive behaviour that we can find in different objects, regardless of how they appear. Each virtual element or physical device is associated with a single category of objects and cannot have more than one. For example, an oven belongs to the Electrical Appliance category, and cannot be assigned to other categories, such as Food. Similar discourse for virtual objects, a model whose shape resembles a chair belongs only to the Furniture category and not, for instance, to the Electrical Appliance category. In our taxonomy, even elements representing 3D application attributes, e.g. cameras or lights, and media content, e.g. image or text, have their category in the Object type.

In this taxonomy, different objects may share the same behaviours. The Behaviour category describes typical behaviours that the EB can assign to different objects. For example, both an oven and a chair can be interactable, where interactable is a Behaviour. Even more, it is possible to assign more than one Behaviour to a virtual/real object, for example, the TV object can be used to both emit sound and play video, composing the set of possible actions the EUDev can program. Therefore, the type Behavior will include those elements/techniques to define environment manipulation and automation, e.g. collectable, counter, timer, container, sound, etc.

Aiming to make the proposed mechanism more flexible to the needs of EB and EUDev, the taxonomy supports domain-specific extensions to allow new categories or behaviours to be added, without requiring changes to the rules execution support.

3.2 Action representation

In the proposed methodology, an action represents an interaction occurring within the system. This interaction could be a change of state, for example “The light turns on”, or it may involve the execution of an operation, e.g. “The smartphone sends an email”, and so on. Notably, the performer of the action can be either the user, the virtual or real objects within the environment, or the environment itself.

In our approach, an action is expressed in natural language (English) and we support different schemas for action definition. The simplest pattern consists of two elements: the subject, identifying the actor who performs the interaction, and the verb, describing the effect of the interaction.

1. Subject / Verb
   “The TV / starts”

To specify more complex interactions, the scheme for defining an action supports a third component or multiple components. These additional components allow, for example, the inclusion of a second participant in the interaction, the assignment of a value, or the modification of a specific property.

2. Subject / Verb / Direct Object
   ”The user / presses / the button”
3. Subject / Verb / Value
   “The animal Nemo / walks to / Position value”
4. Subject / “changes” / property / “to” / value
   ”The furniture chair / changes / price / to / 20.00”
5. Subject / “increases/decreases” / property / “by” / value
   ”The furniture chair / increases / price / by / 20000.00”

Finally, the last schema handles passive actions. While it maintains the same structure from schema 2, Subject / Verb / Direct Object, the main actor becomes the type of the direct object, using the subject of the sentence as a parameter. This approach is particularly useful when aiming to restrict the number of properties definable within the acting object, cleaner and more adaptable code.

6. Subject / Verb / Direct Object (passive action)
   “The character human1 wears the armour”

In the rule language, an action can be used either to describe a rule’s trigger or to define the environment response when a rule is triggered.

4. Rule Language

In our methodology, we will extend the ECA approach, allowing end-users to customize XR interactions that involve virtual objects and physical devices using both events and states as rule's triggers. According to the literature, ECA rule systems limit states to be used only as conditions in the rule's triggers. In order to support both states as triggers for a rule, while avoiding common problems associated with state triggers, this method differentiates triggers in a similar way to the action subdivision showed by Huang and Cakmak [31] and applies different handling methods accordingly:

- Instantaneous actions that do not change the state of a system, e.g. "the user moves his hand", we will have a single rule where the trigger is the action.
- Actions that cause a temporary change of state, is an action that has a beginning and automatically returns to the initial state after a certain time, e.g. "the automatic gate is opening and will close". We will use two events: one for the next state and one for the revert.
- Prolonged action where the object's state does not automatically revert, for example, setting the air conditioner's temperature or turning the lights on/off. In such a case, the system may define one rule for each of the possible states.

Each rule will follow a precise pattern and can be expressed and read by the EUDevs in natural language (English) to ensure clarity in both their definition and context, minimizing the technical knowledge required to understand the rules and further lowering the barrier for users without programming experience. A generic rule has the following structure:

WHEN trigger IF condition(s) THEN action(s)

Where:

- The **WHEN** block describes the *trigger* of the rule. In our methodology, an event is expressed as an *action* within the environment, allowing EUDevs to define rules that react to any action performed by objects ~~or change of status~~ within the XR environment.
- The **IF** block, defines any conditions that must meet for the event to trigger the rule. Notably, this block is optionally: the EUDevs can specify zero, one or more conditions, providing them with a high degree of flexibility in setting rules.
- The **THEN** block denotes the *action* to be performed when the rule is triggered. Unlike the **IF** block, this field is mandatory. However, EUDevs can define multiple actions to be executed in response to the event.

Bibliography

[1] C. Ardito, P. Buono, M. F. Costabile, R. Lanzilotti, A. Piccinno, and L. Zhu, ‘On the transferability of a meta-design model supporting end-user development’, *Univ Access Inf Soc*, vol. 14, no. 2, pp. 169–186, Jun. 2015, doi: 10.1007/s10209-013-0339-7.
[2] M. F. Costabile, D. Fogli, P. Mussio, and A. Piccinno, ‘A meta-design approach to end-user development’, in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC’05)*, Sep. 2005, pp. 308–310. doi: 10.1109/VLHCC.2005.7.
[3] G. Fischer, E. Giaccardi, Y. Ye, A. G. Sutcliffe, and N. Mehandjiev, ‘Meta-design: a manifesto for end-user development’, *Commun. ACM*, vol. 47, no. 9, pp. 33–37, Sep. 2004, doi: 10.1145/1015864.1015884.
[4] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, ‘End-user development: An emerging paradigm’, in *End user development*, Springer, 2006, pp. 1–8.
[5] ‘Hubs by Mozilla’. [Online]. Available: https://hubs.mozilla.com/
[6] ‘Spoke’. [Online]. Available: https://hubs.mozilla.com/spoke/
[7] ‘Ottifox’. [Online]. Available: https://ottifox.com/
[8] ‘Fungus’. [Online]. Available: https://fungusgames.com/
[9] T. R. G. Green and M. Petre, ‘Usability analysis of visual programming environments: a “cognitive dimensions” framework’, *Journal of Visual Languages & Computing*, vol. 7, no. 2, pp. 131–174, 1996.
[10] A. Torres *et al.*, ‘A 360 video editor framework for interactive training’, in *2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH)*, IEEE, 2020, pp. 1–7.
[11] I. Blečić *et al.*, ‘First-person cinematographic videogames: Game model, authoring environment, and potential for creating affection for places’, *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 1–29, 2021.
[12] G. A. Lee, G. J. Kim, and M. Billinghurst, ‘Immersive authoring: What you experience is what you get (wyxiwyg)’, *Communications of the ACM*, vol. 48, no. 7, pp. 76–81, 2005.
[13] G. A. Lee and G. J. Kim, ‘Immersive authoring of Tangible Augmented Reality content: A user study’, *Journal of Visual Languages & Computing*, vol. 20, no. 2, pp. 61–79, 2009.
[14] A. Steed and M. Slater, ‘A dataflow representation for defining behaviours within virtual environments’, in *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, IEEE, 1996, pp. 163–167.
[15] T. Adão *et al.*, ‘A rapid prototyping tool to produce 360 video-based immersive experiences enhanced with virtual/multimedia elements’, *Procedia computer science*, vol. 138, pp. 441–453, 2018.
[16] F. M. De Farias, D. P. De Mattos, G. Ghinea, and D. C. Muchaluat-Saade, ‘Immersive Authoring of 360 Degree Interactive Applications’, *IEEE Access*, vol. 10, pp. 115205–115221, 2022.
[17] L. Zhang and S. Oney, ‘Flowmatic: An immersive authoring tool for creating interactive scenes in virtual reality’, in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 342–353.
[18] T. M. Takala, ‘RUIS: A toolkit for developing virtual reality applications with spatial interaction’, in *Proceedings of the 2nd ACM symposium on Spatial user interaction*, 2014, pp. 94–103.

[19] T. Zarraonandia, P. Díaz, A. Montero, and I. Aedo, 'Exploring the benefits of immersive end user development for virtual reality', in *Ubiquitous Computing and Ambient Intelligence: 10th International Conference, UCAmI 2016, San Bartolomé de Tirajana, Gran Canaria, Spain, November 29–December 2, 2016, Proceedings, Part I 10*, Springer, 2016, pp. 450–462.
[20] Y. Sun, A. Armengol-Urpi, S. N. R. Kantareddy, J. Siegel, and S. Sarma, 'Magichand: Interact with iot devices in augmented reality environment', in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 1738–1743.
[21] A. Mahroo, L. Greci, and M. Sacco, 'HoloHome: an augmented reality framework to manage the smart home', in *Augmented Reality, Virtual Reality, and Computer Graphics: 6th International Conference, AVR 2019, Santa Maria al Bagno, Italy, June 24–27, 2019, Proceedings, Part II 6*, Springer, 2019, pp. 137–145.
[22] E. Stefanidi, D. Arampatzis, A. Leonidis, and G. Papagiannakis, 'BricklAyeR: a platform for building rules for AmI environments in AR', in *Advances in Computer Graphics: 36th Computer Graphics International Conference, CGI 2019, Calgary, AB, Canada, June 17–20, 2019, Proceedings 36*, Springer, 2019, pp. 417–423.
[23] E. Stefanidi, D. Arampatzis, A. Leonidis, M. Korozi, M. Antona, and G. Papagiannakis, 'MagiPlay: an augmented reality serious game allowing children to program intelligent environments', *Transactions on Computational Science XXXVII: Special Issue on Computer Graphics*, pp. 144–169, 2020.
[24] B. Ikeda and D. Szafir, 'PRogramAR: Augmented Reality End-User Robot Programming', *ACM Transactions on Human-Robot Interaction*, 2024.
[25] N. Leonardi, M. Manca, F. Paternò, and C. Santoro, 'Trigger-action programming for personalising humanoid robot behaviour', in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.
[26] A. Bellucci, A. Vianello, Y. Florack, L. Micallef, and G. Jacucci, 'Augmenting objects at home through programmable sensor tokens: A design journey', *International Journal of Human-Computer Studies*, vol. 122, pp. 211–231, 2019.
[27] M. Andrao, F. Gini, A. Bucchiarone, A. Marconi, B. Treccani, and M. Zancanaro, 'Enhance Gamification Design Through End-User Development: a Proposal', 2022.
[28] A. Monge Roffarello and L. De Russis, 'Defining Trigger-Action Rules via Voice: A Novel Approach for End-User Development in the IoT', in *International Symposium on End User Development*, Springer, 2023, pp. 65–83.
[29] D. Fogli, M. Peroni, and C. Stefini, 'ImAtHome: Making trigger-action programming easy and fun', *Journal of Visual Languages & Computing*, vol. 42, pp. 60–75, Oct. 2017, doi: 10.1016/j.jvlc.2017.08.003.
[30] 'IFTTT'. [Online]. Available: https://ifttt.com/
[31] 'Zapier'. [Online]. Available: https://zapier.com/
[32] R. Ariano, M. Manca, F. Paternò, and C. Santoro, 'Smartphone-based augmented reality for end-user creation of home automations', *Behaviour & Information Technology*, vol. 42, no. 1, pp. 124–140, 2023.
[33] W. Brackenbury *et al.*, 'How users interpret bugs in trigger-action programming', in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.
[34] J. Huang and M. Cakmak, 'Supporting mental model accuracy in trigger-action programming', in *Proceedings of the 2015 acm international joint conference on pervasive and ubiquitous computing*, 2015, pp. 215–225.
[35] J. Coutaz and J. L. Crowley, 'A first-person experience with end-user development for smart homes', *IEEE Pervasive Computing*, vol. 15, no. 2, pp. 26–39, 2016.

[36] A. K. Dey and A. Newberger, 'Support for context-aware intelligibility and control', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 859–868.
[37] T. Kubitza and A. Schmidt, 'Towards a toolkit for the rapid creation of smart environments', in *End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings 5*, Springer, 2015, pp. 230–235.
[38] B. R. Barricelli, F. Cassano, D. Fogli, and A. Piccinno, 'End-user development, end-user programming and end-user software engineering: A systematic mapping study', *Journal of Systems and Software*, vol. 149, pp. 101–137, 2019.
[39] M. Manca, F. Paternò, C. Santoro, and L. Corcella, 'Supporting end-user debugging of trigger-action rules for IoT applications', *International Journal of Human-Computer Studies*, vol. 123, pp. 56–69, 2019.
[40] V. Artizzu *et al.*, 'Defining configurable virtual reality templates for end users', *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. EICS, pp. 1–35, 2022.
[41] G. Desolda, F. Greco, F. Guarnieri, N. Mariz, and M. Zancanaro, 'SENSATION: an authoring tool to support event–state paradigm in end-user development', in *Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part II 18*, Springer, 2021, pp. 373–382.
[42] 'Understanding Trigger-Action Programs Through Novel Visualizations of Program Differences | Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems'. Accessed: Mar. 12, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3411764.3445567
[43] L. De Russis and F. Corno, 'HomeRules: A Tangible End-User Programming Interface for Smart Homes', in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, Seoul Republic of Korea: ACM, Apr. 2015, pp. 2109–2114. doi: 10.1145/2702613.2732795.
[44] G. Desolda, C. Ardito, and M. Matera, 'Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools', *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 24, no. 2, pp. 1–52, 2017.
[45] 'Unity'. [Online]. Available: https://unity.com/
[46] 'MRTK'. [Online]. Available: https://aka.ms/mrtkdocs
[47] 'Blender'. [Online]. Available: https://www.blender.org/
[48] 'Github'. [Online]. Available: https://github.com/
[49] 'Sketchfab'. [Online]. Available: https://sketchfab.com/
[50] 'UnrealEngine'. [Online]. Available: https://www.unrealengine.com/en-US
[51] 'Maya'. [Online]. Available: https://www.autodesk.it/products/maya/overview
[52] 'Cinema4D'. [Online]. Available: https://www.maxon.net/en/cinema-4d
[53] 'Substance3DPainter'. [Online]. Available: https://www.adobe.com/products/substance3d-painter.html
[54] 'UnityAssetStore'. [Online]. Available: https://assetstore.unity.com/
[55] 'MetaQuest2'. [Online]. Available: https://www.meta.com/quest/products/quest-2/
[56] R. Seiger, M. Gohlke, U. Aßmann. 2019. Augmented Reality-Based Process Modelling for the Internet of Things with HoloFlows. BPMDS/EMMSAD@CAiSE 2019: 115-129 of Things
[57] R. Seiger, R. Kühn, M. Korzetz & U. Aßmann (2021), HoloFlows: modelling of processes for the Internet in mixed https://www.alexandria.unisg.ch/261944/
[58] P. Knierim, P. W. Woźniak, Y. Abdelrahman, and A. Schmidt. Exploring the potential of augmented reality in domestic environments, in: Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services,

MobileHCI ‘19, Association for Computing Machinery, New York, NY, USA, Article 31, 1–12, 2019. doi:10.1145/3338286.3340142

[59] V. Heun, J. Hobin, P. Maes. 2013. Reality Editor: Programming Smarter Objects. UbiComp (Adjunct Publication) 2013: 307-310J.

[60] Desolda, Giuseppe, Carmelo Ardito, and Maristella Matera. “Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools.” ACM Transactions on ComputerHuman Interaction (TOCHI) 24, no. 2 (2017): 1-52

[61] Ghiani, Giuseppe, Marco Manca, Fabio Paternò, and Carmen Santoro. “Personalization of context-dependent applications through trigger-action rules.” ACM Transactions on Computer-Human Interaction (TOCHI) 24, no. 2, (2017): 1-33

[62] Fogli, Daniela, Matteo Peroni, and Claudia Stefini. “ImAtHome: Making trigger-action programming easy and fun.” Journal of Visual Languages & Computing 42 (2017): 60-75