Evaluating the Use of Augmented Reality and 2D Interfaces for Diagnosing IoT-based Habitable Smart Spaces by Non-Expert Users

IOANNIS KANELLOPOULOS, University of Patras, Greece
IOULIA SIMOU, University of Patras, Greece
ANDREAS KOMNINOS, University of Patras, Greece

Smart space automation has, for decades, proposed a vision for enhancing the quality of human life. The Internet of Things (IoT) is one of the technologies addressing challenges and automating tasks that were once solely dependent on human intervention. However, what happens if an IoT system goes into a fault state? How can field-service technicians and even moderately tech-literate end-users without prior system knowledge diagnose and fix problems? In this paper, we investigate how different interface paradigms support diagnostic tasks in such scenarios, by presenting an Augmented Reality-based diagnostic tool and comparing it against a traditional 2D diagnostic desktop interface in a controlled user evaluation (n=25). We find that participants using the situated and embodied AR system complete tasks faster, task accuracy remains similar across interfaces and that the AR-based system is associated with higher physical demand. We interpret these findings as reflecting trade-offs between situated and embodied AR interfaces and symbolic 2D interfaces, and discuss implications for the design of diagnostic tools in smart space environments.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**; **Ubiquitous and mobile computing**; • **Hardware → Sensor devices and platforms**.

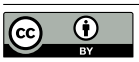
Additional Key Words and Phrases: Augmented Reality, Internet of Things, Sensors, Diagnostics

ACM Reference Format:
Ioannis Kanellopoulos, Ioulia Simou, and Andreas Komninos. 2026. Evaluating the Use of Augmented Reality and 2D Interfaces for Diagnosing IoT-based Habitable Smart Spaces by Non-Expert Users. *Proc. ACM Hum.-Comput. Interact.* 10, 4, Article EICS022 (June 2026), 28 pages. https://doi.org/10.1145/3816774

1 Introduction

Nowadays, technologies such as the Internet of Things and Augmented Reality are highly prevalent, and significant efforts are being made to integrate them into everyday life, both in workplaces and domestic environments, in order to facilitate and improve the quality of human life. Smart devices are increasingly integrated into the human environment, often in the form of sensors that serve both domestic, workplace and industrial automation [4]. In this paper, we are mostly concerned with the deployment of IoT based systems within an indoor domestic or office environment, which we will henceforth refer to as a *habitable smart space* (HBS). In practice, this corresponds to room- or building-scale IoT deployments such as smart homes, smart offices, or smart building spaces; we use the term HBS to emphasize that these environments are designed to be inhabited and maintained by end users and service personnel rather than IoT experts. Inside such HBSs, users need to configure

Authors' Contact Information: Ioannis Kanellopoulos, Computer Engineering & Informatics Department, University of Patras, Rio, Achaea, Greece, kanell@ceid.upatras.gr; Ioulia Simou, Computer Engineering & Informatics Department, University of Patras, Rio, Achaea, Greece, simo@ceid.upatras.gr; Andreas Komninos, Computer Engineering & Informatics Department, University of Patras, Rio, Achaea, Greece, akomninos@ceid.upatras.gr.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2573-0142/2026/6-ARTEICS022
https://doi.org/10.1145/3816774

and diagnose system errors and malfunctions, either during the initial deployment, or as part of routine operation and management of a configured space.

Habitable smart space automation must be configured and diagnosed by users that we define, for the purposes of this research, as *non-experts*, and reflect a target group of digitally literate home-owners or office workers, or field-service engineers who are called to configure or diagnose a system that is unfamiliar to them (e.g. a third-party contractor or service provider). As such, our research is aimed at two groups of potential users that may interact with HBS management systems for configuration or diagnosing purposes. The first group consists of end-users with moderate technical expertise who a) configure smart space systems themselves or b) need to diagnose an HBS system initially set-up by a third party contractor. The second group consists of field engineers with high technical expertise, tasked with a) configuring HBS systems for the first time or b) performing maintenance on such systems without prior knowledge of their initial configuration.

The majority of user interfaces for the management of HBSs are limited to basic monitoring device state and operating rules in a 2D environment, primarily accessible through desktop computer and mobile devices. These interfaces can often become very complex as the number of devices added to them increases, generating vast amounts of data [7]. Consequently, diagnosing and resolving potential error situations that may occur in one of the devices becomes challenging or even impossible for the type of user defined previously as non-expert [24].

While prior work on Augmented Reality interfaces for smart environments has largely focused on device control and rule authoring, our study investigates how an Augmented Reality hardware and software management system can support the process of fault diagnosis and troubleshooting, and examines how a situated and embodied AR visualisation approach compares to a traditional 2D desktop interface, which presents information in a more abstract and symbolic form. Our goal is to explore how their differing representational and interaction paradigms influence users' understanding of system behaviour during diagnostic tasks. To explore this concept, we present a scalable technical prototype for a configurable HBS system based on commercial off-the-shelf open source platforms. As an add-on, we present its integration with a wearable Head Mounted Display (HMD) Augmented Reality visualisation system. By enabling users to see the sensors in the physical environment, visualise their connections with other devices, monitor their current state, and the digitized data, the system provides a situated view of system behaviour that may support the processes of error diagnosis and system maintenance for the user.

Although AR interfaces for HBS management have been frequently examined in previous literature and have been heralded as a promising interaction paradigm [5, 28, 31], there is still limited empirical evidence comparing their effectiveness with traditional 2D desktop interfaces for fault diagnosis tasks. In this study we address this gap by examining how these two interface paradigms support diagnostic reasoning and introducing a visualisation approach to enhance the rules of HBS automations, by visualising data interconnections between devices participating in rules. Our findings demonstrate that users can be effective using both types of interface. While we observed shorter task completion times when using the situated and embodied AR system, this system was associated with increased physical demand, highlighting a trade-off between spatially situated representations and ergonomic considerations. These results suggest the need for more intuitive and natural interaction mechanisms.

2 Related Work

2.1 Non-expert experiences with HBS systems

2.1.1 Core paradigms and challenges. A recent and comprehensive review of the extensive extant literature in concepts, methods and tools in the context of smart spaces is offered in [4]. In this

section we outline the basic elements of this research area and refer the interested reader to the detailed coverage in [4]. The core challenge in HBS ecosystems, such as smart homes, is to support people in understanding, modifying, and creating automations themselves, even without programming knowledge [4, 47]. This need arises because, for automations to be effective, they must be under user control. The field of End-User Development (EUD) addresses this by enabling non-technical users to customise the behaviour of their devices and applications [4, 17, 24, 30, 47]. A widely adopted paradigm for EUD in smart ecosystems is Trigger-Action Programming (TAP), also known as Event-Condition-Action (ECA) rules [4, 7, 17, 24, 30, 47]. TAP is considered suitable because its compact structure links events and/or conditions with actions without requiring complex programming. Beyond basic TAP (one trigger, one action), methods have been proposed to extend the syntax to include multiple events and actions, as well as temporal and spatial constraints, for example adopting the 5W model (Which, What, When, Where, and Why) to guide users through rule definition [17]. Semantic properties can also help domain experts visually specify ECA rules based on their expertise using templates at three hierarchical levels: high-level intentions, medium-level realization plans, and low-level device scheduling logic (like TAP rules), aiming for automatic transformation between levels [4, 13].

Designing for shared HBSs with multiple users and varying preferences requires methods and tools that account for potential conflicts and support socially-oriented automations [4, 24, 47]. This includes providing operators and functionalities that consider user heterogeneity [4]. Additionally, automation rules can introduce security and privacy risks [4, 30]. Research explores user concerns, classifies threats, and develops methods, such as using textual descriptions, to explain why a rule might be dangerous [4, 29]. Context-awareness may help provide personalised security recommendations [4].

2.1.2 Programming and configuring HBSs. Various interfaces and interaction modalities are explored to enable TAP rule composition and modification. Visual-based paradigms are common, including form-based and data-flow approaches [4, 7, 17, 24]. Examples include commercial platforms like IFTTT, Zapier, and Atooma. While widespread, such platforms can offer limited flexibility compared to tools requiring programming skills like Node-RED [4]. Visual systems can also lose clarity when multiple rules are needed for complex tasks [7]. Inspired by block programming languages like Scratch, other approaches use visual blocks that users connect to build rules [14, 24, 47]. Studies compare the usability and difficulties encountered by programmers and non-programmers using such tools for smart home rule development [24]. While helpful for basic concepts, complex programming structures can still pose difficulties [24]. Hierarchical block solutions like CCBL are designed to mitigate coordination problems like redundancy found in traditional ECA rules [4]. Recommendations can also be incorporated into block-based environments [4, 24]. Recommender systems can assist users by suggesting relevant objects, services, or even complete automation rules [4, 15, 24, 47]. Strategies include leveraging rules created by other users or mining frequent patterns from the user's own behaviour (e.g., PBE systems) [4]. Recommendations can be presented as full rules or partial suggestions [4]. Combining these strategies is seen as a challenging future direction [4]. Technological advancements enable tangible interfaces integrated with pattern recognition and computer vision, conversational user interfaces, and Augmented Reality (AR) for creating home automations [4, 23]. Mixed Reality has also been explored for process modelling in IoT [4]. However, more studies comparing these approaches with traditional visual tools and assessing their long-term impact in realistic contexts are needed.

Beyond rule composition, tools and methods address user challenges like complexity, errors, and understanding. Users often make mistakes in composing rules, resulting in unsatisfactory or potentially dangerous system behaviour [4, 24, 37]. Tools are being investigated to help users

debug and understand errors by simulating rule execution and identifying potential issues like rule prevention, collision, or unexpected rule chains [4, 14, 15, 63]. However, more research is needed on how end users approach debugging and their strategies [4, 15].

2.1.3 Evaluation methods. Evaluating the effectiveness of these methods and tools is critical. While general usability methods apply, domain-specific methods are needed to capture aspects like errors users make [4]. Evaluation often involves qualitative methods like contextual inquiry and interviews, controlled experiments, and usability evaluations comparing different approaches [4, 7, 17, 24]. Longitudinal studies and "in the wild" trials are considered crucial for understanding long-term adoption and use by non-technical users in realistic environments [4, 24, 30, 47]. Common metrics include task completion, accuracy, time-on-task, and user satisfaction [4].

2.2 Augmented Reality in Smart Spaces

2.2.1 Extended Reality and the IoT. Augmented Reality (AR) is a key technology for supporting and enhancing the monitoring, diagnosis, and troubleshooting of general IoT systems [56], seamlessly integrating digital content with the physical environment, allowing the overlay of digital elements onto the real world. Its conceptual foundations trace back to early interactive and immersive systems (e.g. Sensorama [25], responsive environments [34]), while formalisation of AR/MR concepts and terminology emerged later [10, 43]. Subsequent developments led to the first integrated AR systems and functional classifications [19, 50], including marker-based, marker-less, projection-based, and superimposition-based AR.

AR technology is part of the broader concept of Extended Reality (XR), which includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), and is presented as a fundamental supporting technology in the era of Industry 4.0, aiming to transform traditional industries into smart ones [1, 9, 16]. Prior work shows IoT's impact on sustainability, efficiency, and safety [6], with numerous AR/IoT applications across domains. These include industrial monitoring, maintenance, and process control systems [33, 45, 54], real-time operational support (e.g. aviation procedures [55]), and Mixed Reality - Internet of Things (MR-IoT) solutions for infrastructure management [35]. Additional applications span medical guidance (e.g. surgical damage identification [48]), manufacturing (assembly support and production optimisation [46, 60]), and collaborative visualisation and safety systems [18, 32]. Emerging approaches also integrate AR with brain-computer interface (BCI) technology for assistive communication [51, 58].

Beyond the Industry 4.0 environment, the same technological advancements in AI, IoT, and XR are also driving changes in other domains, including smart homes and smart buildings [9, 39]. Smart Homes are defined by the widespread adoption of IoT technology, enabling household appliances to be internet-connected and managed remotely. The integration of XR, particularly AR and MR, offers new possibilities for interacting with and managing these complex environments [12, 20, 39]. For example, Seiger et al. [53] propose a novel approach for modeling IoT processes, allowing users to visualise processes and rules between sensors using mixed reality. Ariano et al. [5] introduced an AR system designed to offer a more engaging user experience when creating everyday automations involving sensors and devices. Purmaissur et al. [49] introduced an innovative, low-cost platform for controlling a smart home and monitoring energy consumption using Augmented Reality.

2.2.2 Core Technologies and Devices. The core technologies underpinning AR/MR monitoring of IoT systems in homes and smart buildings are IoT and XR itself [9, 12, 16, 20]. IoT provides the network of connected devices, sensors, and actuators, capable of monitoring, communicating, and transmitting data. XR technologies, specifically AR and MR, provide the interface for users to perceive and interact with this underlying IoT infrastructure [39, 40], though VR or pass-through VR may also be used [1, 11]. In extant literature, we find various devices are used to enable

AR/MR experience, but mostly AR applications involve the use of standard smartphones and AR glasses (or Head Mounted Displays-HMDs). Smartphones are widely used as they leverage existing camera capabilities to superimpose virtual information onto the real world. Mobile AR (MAR) applications on smartphones allow users to frame objects and see associated data or controls. Some example applications are found in [3, 5, 20, 22, 28, 41]. Head-Mounted Displays like Microsoft HoloLens and Vario XR-3 or other types of AR glasses are specifically designed for hands-free, more immersive AR/MR experiences and mentioned in literature as potential devices for identifying appliances based on location and providing controls in the user's line of sight. These devices can facilitate spatially registered information and interaction [2, 11, 12, 39, 40, 42, 64]. Prior work spans multiple application contexts, including Smart Homes, where the focus rests on domestic comfort, convenience, security, and energy efficiency. Examples include controlling lights, temperature, and entertainment systems, as well as managing energy consumption. Further applications include smart buildings and public infrastructure, which covers a range of contexts including nursing homes and smart campuses, and focus on leveraging IoT and AR/MR for operation and maintenance tasks.

2.2.3 Types of Information Visualised via AR/MR. Before discussing how information is presented in AR/MR systems, it is useful to consider the concept of *situated visualisation*. Situated visualisation refers to the embedding of data representation directly within the physical environment to which they relate. Presenting data in-situ enables users to interpret information in spatial context rather than through abstract mappings [62]. Prior work has explored such approaches in AR, demonstrating how spatially aligned visualisations can support interpretation and decision-making in context-aware settings [38, 57, 61]. In the context of IoT-based habitable smart spaces, AR and MR interfaces can support this form of situated visualisation by overlaying digital content directly onto the physical objects it relates to [3, 64]. The types of information visualised include:

(1) Real-time Sensor Data: AR/MR can display live data such as energy consumption (voltage, current, power factor, active power, e.g [3]), temperature, humidity, luminosity, or air quality (e.g. [44, 64]). This data is presented visually, often linked directly to the appliance or sensor reporting it.
(2) Device Status and Control Options: Users can see if a device is on or off and interact with virtual control panels or buttons superimposed over the physical object to change settings (e.g. [5, 12, 26, 44]).
(3) Automation Status and Configuration: AR/MR can show which automations are currently active or associated with a specific device or area. Users can explore, edit, or create new automation rules (e.g. trigger-action rules) directly within the AR view by interacting with the physical objects involved (e.g. [5, 41]). Crucially, while several papers in extant literature include visualisations of automation rules in AR/MR, these are limited to simply displaying textual or schematic representations of these rules. We found only one item in literature where visual depictions of the interconnections (data flows) between devices participating in a certain rule are demonstrated [11], however the effectiveness of this type of visualisation is not explicitly evaluated.
(4) Suggestions and Recommendations: Some advanced systems can provide recommendations for possible device configurations or automation rules based on the user's context or intent (e.g. [5, 64]).
(5) Instructions and Guidance: Particularly in contexts like Ambient Assisted Living or maintenance, AR can provide step-by-step visual or animated instructions overlaid onto the real environment to guide users through tasks (e.g. [39]).
(6) Identification and Metadata: AR can help users quickly identify devices, access product information, technical schemes, or maintenance manuals (e.g. [3, 9, 26, 31]).

(7) Alerts and Warnings: Important notifications like doorbell rings, alarms, or safety alerts can be visualised on wearable AR devices (e.g. [42]).

2.3 Summary

In summary, literature sources highlight that enabling end-user control of automations in smart ecosystems primarily revolves around developing usable EUD methods and tools based on the TAP paradigm, addressing rule complexity, composition interfaces, and support for debugging, understanding, recommendations, and social, security, and privacy concerns. Further real-world evaluation is needed to improve these solutions for diverse users [4]. Within this paradigm for HBS configuration and management, the situated and context-aware visualisation is presented as a key advantage of using AR/MR over traditional abstract interfaces [5, 28, 31]. Several arguments are put forward by extant empirical studies towards this end. Visualising system status and automation rules through AR/MR offers significant advantages for understanding system behaviour and diagnosing issues. Compared to traditional 2D interfaces, which rely on abstract representations that can hinder users' ability to map digital elements to physical devices and increase cognitive effort, AR/MR links information directly to real-world objects, improving system transparency, contextual awareness, and understanding of system behaviour. This supports tasks such as identifying device states, diagnosing issues, and interpreting IoT data, while reducing information overload through context-specific visualisation.

The process of creating and debugging rules can be abstract and challenging for non-experts in traditional interfaces. Traditional editors may present large sets of triggers and actions, requiring high cognitive effort to identify relevant elements, further exacerbated by the growing number of IoT devices. While AR can simplify creation, visualising and debugging complex rule interactions across multiple devices remain challenging and are an area of ongoing research. Direct comparisons between AR/MR and 2D interfaces in ecologically valid HBS contexts are scarce, with very few studies available. In fact, only two studies were identified. In [21], user performance across AR (gaze, gesture), voice, and mobile interfaces was evaluated, but focused only on device control, finding the 2D mobile interface preferable to AR. In [65], researchers compared energy consumption visualisations on a tablet versus AR, but the study was qualitative and focused on understanding and state changes rather than troubleshooting. The results suggested that 2D interfaces better support cognitive tasks, while AR may better support physical interaction.

Hence, the main research objective for this study can be positioned within the field of enquiry that explores AR/MR as a way to make HBS management more intuitive and accessible for non-expert users by providing spatially relevant information and direct interaction, thereby addressing some of the challenges posed by traditional 2D interfaces and the increasing complexity of IoT ecosystems. Specifically, we investigate how the situated and embodied visual representation of state and logic in 3D AR space affects diagnostic performance, compared to informationally equivalent 2D diagnostic desktop interfaces. To address the identified gap in literature, we present the design and development of a comprehensive HBS management system, and conduct an empirical evaluation of a situated and embodied AR interface and 2D diagnostic desktop interface in the domain of understanding and debugging problems arising from device faults, or automation rules by non-expert users.

3 System Design and Architecture

To investigate our main research question, we developed a complete HBS system using the technologies described and configured its devices and operating rules. As the focus is on users' cognitive burden in understanding system behaviour and diagnosing faults or abnormal conditions, the initial configuration was performed by the researchers.

3.1 Materials and Tools

This section provides an overview of the tools used for the development of the proposed application. This section also explains how the system was configured for the purposes of the usability experiment described in Section 5.

The system was built around the open-source ThingsBoard IoT platform, that enables the development, management, and scaling of IoT solutions. ThingsBoard was used as the core of our HBS management system to model devices and operating rules, and to provide the data layer supporting both the 2D (desktop) and AR-based interfaces. It was deployed alongside PostgreSQL for internal data storage and Apache Kafka for managing and processing data streams.

The AR experience was delivered using the NREAL Light head-mounted display, which is accompanied by a computing unit for running AR applications and a handheld controller for user interaction. The AR application was developed in Unity (version 2021.3.8f1), a cross-platform engine supporting C# scripting and integration with AR toolkits. Visual Studio was used for implementing and debugging the application code.

To enable AR functionality, we used the NRSDK toolkit provided by NREAL, which offers features such as *Spatial Computing* which includes motion tracking, image recognition, user hand position tracking, etc., *Optimized Rendering*, and *Multimodal Interactions*. In addition, the Immersal SDK was used for spatial mapping of the environment. This involved capturing images of the space to generate a point cloud representation, which was used to construct a 3D model of the experimental environment. This allowed sensors and devices to be positioned within a shared coordinate system, stored in ThingsBoard, and visualised in the AR-based interface.

Finally, the system was deployed using Docker, which enables the creation and management of containerized applications, ensuring consistent execution across environments.

3.2 System Architecture

The implementation of our HBS system, consists of multiple software components that together form the architecture of the proposed solution, as shown in Figure 1. The main components of the system are outlined next.

IoT Devices. Physical devices (e.g. sensors) that connect to the network and can generate and transmit data. For this study, we did not employ actual physical devices, in order to avoid potential hardware and software protocol incompatibilities, and also to simplify development by avoiding the need for software bridges to translate device generated data into a uniform structure to be supported by the IoT platform. Instead, devices were simulated via Python scripts generating and transmitting data, though the system supports integration of real devices in future.

IoT Platform. In order to aggregate data from various IoT devices and enable the management of the HBS system, an IoT platform is used to register and integrate devices, as well as for communication and data exchange between devices and other software components. We selected ThingsBoard as the IoT platform, since it provides all the necessary software modules that are required in order to deploy and manage a complete HBS system:

- User Accounts: ThingsBoard allows for several account types that allow access to various UI elements and API calls, depending on desired access levels. We use two types of account, one that has management access over the devices and rules, and one that is intended for end-users (experiment participants) that allows access over data visualisation dashboards only.
- Transport Layer: This layer enables secure and efficient data transfer between the IoT devices and the platform.

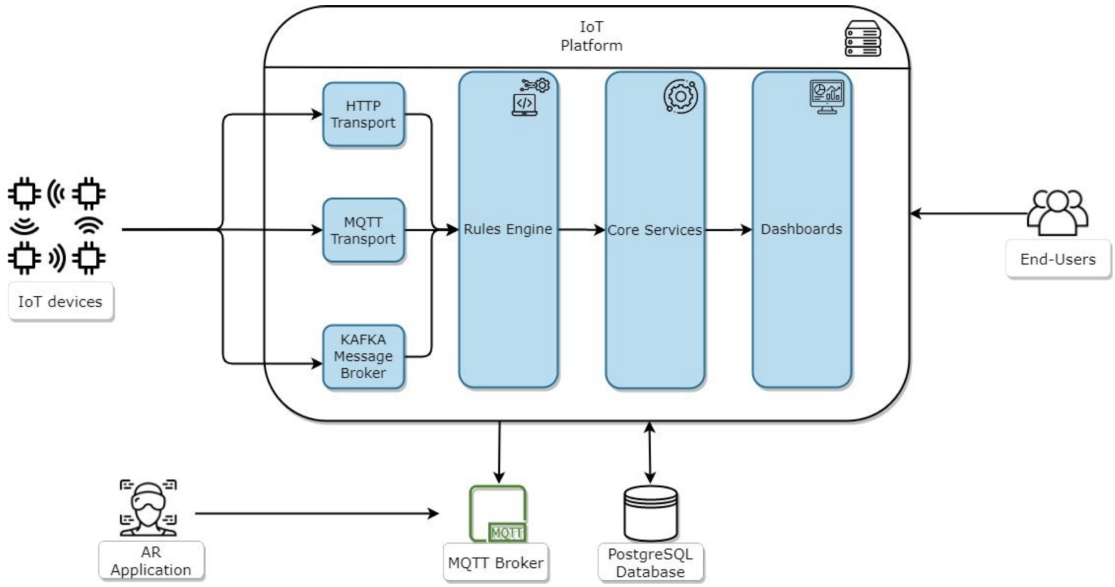


Fig. 1. High-level overview of our Habitable Smart Space system Architecture

- Rules Engine: This component supports the business logic layer (BLL) by managing data flow and processing. The IoT platform provides a graphical interface for creating workflows and triggering actions based on rules and conditions defined using JavaScript.
- Core Services: These services belong to the data-access layer and handle communication with the database, user authentication and authorization, and the registration of new devices and their identifiers.
- Management UI: ThingsBoard provides a complete web-based UI for the management of devices and rules within the platform. We leverage this functionality for the initial configuration of the HBS. This management UI is restricted to system administrators and is not meant for use by system end-users.
- Dashboards: ThingsBoard provides a method for creating end-user interfaces for visualising data, in the form of dashboards that incorporate graphical widgets. We describe in Section 5.1 how we leveraged this functionality to develop a 2D (desktop) management application for end-users, as part of our usability study.

Additional system components include the *PostgreSQL database*, *Kafka broker*, *MQTT broker*, and the *AR application*. The *PostgreSQL database* supports efficient storage and management of large data volumes, with the IoT platform's data-access layer handling storage and retrieval via queries. The *Kafka broker* (via ThingsBoard integration) enables device-platform communication and supports potential pre-processing of data streams by external services (e.g. Apache Spark), which extends beyond the platform's limited rule-chain processing (e.g. no advanced aggregation or machine learning). We leave such applications as a subject of future work as they are not in scope for this research. The *MQTT (Message Queuing Telemetry Transport) broker* facilitates message exchange between clients, enabling communication between the IoT platform and the AR application. Finally, the *AR application* provides a 3D visualisation of the IoT system for fault and anomaly diagnosis. We

describe in Section 4 how the AR management application using a wearable headset was developed for end-users, as part of our usability study.

3.3 Configuration of the IoT platform

Core system components include functions related to device registration, data transfer and management between the devices and the IoT platform, rule chains for processing, and data transfer from the platform to the AR application.

3.3.1 Device Registration and Integration into the System. Device registration on the IoT platform follows a structured and secure approach. Initially, the system requests specific characteristics of the devices (e.g. name, description). Each device is assigned to a profile, which serves as a template for each device and defines the characteristics and behavior of a device category. Additionally, each device has some supplementary characteristics, which are either defined by the user or automatically set and updated by the ThingsBoard platform. For our usability experiment, five devices were registered in the IoT platform and connected to distinct profiles. These were an indoor temperature sensor, an outdoor temperature sensor, a humidity sensor, an air conditioning unit and a smart windows device. As explained in Section 4.2.7, device registration can take place by leveraging the ThingsBoard RESTful (Representational State Transfer) API through the AR application, allowing the user to indicate the 3D coordinates of the device by simple visual positioning of the device, or via the ThingsBoard desktop UI.

3.3.2 Rule Chains. Rule chains implement the business logic of the system. There are connections between the chains that support the data flow necessary to execute this logic. For our usability experiment, seven rule chains were developed (*Smart Windows*, *Thermostat_Outdoor*, *Task1 (a)*, *Task1 (b)*, *Aircondition_DT*, *Humidity_DT*, *Thermostat_DT*), each linked to a specific device profile, responsible for data processing, validation checks, and communication with the AR application. The execution priority of these rule chains in real time, their interconnections, and the specific logic they implement are analyzed below, with the full configuration provided in the supplementary materials.

(1) The data flow begins with the *Task1 (a)* rule chain, connected to all sensor devices, which receives sensor data, detects inactivity, and forwards data to the *Root Rule Chain*. Within this chain, seven nodes manage data reception and determine the appropriate actions based on a probable detection of inactivity. Once identifying and addressing a potential inactivity state, the chain updates the system regarding the resolution of the situation.
(2) The data flow then proceeds to the *Root Rule Chain*, where the sensor telemetry data is simply forwarded to the *Main Chain* (Fig. 2).
(3) In the *Main Chain* (Fig. 2), telemetry data is parsed and distributed to sensor-specific chains and to the *originator*, which retains all relevant information and digitally represents the physical environment.
(4) The *Sensor's Rule Chain* processes sensor data. It receives the value transmitted by a sensor and updates the system status based on whether the value is within the allowable limits or if a device is in a critical state.
(5) The rule chain responsible for system devices that receive values from sensors manages the activation or deactivation of each device based on those values and the conditions they meet. The corresponding information is then transmitted to the augmented reality application.

3.3.3 Scanning the Physical Environment. Spatial information from the physical environment is recorded using the Immersal Mapper application, that generates a point cloud representation of the space, which is a collection of data points within a three-dimensional coordinate system describing

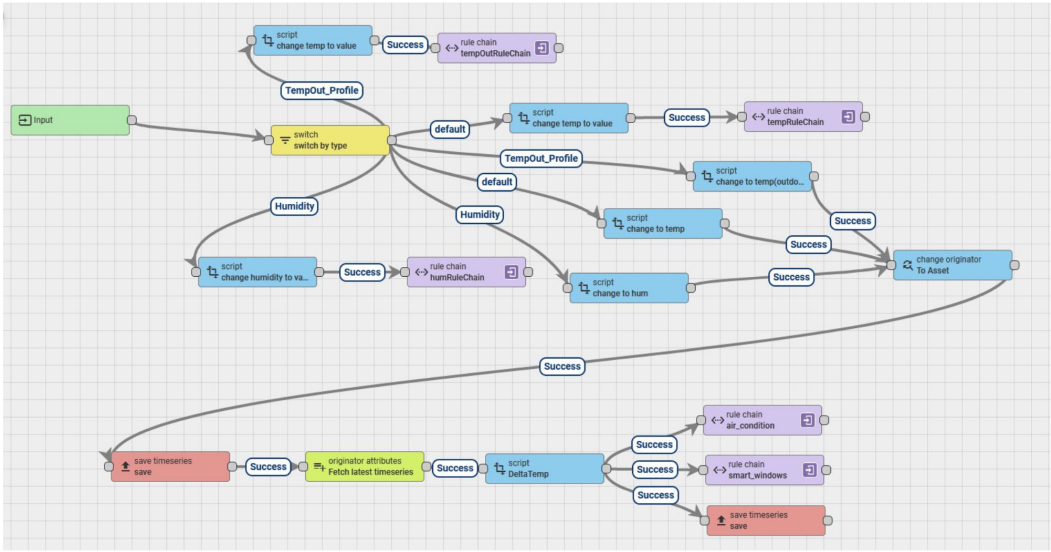


Fig. 2. Main Rule Chain

the geometry and spatial characteristics of objects in the environment. This application employs advanced computer vision techniques to analyze visual inputs and extract key information about the structure and layout of the scene. As explained in Section 5, the Immersal SDK was used to create a 3D model of the experiment environment, and to position the sensors and devices in the created 3D space, storing these coordinates in the ThingsBoard device representations.

4 The AR Application

We developed an AR-based application that augments the physical world with digital information, enabling users to gain insights and make decisions in real-time. The AR-based application initially connects in real-time with the IoT platform, and its design has been made in a dynamic way, allowing for easy extension of its capabilities and ensuring it can be implemented as ”plug and play” solution in any IoT system.

4.1 Communication and Connectivity

Communication between the AR application and the IoT platform uses REST (HTTP) and MQTT protocols for bidirectional data exchange. The application interacts via API requests, while MQTT enables real-time updates. This ensures bidirectional communication, enabling the application to receive real-time updates on sensor values, critical conditions, and system notifications. The communication architecture between the AR application and the IoT platform is illustrated in Figure 3.

Furthermore, an MQTT listener script continuously monitors broker messages and routes data to business logic functions. Data is exchanged in JSON format, enabling efficient and flexible communication between components.

4.2 Application Features

The AR application offers a variety of key features that enhance diagnostic capabilities and the user experience, with the main menu shown in Figure 4a. Users can interact with the elements

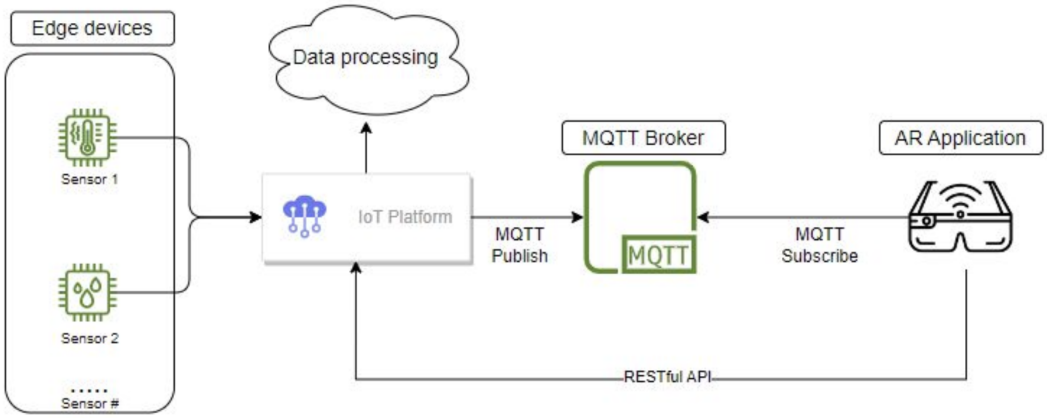
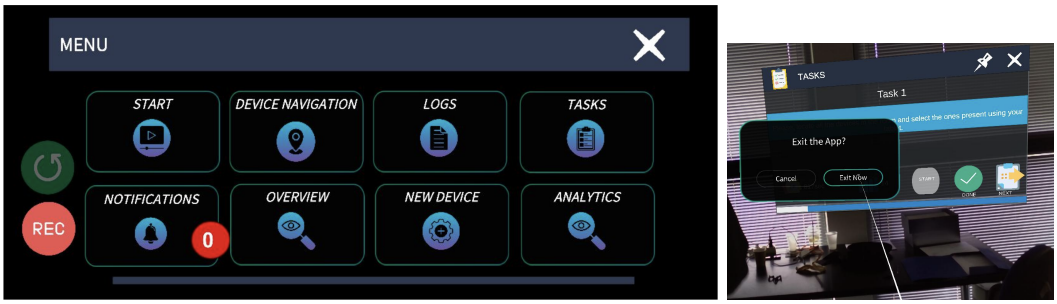


Fig. 3. Communication architecture between Devices and Software



(a) Application Features - Main Menu (b) Virtual Beam

Fig. 4. AR application - Features and Interaction

of the AR application using a handheld controller. Pointing the controller at an element creates a virtual beam, and selection is performed by clicking on the element using the controller (Fig. 4b). Following the principles of situated visualisation (Section 2.2.3), the AR interface minimizes visual clutter and supports in-situ inspection by anchoring panels to physical devices, using consistent status encodings.

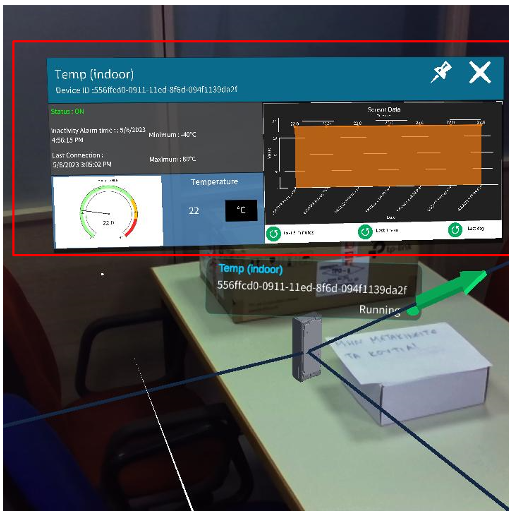
4.2.1 Start. The AR application guides users through a step-by-step environment setup and initiation of the monitoring and diagnosis process within the AR environment. Users begin by selecting *Start* from the main menu. This triggers the retrieval of all assets associated with the user from the ThingsBoard Platform via a RESTful API call to Immersal's servers, where each user-generated point cloud is securely stored. The user then selects the desired asset, which corresponds to the point cloud of the physical space. Subsequently, the application retrieves all the devices that have been registered on the IoT platform and associated with this space and prompts the user to confirm their positioning in the space. Once the devices have been positioned, the user needs to look around the physical space several times, allowing the Immersal SDK to ensure that the point cloud has adapted to the physical environment, and to localise the user within that environment (i.e. determines the user's initial position). Upon success, the devices are shown as 3D objects in the environment,



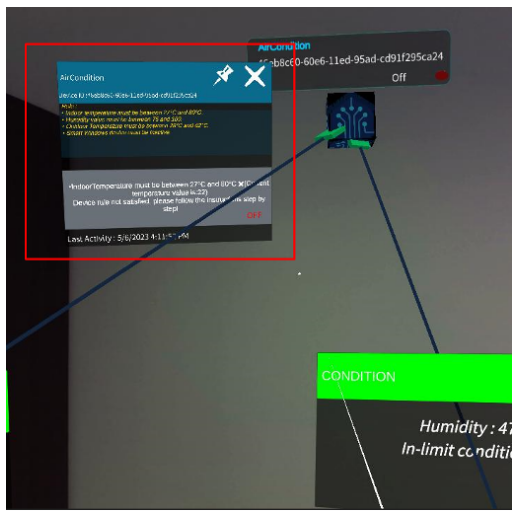
Fig. 5. Digitally augmented laboratory space: The user sees three devices - an indoor temperature sensor (left), a humidity sensor (bottom - middle) and an air conditioning device (top - middle) - along with their interconnections. The panels with the green header indicate that the operational conditions of each devices are met, i.e. the values of each device are between the permitted range. Device panels with blue headers provide information about each device, such as its name, status, permitted operational range, current value and graphs with analytics. The interconnections between the devices are depicted as straight lines connecting each present device. Green arrows on these lines indicate the exchange of data between the devices, with the direction of the arrow revealing which device is the sender and which the recipient of this exchange. The virtual beam from the controller can be seen at the bottom of the image.

and connecting lines between devices show the rule-based data flows across collaborating devices, using superimposed arrows that move along the connecting line.

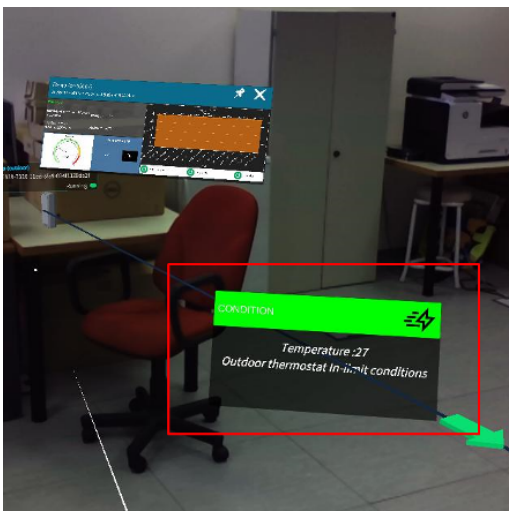
4.2.2 Inspecting devices and rules. By selecting a device with the controller, the user can bring up a panel over each device to show its current operating state. For sensor-type devices (Fig. 6a), the user can see the device ID, its current state, the minimum and maximum permissible values of the device, the last connection time of the device to the platform, the time and date of the latest inactivity alarm raised by the device, its current reported value (via graphical widget) and a time-series of reported values (as a line chart). For non-sensor devices (Fig. 6b), the user can see the device id, its current operating state (e.g. on/off), the rules that govern operation of the device, and any problems detected in the normal operation of these rules or explanations for the current device state (e.g. ”indoor temperature must be >20 degrees for device to be ON”). Connecting lines for collaborating devices always show a panel above the mid-point of the connecting line, to depict the rule logic binding the devices operation (Fig. 6c). A panel with a green header shows devices that are contributing data normally in the logical space of a rule’s execution (i.e. both sending data or appear as active to the platform, and the data is within normal operating range). An orange header indicates that both devices are active but the data being sent is outside normal parameters. Finally, if one device is inactive, the panel disappears and an animated exclamation mark is shown next to the device (Fig. 6d).



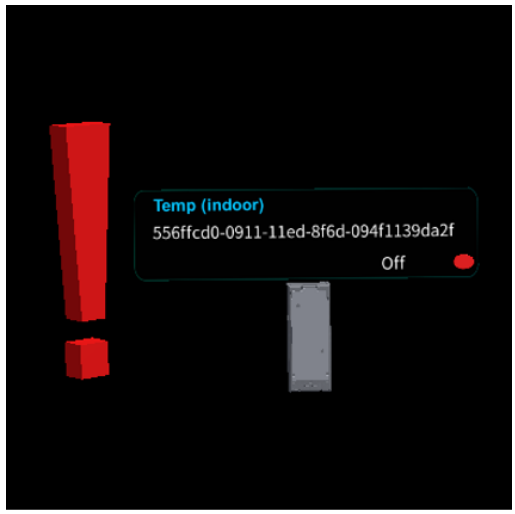
(a) Sensor panel: Users can see the ID, its current state, min and max values and other details about the selected sensor.



(b) Non-sensor panel: Users can see details about the selected device and the rules governing its operation in textual form.



(c) Panel showing the rule in textual form, that binds the devices operation.



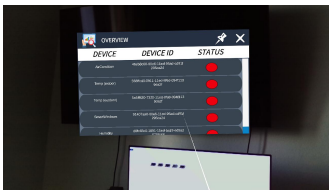
(d) An animated exclamation mark to show that a device is inactive.

Fig. 6. Device and Rule Inspection visual elements within the AR application.

4.2.3 Locating Devices. The AR application allows users to locate and inspect registered devices. The user first selects the target device from a list of devices present in the room. A C# script takes as input the target device and the position of the user's camera in the 3D space and calculates the distance between the user and the desired device, generating a virtual arrow guiding the user toward the device's location (Fig. 7a).



(a) The Device Location feature. The virtual arrow in the middle of the field of view of the user guides them towards the Humidity sensor.



(b) List of the present devices. Users see the device name, its id and current status. Here all devices appear to be inactive (red indicator).



(c) Users see graphs with analytics about e.g. the present devices, or the number of amount of activations at a certain period.

Fig. 7. Device Location, System Overview and Analytics

4.2.4 Logs. Provides logs related to communication between devices and the AR application. These logs accurately record when a device sent data or which device activated another by recording the corresponding timestamps.

4.2.5 Notifications. Notifications inform users of system events, enabling rapid response to issues. They are implemented using C# scripts, and each notification is triggered when an exception or error occurs in the system, and the user is alerted by the application also through a voice message, prompting them to refer to the notifications panel.

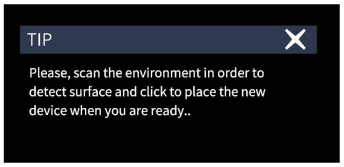
4.2.6 Overview. The AR application provides a list of all IoT devices and their current status, retrieved via HTTP and MQTT communication with the IoT platform. The displayed information includes the device name, its ID and status (Fig. 7b). A red indicator in the Status attribute signifies an inactive device, while a green indicator represents an active device. A yellow indicator denotes a device in a critical state, requiring the user's attention.

4.2.7 New Device. The AR application enables users to add and automatically integrate new devices at desired locations.. This feature leverages the NRSDK module, which is responsible for identifying flat surfaces within the physical space, ensuring the positioning of the virtual content with high accuracy and facilitating interaction through the AR application. To register a new device in the system, the user has to follow specific steps, as guided by the application. Initially, the user has to scan the physical environment, to detect a flat surface and then indicate the desired position of the new device (Fig. 8a). Upon selection, the user is presented with a panel displaying the 3D position of the device within the space (Fig. 8b). If the user confirms the position, they proceed to a subsequent panel, where they specify whether the device is a sensor and complete the device registration (Fig. 8c). The device is automatically added to the system and integrated into the IoT platform via a RESTful API. If the user wishes to create rule chains involving the new device, they can do so through the ThingsBoard interface.

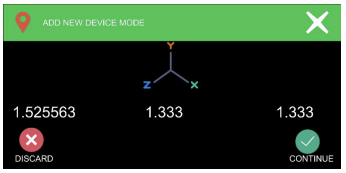
4.2.8 Analytics. The application also provides detailed statistical insights using graphs. Once historical data is retrieved from the IoT platform, the AR application uses algorithms for the required analysis and application of business logic (Fig. 7c).

4.3 Integration of the AR app with the Habitable Smart Space

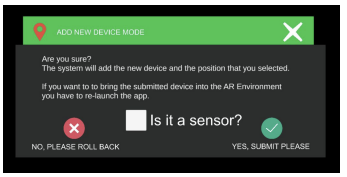
To configure the AR application for use within the HBS, users must follow specific steps to ensure seamless integration within the designated space. First, users scan the space using the Immersal



(a) User should scan the environment and select the position of the new device.



(b) 3D coordinates of the selected position for the new devices.



(c) User needs to indicate whether the new device is a sensor and confirm its creation.

Fig. 8. Device Registration steps

Mapper application. This requires uploading two-dimensional images of the target space to generate its point cloud representation, which is stored on Immersal's servers and is made available via a RESTful API call as an asset-type object within the IoT platform ThingsBoard.

In addition to spatial mapping and initial positioning of devices through the AR application (as per the "New device" feature), users must further configure the devices through the ThingsBoard platform. This includes defining their attributes and the connections with the environment and other devices. When adding a device, users can specify its type, name, characteristics related to the device or its general profile/category, or the allowed operational values of the device. The connections between devices and other system entities are determined by the rule chains users define within the ThingsBoard platform. ThingsBoard provides an interface that facilitates the creation and management of these rule chains for each system. Once the configuration process is complete, the AR application can be deployed for use. All necessary data is retrieved from ThingsBoard and the application overlays the digital elements on the physical environment, thereby enabling augmented interactions within the space.

4.4 Scaling and Performance

We tested the AR application's performance with up to 10 simulated IoT devices without any emergent issues in response time, visualisation, latency or other factors affecting its performance. The AR application is meant to work at a room-level scale, where it is unlikely that hundreds of IoT devices might be simultaneously present (in contrast to, for example, an industrial environment). Therefore it is easy to scale the system across multiple rooms in a HBS, by modelling each space and its devices separately in ThingsBoard, with devices then loaded dynamically based on user location. As such, we do not anticipate any performance issues even if the application environment refers to expansive settings, such as a multi-bedroom home, or even a large residential or office setting with many rooms.

5 Evaluation

To assess the effectiveness and usability of the proposed system, an experiment was conducted to explore how different representational paradigms support fault diagnosing in an IoT system.

5.1 User Interfaces

The experiment was conducted in a controlled laboratory setting, comparing two distinct representational paradigms. The first paradigm involved the use of a 2D desktop interface, specifically designed for the experiment within the ThingsBoard environment, accessed via desktop (Figure 9). To build the 2D desktop interface, we leveraged ThingsBoard's dashboard implementation feature, and designed the UI to resemble a typical dashboard desktop interface for IoT systems,

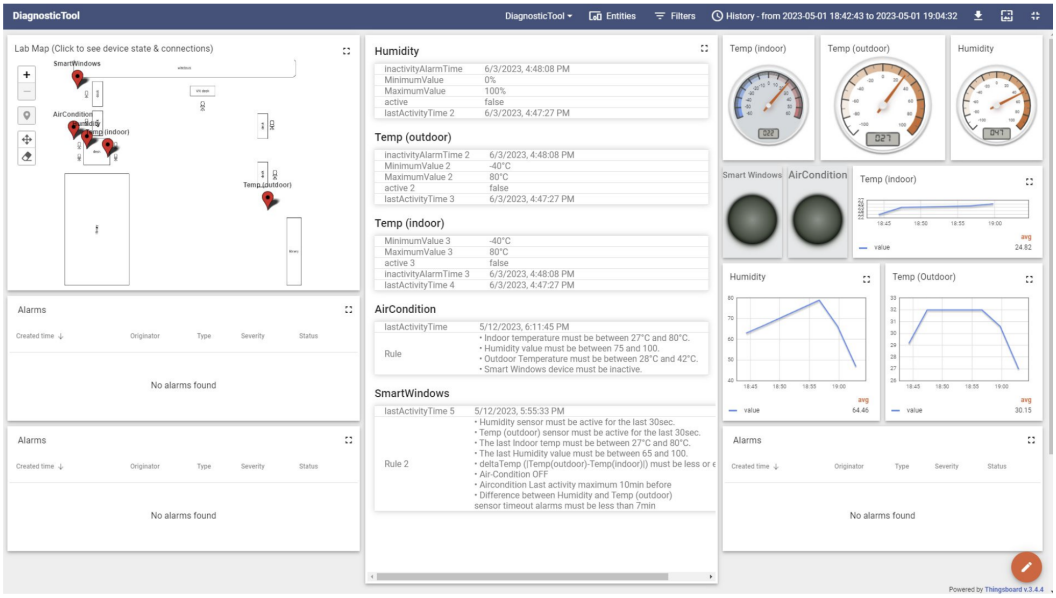


Fig. 9. 2D diagnostic tool - Users view panels with information for each present device, including their current values, graphs and panels including alarms when a device need the user's attention. A lab map is also available, indicating the exact position of each device inside the laboratory.

as could be expected to be present in many smart home deployments, either provided by the OEM (e.g. https://www.tp-link.com/app/tapo/tapo-for-pad/, https://www.apple.com/home-app), or through third parties (e.g. https://www.home-assistant.io/). The 2D desktop interface includes often-encountered graphical features in commercial applications, such as a floor plan, gauges, graphs and data readouts. Such dashboards are adaptive to the specific devices present in the smart environment and are user-configurable in terms of display style and content layout. Our 2D desktop environment offers a similar experience pre-configured by us, which is necessary in order to provide a homogeneous experience for all participants and remove confounding factors from the experiment. Users interacted with the platform by clicking on its elements using the desktop's mouse.

The second paradigm involved using the proposed AR application within the digitally augmented physical laboratory space (Figure 5). Users could see the digital elements via the AR glasses and interact with them using the handheld controller. The two interfaces differ fundamentally in their representational and interaction paradigms. The 2D desktop interface presents symbolic and abstract representations of system state and rules, whereas the AR interface provides embodied, situated visualisations directly aligned with the physical devices. Crucially, in both interfaces the rules are depicted in textual form. These differences are inherent to the respective interfaces, as the AR interface naturally affords spatial embedding and interaction within the physical environment, while the 2D interface affords access to the IoT setup through abstract and symbolic representations, but are central to the comparison, as they reflect distinct real-world interface paradigms for interacting with IoT systems. Therefore, this study does not aim to isolate the effect of AR as a single factor, but rather to explore how different representational approaches support diagnostic reasoning.

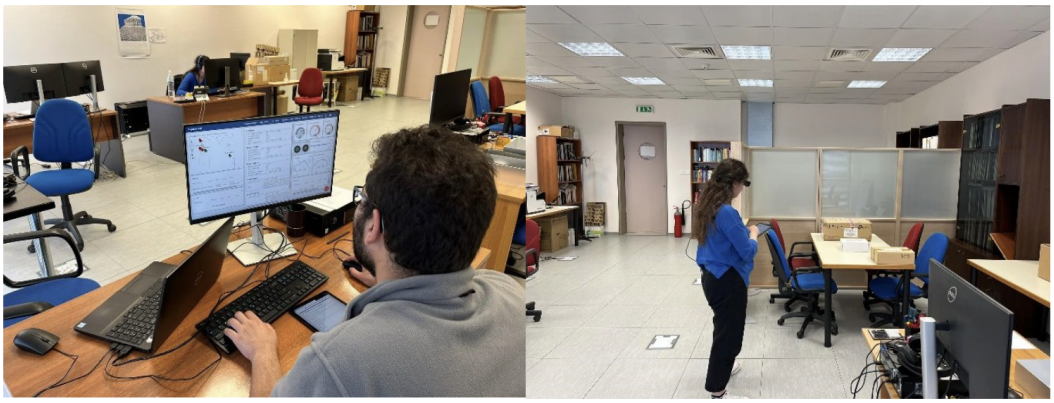


Fig. 10. **Left**: Experiment participant using the 2D desktop application; **Right**: Experiment participant using the AR application.

5.2 Experiment configuration

The experiment aimed to evaluate the usability and effectiveness of the proposed solution in diagnosing faults in IoT systems, and to assess the overall user experience. The experiment HBS was set up in one of our university's laboratories, and was pre-configured by the researchers, with five virtual devices (an indoor temperature sensor, an outdoor temperature sensor, a humidity sensor, an air conditioning unit, and a smart windows device) created and positioned in the pre-scanned space's point cloud. The necessary HBS configuration was carried out by the researchers, with participants needing to familiarise themselves with the HBS environment and carry out diagnostic tasks.

Participants were first welcomed to the laboratory, informed about the procedure, and asked to sign the necessary consent form. Then, they completed a demographic and technology familiarity questionnaire, which focused on the participants' familiarity with the concepts of the Internet of Things (IoT) and Augmented Reality (AR). Before proceeding to the tasks, participants underwent a brief training session to familiarize themselves with the user interfaces. All participants tested both techniques in a within-groups design. Scenario order was counterbalanced, with half of the participants starting with the AR-based system and the other half starting with the 2D desktop system (Fig. 10). After completing the tasks in each scenario, participants were asked to fill out a NASA-TLX questionnaire to evaluate the perceived workload associated with the system they used.

5.3 Participants

A total of 25 participants were recruited for the experiment. The duration of each participant's trial was one hour. 20 out of the 25 participants (80%) were male. The participants' ages ranged from 19 to 34 years ($\bar{x}$ = 23.32, σ = 3.6482). Eighteen participants (72%) were undergraduate students, and four (16%) were postgraduate students from the local university. Two participants (8%) were PhD students, and one held a master's degree. Regarding their academic backgrounds, 84% of the participants were from the Computer Engineering - Computer Science field, 12% were from the Electronic and Electrical Engineering field, and 4% were from the Mechanical Engineering field. After completing the demographics questionnaire, participants were asked about their level of familiarity with the concepts of the Internet of Things (IoT) and Augmented Reality (AR). Eighteen participants (72%) reported being familiar with the concept of IoT, although a smaller percentage indicated that they work professionally with or develop such systems. Similarly, 17 participants

(68%) were familiar with the concept of AR, though only a few were professionally involved in or develop AR systems. The participants' responses to familiarity questions are included in the paper's supplementary materials.

Given the orientation of our system to non-expert users, as defined previously, the sample chosen serves as a proxy for the intended user group. To further clarify, *non-expert* in our study does not refer to users with low digital literacy, but to those who are not experts in the specific IoT deployment being diagnosed (i.e. system-agnostic troubleshooters). This aligns with our target population, as described in Section 1, namely digitally literate users who may have general technical competence but lack prior knowledge of the particular system they are asked to diagnose. Hence, we recruited participants with general technical backgrounds but ensured that none had prior exposure to the experimental smart-space configuration, the devices, the implemented rules, or the fault scenarios used in the study. Our participants can be considered as non-experts in the sense that while they have a level of technical skill, they are not familiar with the particular deployment they are called to troubleshoot. While our participant sample may not fully represent all real-world users of IoT systems (e.g. homeowners or field-service engineers), it provides a reasonable proxy of digitally literate users capable of reasoning about system behavior without prior system-specific knowledge. IoT Automation systems such as Smart Home Technology (SHT) appeal to specific parts of the general population, with owners and persuaded persons characterised by higher income, being male, homeownership, larger home size, higher rates of technology adoption and use, greater perception of SHT benefits, and less concern with SHT risks [36]. Further, perceived ease of use (related to digital literacy and technology skills) is shown to be a primary driver for SHT adoption in multiple studies [27, 52]. Our sample reflects several of these characteristics (mostly male, computer or electronics engineering students and graduates that understand smart home technology benefits, high adopters of technology, digitally literate, expected to have higher salaries in their professional careers). The participant sample size (25) exceeds that in other studies in related literature (ranging between 10-18 in [39, 41, 64]) and falls into the upper levels of sample sizes for in-person usability studies published at prestigious conferences such as CHI (mean=16, SD=12 participants) [8].

5.4 Tasks

Eight tasks were designed to simulate plausible issues within the HBS system. The diagnostic tasks were designed to reflect common fault scenarios in IoT-based smart environments, drawing inspiration from existing taxonomies of smart home failures and debugging challenges reported in prior work [59]. These tasks were structured to cover a range of interactions with increasing difficulty, thereby enabling the evaluation of the effectiveness of the application. Participants were given a tablet that provided instructions for each task. They also used the tablet to record their responses to multiple-choice questions, by selecting the most appropriate answers based on the given task. The tasks required participants to retrieve information about the HBS and its state. This information relates to the diagnosis of potential issues, but participants were not required to interact with the system to actually fix problems.

The tasks given to the participants were of increasing difficulty ratings [Low: $T1 - T3$, Medium: $T4 - T5$, High: $T6 - T8$]. The low difficulty tasks were intended to introduce users to the basic means of acquiring information about the HBS's basic building block (devices, sensors and rules), while medium and high difficulty tasks were intended to introduce participants to more complex diagnosing, which involved combining information from multiple elements. In detail, the tasks were:

- **Task 1**: Participants were asked to locate all the devices in the room. They had to select the present devices from a multiple choice list, which included the devices, among other IoT

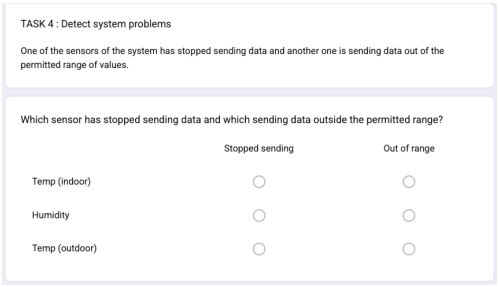
devices that were not present in the room. This task was designed to familiarize participants with the environment.

- **Task 2**: Participants were instructed to locate the device of the indoor temperature sensor. Having located it, they were asked to answer three multiple choice questions to indicate the correct maximum and minimum allowed temperature value of the sensor and its current value. The goal of this task was to help participants get acquainted with interacting with devices and finding the required information on the device panels.
- **Task 3**: Participants were asked to identify the rule that applied to the operation of the Air Conditioning device. The rule had to be selected from a list of six choices. The aim was to enable participants to interact with the device and utilize the device panel to retrieve the necessary information.
- **Task 4**: Participants had to detect which of the sensors had stopped sending data and which sensor was sending data out of range. They were presented with a list of the three sensor devices and had to indicate the status of each one (Fig. 11a). This task helped participants quickly identify such problems in the IoT system and provided feedback on visualisations and messages.
- **Task 5**: Participants were instructed to identify the exchange of data between devices in the HBS (Fig. 11b). The objective was to evaluate the ease of discovering data flow from one device to another and the connections between them.
- **Task 6**: In this task, the Air Condition device was inactive and participants were asked to select from a list the reasons for its state. This task aimed to help users identify the rule of the device and the instructions provided by the system, while cross-checking these instructions with the current values of other sensors and devices.
- **Task 7**: Similarly to $T6$, participants were asked to indicate why the Air Condition device was inactive. This task also focused on diagnosing the state of the air conditioner, with an increased level of difficulty. However, the possible answers urged the participants to identify which conditions were not satisfied, which meant that users had to perform a combined value check, whereas in $T6$ the provided options guided users to check whether a single value was inside the permitted range.
- **Task 8**: Participants were instructed to select from a list of conditions which ones were not satisfied, causing the Smart Windows device to be inactive. The purpose of this task was to evaluate both scenarios (2D and AR) for a complex diagnostic task in the system within a complex ecosystem.

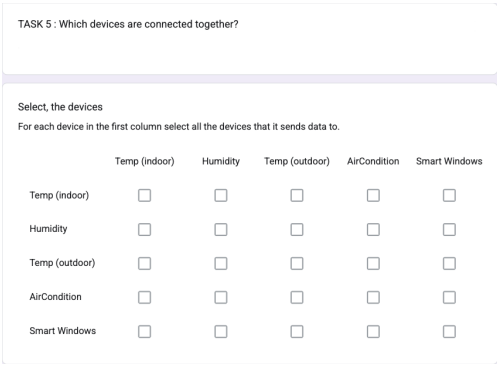
The questions and possible answers are provided for completeness in the supplementary material for this paper. Since participants carried out the same tasks for both conditions, it was important to ensure that the correct answers to the task questions were different between conditions, to prevent participants from remembering responses from the first condition of exposure. Therefore, at the beginning of each task, we manipulated the device states (which in turn affected rule execution) in each condition appropriately, to ensure that the correct responses to each question were different for each condition. We also ensured that the number of correct choices in each task (single or multiple checked items) remained the same across conditions. For all tasks, appropriate Python scripts were developed to capture user interactions with the applications in real time. These interaction logs were subsequently used for analysis.

6 Results

The results presented in this section should be interpreted in light of the differences between the two representational paradigms. The AR and 2D interfaces vary not only in display modality, but



(a) The question and possible answers for Task 4.



(b) The question and possible answers for Task 5.

Fig. 11. Questions participants had to complete during the two conditions.

also in their representational and interaction capabilities, as AR offers the potential for spatial embedding of information and the embodied exploration of information-rich workspaces. As such, the comparison does not relate to the effect of AR technology per se - for example, we could have used an AR headset to display a symbolic floating 2D interface, similar to, if not exactly the same as our desktop version. Instead, the findings reflect how the implementation of situatedness and embodied interaction designs that are uniquely available in AR, influence users' diagnostic performance and perceived workload. Consequently, observed differences between conditions should be understood as arising from the combined effects of representation, embodied interaction and situatedness, rather than being attributed to any single factor.

6.1 Task completion times

All participants successfully completed all tasks. The average execution times (in seconds) for each task across the two conditions are presented in Table 1. To assess whether the difference in completion times across all tasks between the two conditions was statistically significant, a Shapiro-Wilk test was initially conducted to determine whether the times in all tasks for all participants in both conditions followed a normal distribution. Statistical results are reported with a significance level of $\alpha = 0.05$. For tasks where the execution time data did not follow a normal distribution, the Wilcoxon signed-rank test was applied. Conversely, for tasks where the data was normally distributed, the paired t-test was used.

Statistical test results indicate a significant difference in the completion time of higher-difficulty tasks across the two conditions, with users completing these three tasks more quickly using the situated and embodied AR interface. The results also show a significant difference in average execution times for the first task, which was designed to familiarize users with the interfaces. This difference may reflect that users might have spent more time exploring the multiple items displayed together in the 2D desktop interface (potential information overload), whereas using the situated and embodied AR sytem, they primarily focused on locating the relevant devices.

The mean total completion time for all tasks using the situated and embodied AR interface was 734.52 seconds ($\sigma = 262.70$), while using the 2D desktop interface, it was 951.08 seconds ($\sigma = 322.91$). The Shapiro-Wilk test results indicated that the data did not follow a normal distribution (AR: $p = 0.029$; 2D: $p = 0.038$). Therefore, the Wilcoxon Signed Rank Test was selected as a non-parametric alternative, and its results indicated that the difference between the total execution times for the two conditions was statistically significant ($W = 62.0, p = 0.0056, Z = -2.704, r = 0.541$).

Task	Difficulty Level	AR		Desktop		Statistic	p-value	Effect Size
		$\bar{x}$	σ	$\bar{x}$	σ			
1	Low	61.88	52.98	169.90	250.92	W = 65.0	**0.007**	r = 0.525 (Z = -2.623)
2	Low	47.56	26.41	61.85	51.69	W = 124.0	0.312	r = 0.207 (Z = -1.036)
3	Low	59.48	41.26	64.11	28.06	W = 102.0	0.107	r = 0.326 (Z = -1.628)
4	Medium	119.32	54.69	121.45	48.13	t = 0.931	0.866	d = -0.034
5	Medium	108.48	77.72	106.50	45.24	W = 130.0	0.396	r = 0.175 (Z = -0.874)
6	High	76.76	57.80	100.27	46.66	W = 78.0	**0.022**	r = 0.455 (Z = -2.274)
7	High	77.72	74.78	97.36	37.55	W = 70.0	**0.011**	r = 0.498 (Z = -2.489)
8	High	183.32	126.77	229.59	80.51	W = 77.0	**0.020**	r = 0.460 (Z = -2.301)

Table 1. Average execution time (in seconds) per task.

Group of Tasks	AR		Desktop		Wilcoxon statistic	p-value	Effect Size
	$\bar{x}$	σ	$\bar{x}$	σ			
(T1, T2, T3)	56.30	28.45	98.61	84.98	62.0	**0.0058**	r = 0.541 (Z = -2.704)
(T4, T5)	113.90	47.05	113.98	36.30	137.0	0.5077	r = 0.137 (Z = -0.686)
(T6, T7, T8)	112.60	74.12	142.40	47.14	74.0	**0.0160**	r = 0.476 (Z = -2.381)

Table 2. Statistical test results for differences across different groups.

Specifically, the total execution time using the situated and embodied AR interface was statistically significantly shorter compared to the total time using the 2D desktop interface.

Next, the tasks were divided into 3 levels of difficulty, and the mean execution times for each level of difficulty were calculated. The execution times did not follow a normal distribution in none of the groups. As a result, the Wilcoxon Signed Rank Test was applied again, and the findings are summarized in Table 2. Users executed higher-difficulty tasks significantly faster using the situated and embodied AR interface.

6.2 Task completion accuracy

Following, the accuracy of the users' responses for each task in both scenarios was evaluated. For Task 1 and Task 3, all users answered correctly in both scenarios. For Task 2, using the 2D desktop interface, all users answered correctly, whereas only one user answered incorrectly using the situated and embodied AR interface. For Task 4, 76% of the users selected all correct answers using the 2D desktop interface, while using the situated and embodied AR interface, 72% did so. In

	AR		2D	
Dimension	$\bar{x}$	σ	$\bar{x}$	σ
Mental Demand	55.4	22.90	61	23.49
Physical Demand	36.8	23.08	10.4	5.75
Temporal Demand	41.4	21.33	43.8	23.77
Performance	32	21.40	44.8	25.87
Effort	54.4	21.37	55.2	20.99
Frustration	20.8	20.19	24.8	19.12

Table 3. Mean values across NASA-TLX metrics

both scenarios, 84% of the participants selected all correct answers in Task 5, 92% in Task 6, and 64% in Task 7. Finally, for Task 8, 60% of participants selected all correct conditions using the situated and embodied AR interface, compared to only 32% using the 2D desktop interface.

To assess whether there was a statistically significant difference between the two scenarios in terms of user errors in the tasks, the total percentage of incorrect answers per user was calculated. The average number of incorrect answers per user using the situated and embodied AR interface was 1.6 ($\sigma = 1.02$), while using the 2D desktop interface it was 1.88 ($\sigma = 1.14$). The test results ($W = 7.0$, $p = 0.1072$) indicated that there was no statistically significant difference between the two scenarios regarding users' total incorrect answers.

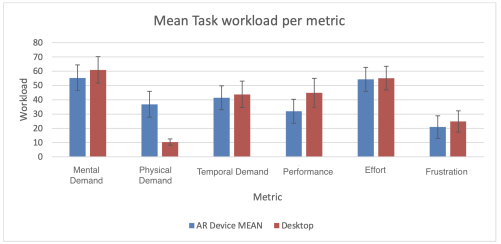
6.3 Subjective task difficulty

Finally, the data collected from the NASA-TLX questionnaires, completed by participants after each scenario, were analyzed to evaluate the two scenarios in terms of the workload experienced by users during interaction with the alternative user interfaces. The mean values for each workload metric (Table 3) are presented in Figure 12a, while the overall mean workload per scenario for all participants is displayed on Figure 12b.

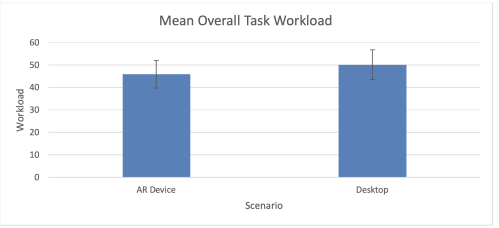
The total mean workload using the situated and embodied AR interface was 45.88 ($\sigma = 15.37$) and using the 2D desktop interface was 50.16 ($\sigma = 16.54$). The difference in the total mean workload between the two scenarios was not statistically significant ($t = -1.076$, $p = 0.2926$, $d = -0.215$). However, when examining each metric individually, statistically significant differences were observed between the two scenarios for *Physical Demand* ($W = 0.0$, $p = 0.0003$, $Z = -4.197$, $r = 0.875$) and *Performance* ($W = 35.5$, $p = 0.009$, $Z = -2.59$, $r = 0.580$). The higher Physical Demand in the AR scenario may be explained by the need for users to move around the physical space of the lab, whereas in the 2D scenario, they remained seated. The increased physical effort in the AR scenario likely contributed to the observed difference. Additionally, the lower Performance ratings in the AR scenario may be attributed to users' uncertainty when interacting with an unfamiliar system. As they adapted to interacting with the situated and embodied AR interface, they might have perceived their performance as lower, despite completing the tasks correctly.

7 Discussion

Previous research in Augmented Reality for habitable smart spaces has focused on usability evaluations of this technology for the purposes of state monitoring and control (e.g. [5, 28, 31]), while some past work has directly compared AR and 2D user interfaces for the same purpose [21, 65]. Fault diagnosis and debugging in HBSs has previously been investigated with a focus on 2D user interfaces only (e.g. [4, 14, 15, 63]). To our knowledge, there has been no previous research that examined how different representational paradigms, such as spatially embedded AR interfaces



(a) Mean workload per metric in both scenarios.



(b) Mean overall workload per scenario.

Fig. 12. Results of Nasa-TLX questionnaires

and traditional symbolic desktop dashboards, support diagnostic reasoning in IoT-based HBSs. To address this gap, our work presents the engineering aspects of a custom HBS implementation using open-source technologies, and examines how a situated and embodied AR system and a 2D desktop interface, built on the same underlying IoT management system, support diagnostic tasks in a usability study with a proxy sample of digitally literate users unfamiliar with the specific system. Our core findings indicate that participants using the situated and embodied AR system completed diagnostic tasks more quickly, particularly for higher-complexity tasks, while also experiencing increased physical demand, highlighting the trade-offs between representational approaches.

The results demonstrated that participants completed diagnostic tasks, especially those rated higher in difficulty, significantly faster using the situated and embodied AR system compared to the 2D desktop interface. This outcome supports claims in existing literature suggesting AR's potential benefits [5, 28, 31]. One possible explanation is the spatial visualisation provided by the situated and embodied AR system, which presents digital information overlaid directly onto physical devices and visualise data interconnections through dynamic lines and arrows. Such representations may reduce the need for mental mapping between abstract representations (like those in the 2D desktop interface) and the physical world, and may support users' understanding of system behaviour when diagnosing issues involving multiple devices. The significantly longer time observed for the initial device identification task (Task 1) using the 2D desktop interface may also reflect increased cognitive demands when navigating multiple items presented simultaneously in traditional desktop dashboards compared to the more focused, spatially anchored information presented in AR.

Interestingly, the increased efficiency using the situated and embodied AR system did not correspond to higher diagnostic accuracy; error rates were statistically similar between the two interfaces. This suggests that, while participants completed tasks more quickly using the situated and embodied AR system, both interfaces provided sufficient information clarity for users to achieve comparable levels of correctness for the specific tasks evaluated. One possible interpretation is that factors such as the novelty of the AR interaction or minor usability challenges may have influenced performance, potentially affecting how users approached the tasks. Our findings also are consistent with those of Zhou et al. [65], where 2D interfaces supported cognitive tasks as effectively as AR, and Flick et al. [21], who reported that for certain HBS control tasks, 2D interfaces can be equally or more effective than AR. Taken together, our results indicate no clear difference between conditions in terms of diagnostic accuracy.

Analysis of subjective feedback collected using the NASA-TLX surveys also revealed important nuances. The situated and embodied AR system was associated with significantly higher Physical Demand. This may stem directly from the requirement for users to physically move within the laboratory space, locate devices, and interact using the handheld controller and head movements,

contrasting with the seated interaction of the 2D desktop interface. This suggests a trade-off, where the spatial characteristics of the AR system currently necessitate increased physical effort, which could limit its applicability for extended use or for users with mobility constraints. Furthermore, participants rated their own Performance significantly lower when using the situated and embodied AR system, despite achieving comparable accuracy and faster completion times. This perception might reflect factors such as unfamiliarity with HMD-based AR interaction, the cognitive effort of managing both physical and digital information simultaneously, or minor usability challenges with the controller-based interaction. This discrepancy underscores the need for intuitive interaction design and effective user onboarding in AR diagnostic tools.

Our claims, therefore, are presented under some nuance. Participants using the situated and embodied AR system completed diagnostic tasks more quickly, particularly for complex IoT scenarios, which may be related to the spatial and situated representation of system information. However, no differences in diagnostic accuracy were observed between interfaces, and the situated and embodied AR system was associated with greater physical demand. These findings suggest that practical deployment of AR-based diagnostic tools requires careful consideration of ergonomic factors and interaction design, particularly to support user confidence and perceived performance. Taken together with the findings in prior work, our results indicate that situated and embodied AR may be considered a complementary interface for time-critical, spatially oriented diagnostic tasks, rather than a complete replacement for traditional 2D desktop interfaces in IoT fault management.

Despite the promising results, certain limitations must be acknowledged. The study was conducted in a controlled laboratory setting, which may enhance internal validity, but may not fully capture the complexities of real-world smart environments (e.g. variable lighting, physical clutter, unpredictable device failures). The participant sample, primarily young engineering students, while reflecting a proxy sample of technically literate users unfamiliar with a specific system, may not reflect the experiences of the broader population, including older adults or those with lower digital literacy, who are potential beneficiaries of smart home technology (e.g. in Ambient Assisted Living contexts). Results could differ with these groups, potentially impacting adoption barriers such as perceived ease of use and physical comfort. Additionally, the AR system may have been influenced by novelty effects, as some participants had limited prior experience with head-mounted AR devices, which could have influenced both subjective workload ratings and engagement. Finally, the specific hardware used (NREAL Light glasses, handheld controller) has its own limitations (e.g. field-of-view, interaction ergonomics) that could influence findings.

Future research should prioritize several directions. Firstly, longitudinal studies ”in the wild” are crucial to evaluate the system’s usability and effectiveness in realistic, everyday smart home or office environments. Such studies should involve more diverse participant groups, including homeowners and professional technicians. Secondly, addressing the physical demand identified requires exploring and evaluating alternative interaction modalities, such as gesture or voice control, comparing their ergonomic impact and efficiency against the current controller-based method. Thirdly, future work should aim to more systematically isolate the effects of AR as a representational medium by controlling for differences in visualisation and interaction design. For instance, comparative studies could evaluate situated and embodied AR systems against 2D desktop interfaces that incorporate similar graph-based or spatial representations of rule connections, allowing for a more precise examination of the contribution of spatial embedding itself. Fourthly, further investigation into optimizing AR visualisations is warranted – could adaptive displays, different representations of rule logic, or context-aware information filtering further enhance understanding and reduce cognitive load or errors? Furthermore, while NASA-TLX provides a validated subjective workload measure, it does not directly isolate cognitive load mechanisms;

future work could incorporate additional objective indicators such as error recovery time, eye-tracking, or physiological measures. Finally, integrating advanced functionalities, such as machine learning algorithms for predictive maintenance, directly within the AR interface could significantly enhance the tool's diagnostic power and preventative capabilities, providing users with proactive insights rather than purely reactive diagnostic information.

8 Conclusion

This work investigated the use of Augmented Reality (AR) for supporting the diagnosis of IoT system faults in HBSs by non-expert users, comparing an AR-based diagnostic tool with a traditional 2D desktop interface. Addressing a gap in empirical studies in this domain, our study showed that participants using the situated and embodied AR system completed diagnostic tasks more quickly, particularly for higher-complexity tasks, while achieving comparable levels of accuracy to when using the 2D desktop interface. This suggests AR's situated visualisation and direct interaction capabilities may effectively support the process of understanding system states and identifying issues within complex IoT ecosystems. However, this efficiency gain was counterbalanced by increased physical demand associated with the use of the situated and embodied AR system.

In conclusion, this study highlights potential benefits and trade-offs associated with AR-based diagnostic tools for IoT systems. While the AR-based system was associated with faster task completion, it also involved greater physical effort. These findings suggest that the design of AR-based diagnostic interfaces should carefully consider ergonomic factors and interaction techniques to support usability and user confidence. Although AR shows promise for supporting diagnostic tasks in complex smart environments, further refinement and evaluation in real-world contexts are needed to better understand its applicability and limitations. The continued development of intuitive and less physically demanding interaction techniques will be important for enabling broader integration into everyday HBS management.

References

[1] Leonor Adriana Cárdenas-Robledo, Óscar Hernández-Uribe, Carolina Reta, and Jose Antonio Cantoral-Ceballos. 2022. Extended Reality Applications in Industry 4.0. – A Systematic Literature Review. *Telematics and Informatics* 73 (Sept. 2022), 101863. doi:10.1016/j.tele.2022.101863

[2] Zaid Abbas Al-Sabbag, Chul Min Yeum, and Sriram Narasimhan. 2022. Enabling Human–Machine Collaboration in Infrastructure Inspections through Mixed Reality. *Advanced Engineering Informatics* 53 (Aug. 2022), 101709. doi:10.1016/j.aei.2022.101709

[3] Manuel Alonso-Rosa, Aurora Gil-de-Castro, Antonio Moreno-Munoz, Joaquín Garrido-Zafra, Elena Gutierrez-Ballesteros, and Eduardo Cañete-Carmona. 2020. An IoT Based Mobile Augmented Reality Application for Energy Visualization in Buildings Environments. *Applied Sciences* 10, 2 (Jan. 2020), 600. doi:10.3390/app10020600

[4] Margherita Andrao, Fabrizio Balducci, Bernardo Breve, Federica Cena, Giuseppe Desolda, Vincenzo Deufemia, Cristina Gena, Maristella Matera, Andrea Mattioli, Fabio Paternò, Carmen Santoro, Barbara Treccani, Fabiana Vernero, and Massimo Zancanaro. 2023. Understanding Concepts, Methods and Tools for End-User Control of Automations in Ecosystems of Smart Objects and Services. In *End-User Development*, Lucio Davide Spano, Albrecht Schmidt, Carmen Santoro, and Simone Stumpf (Eds.). Springer Nature Switzerland, Cham, 104–124. doi:10.1007/978-3-031-34433-6_7

[5] Raffaele Ariano, Marco Manca, Fabio Paternò, and Carmen Santoro. 2023. Smartphone-based augmented reality for end-user creation of home automations. 42, 1 (2023), 124–140. doi:10.1080/0144929X.2021.2017482 Publisher: Taylor & Francis.

[6] Guilherme Brittes Benitez, Néstor Fabián Ayala, and Alejandro G. Frank. 2020. Industry 4.0 innovation ecosystems: An evolutionary perspective on value cocreation. 228 (2020), 107735. doi:10.1016/j.ijpe.2020.107735

[7] Julia Brich, Marcel Walch, Michael Rietzler, Michael Weber, and Florian Schaub. 2017. Exploring End User Programming Needs in Home Automation. *ACM Trans. Comput.-Hum. Interact.* 24, 2 (April 2017), 11:1–11:35. doi:10.1145/3057858

[8] Kelly Caine. 2016. Local Standards for Sample Size at CHI. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 981–992. doi:10.1145/2858036.2858498

[9] Marco Casini. 2022. Extended Reality for Smart Building Operation and Maintenance: A Review. *Energies* 15, 10 (Jan. 2022), 3785. doi:10.3390/en15103785

[10] T.P. Caudell and D.W. Mizell. 1992. Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences* (Kauai, HI, USA, 1992). IEEE, 659–669 vol.2. doi:10.1109/HICSS.1992.183317

[11] Khalil Chakal, Mikko Korkiakoski, Hassan Mehmood, Theodoros Anagnostopoulos, Paula Alavesa, and Panos Kostakos. 2023. Augmented Reality Integration for Real-Time Security and Maintenance in IoT-Enabled Smart Campuses. In *2023 IEEE 31st International Conference on Network Protocols (ICNP)*. 1–6. doi:10.1109/ICNP59255.2023.10355580

[12] Hua-En Chang and Xin-Mao Huang. 2024. Utilizing AR Glasses for Identifying Smart Appliance Based on Appliance Relative Locations within Smart Home. In *2024 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*. 513–514. doi:10.1109/ICCE-Taiwan62264.2024.10674142

[13] Xiaohong Chen, Shi Chen, Zhi Jin, Han Bian, Zihan Chen, and Haotian Li. 2025. Expressing the Needs in Smart Home: What Is the End Users' Favorite Way. *ACM Trans. Comput.-Hum. Interact.* (Jan. 2025). doi:10.1145/3715114

[14] Sven Coppers, Davy Vanacken, and Kris Luyten. 2020. FORTNIoT: Intelligible Predictions to Improve User Understanding of Smart Home Behavior. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4 (Dec. 2020), 124:1–124:24. doi:10.1145/3432225

[15] Sven Coppers, Davy Vanacken, and Kris Luyten. 2022. FortClash: Predicting and Mediating Unintended Behavior in Home Automation. *Proc. ACM Hum.-Comput. Interact.* 6, EICS (June 2022), 154:1–154:20. doi:10.1145/3532204

[16] Corentin Coupry, Sylvain Noblecourt, Paul Richard, David Baudry, and David Bigaud. 2021. BIM-Based Digital Twin and XR Devices to Improve Maintenance Procedures in Smart Buildings: A Literature Review. *Applied Sciences* 11, 15 (Jan. 2021), 6810. doi:10.3390/app11156810

[17] Giuseppe Desolda, Carmelo Ardito, and Maristella Matera. 2017. Empowering End Users to Customize Their Smart Environments: Model, Composition Paradigms, and Domain-Specific Tools. *ACM Trans. Comput.-Hum. Interact.* 24, 2 (April 2017), 12:1–12:52. doi:10.1145/3057859

[18] Suyang Dong, Amir H. Behzadan, Feng Chen, and Vineet R. Kamat. 2013. Collaborative visualization of engineering processes using tabletop augmented reality. 55 (2013), 45–55. doi:10.1016/j.advengsoft.2012.09.001

[19] Yassir El Filali and Salah-ddine Krit. 2018. Augmented Reality Types and Popular Use Cases:. In *Proceedings of the 1st International Conference of Computer Science and Renewable Energies* (Ouarzazate, Morocco, 2018). SCITEPRESS - Science and Technology Publications, 107–110. doi:10.5220/0009776301070110

[20] Ali A. El-Moursy, Fadi N. Sibai, Jahanzeb Rehman, Omar M. Gouda, Abdelrahman T. Gaber, and Ahmed M. Khedr. 2022. Home Automation Using Augmented Reality (HAAR). *Wireless Personal Communications* 124, 2 (May 2022), 1525–1555. doi:10.1007/s11277-021-09419-7

[21] Connor Daniel Flick, Courtney J Harris, Nikolas T Yonkers, Nahal Norouzi, Austin Erickson, Zubin Choudhary, Matt Gottsacker, Gerd Bruder, and Greg Welch. 2021. Trade-Offs in Augmented Reality User Interfaces for Controlling a Smart Environment. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction (SUI '21)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3485279.3485288

[22] Daniel Fuentes, Luís Correia, Nuno Costa, Arsénio Reis, João Barroso, and António Pereira. 2021. SAR.IoT: Secured Augmented Reality for IoT Devices Management. *Sensors* 21, 18 (Jan. 2021), 6001. doi:10.3390/s21186001

[23] Yi Gao, Kaijie Xiao, Fu Li, Weifeng Xu, Jiaming Huang, and Wei Dong. 2024. ChatIoT: Zero-code Generation of Trigger-action Based IoT Programs. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3 (Sept. 2024), 103:1–103:29. doi:10.1145/3678585

[24] Mateus Carvalho Gonçalves, Lara , Otávio Neves, de Bettio , Raphael Winckler, and André Pimenta and Freire. 2021. End-User Development of Smart Home Rules Using Block-Based Programming: A Comparative Usability Evaluation with Programmers and Non-Programmers. *Behaviour & Information Technology* 40, 10 (July 2021), 974–996. doi:10.1080/0144929X.2021.1921028

[25] Morton L. Heilig. 1962. Sensorama simulator. https://patents.google.com/patent/US3050870A/en

[26] Jie Hua, Sangsu Lee, Gruia-Catalin Roman, and Christine Julien. 2021. ArcIoT: Enabling Intuitive Device Control in the Internet of Things through Augmented Reality. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops)*. 558–564. doi:10.1109/PerComWorkshops51409.2021.9431115

[27] Marco Hubert, Markus Blut, Christian Brock, Ruby Wenjiao Zhang, Vincent Koch, and René Riedl. 2018. The Influence of Acceptance and Adoption Drivers on Smart Home Usage. *European Journal of Marketing* 53, 6 (Sept. 2018), 1073–1098. doi:10.1108/EJM-12-2016-0794

[28] Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. 2018. Scenariot: Spatially Mapping Smart Things Within Augmented Reality Scenes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3173574.3173793

[29] Timo Jakobi, Corinna Ogonowski, Nico Castelli, Gunnar Stevens, and Volker Wulf. 2017. The Catch(Es) with Smart Home: Experiences of a Living Lab Field Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1620–1633. doi:10.1145/3025453.3025799

[30] Timo Jakobi, Gunnar Stevens, Nico Castelli, Corinna Ogonowski, Florian Schaub, Nils Vindice, Dave Randall, Peter Tolmie, and Volker Wulf. 2018. Evolving Needs in IoT Control and Accountability: A Longitudinal Study on Smart Home Intelligibility. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4 (Dec. 2018), 171:1–171:28. doi:10.1145/3287049

[31] Dongsik Jo and Gerard Jounghyun Kim. 2019. AR Enabled IoT for a Smart and Interactive Environment: A Survey and Future Directions. *Sensors* 19, 19 (Jan. 2019), 4330. doi:10.3390/s19194330

[32] Kinam Kim, Hongjo Kim, and Hyoungkwan Kim. 2017. Image-based construction hazard avoidance system using augmented reality in wearable device. 83 (2017), 390–403. doi:10.1016/j.autcon.2017.06.014

[33] Michal Kostoláni, Justín Murín, and Štefan Kozák. 2019. Intelligent predictive maintenance control using augmented reality. In *2019 22nd International Conference on Process Control (PC19)* (2019-06). 131–135. doi:10.1109/PC.2019.8815042

[34] Myron W. Krueger. 1977. Responsive environments. In *Proceedings of the June 13-16, 1977, national computer conference* (New York, NY, USA, 1977-03-13) *(AFIPS '77)*. Association for Computing Machinery, 423–433. doi:10.1145/1499402.1499476

[35] Wei Li, Zhoujing Ye, Yajian Wang, Hailu Yang, Songli Yang, Zhenlong Gong, and Linbing Wang. 2023. Development of a distributed MR-IoT method for operations and maintenance of underground pipeline network. 133 (2023), 104935. doi:10.1016/j.tust.2022.104935

[36] Wenda Li, Tan Yigitcanlar, Isil Erol, and Aaron Liu. 2021. Motivations, Barriers and Risks of Smart Home Adoption: From Systematic Literature Review to Conceptual Framework. *Energy Research & Social Science* 80 (Oct. 2021), 102211. doi:10.1016/j.erss.2021.102211

[37] Chieh-Jan Mike Liang, Lei Bu, Zhao Li, Junbei Zhang, Shi Han, Börje F. Karlsson, Dongmei Zhang, and Feng Zhao. 2016. Systematically Debugging IoT Control System Correctness for Building Automation. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. Association for Computing Machinery, New York, NY, USA, 133–142. doi:10.1145/2993422.2993426

[38] Qimeng Lu. 2024. Application of Data Visualization Technology in Fault Diagnosis and Maintenance of Intelligent Electromechanical Systems. 243 (2024), 716–723. doi:10.1016/j.procs.2024.09.086

[39] Atieh Mahroo, Luca Greci, Marta Mondellini, and Marco Sacco. 2023. Assessment of a Mixed Reality Smart Home Controller: HoloHome Pilot Study on Healthy Adults. *Virtual Reality* 27, 3 (Sept. 2023), 2673–2690. doi:10.1007/s10055-023-00834-8

[40] Atieh Mahroo, Luca Greci, and Marco Sacco. 2019. HoloHome: An Augmented Reality Framework to Manage the Smart Home. In *Augmented Reality, Virtual Reality, and Computer Graphics*, Lucio Tommaso De Paolis and Patrick Bourdot (Eds.). Springer International Publishing, Cham, 137–145. doi:10.1007/978-3-030-25999-0_12

[41] Andrea Mattioli, https://orcid.org/0000-0001-6766-7916, View Profile, Fabio Paternò, https://orcid.org/0000-0001-8355-6909, and View Profile. 2023. A Mobile Augmented Reality App for Creating, Controlling, Recommending Automations in Smart Homes. *Proceedings of the ACM on Human-Computer Interaction* 7, MHCI (Sept. 2023), 1–22. doi:10.1145/3604242

[42] Ning Miao, Xun Peng, Xintong Chen, Gang Ren, and Gang Wang. 2021. Investigating the Situational Awareness with Optical See-Through Augmented Reality in IoT-Based Environments. In *2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*. 40–43. doi:10.1109/ICEIB53692.2021.9686374

[43] Paul MILGRAM and Fumio KISHINO. 1994. A Taxonomy of Mixed Reality Visual Displays. (1994). https://search.ieice.org/bin/summary.php?id=e77-d_12_1321 Publisher: The Institute of Electronics, Information and Communication Engineers.

[44] Ayaskanta Mishra, Sayan Karmakar, Ankush Bose, and Ankita Dutta. 2020. Design and Development of IoT-based Latency-Optimized Augmented Reality Framework in Home Automation and Telemetry for Smart Lifestyle. *Journal of Reliable Intelligent Environments* 6, 3 (Sept. 2020), 169–187. doi:10.1007/s40860-020-00106-1

[45] Dimitris Mourtzis, Vasileios Siatras, and John Angelopoulos. 2020. Real-Time Remote Maintenance Support Based on Augmented Reality (AR). 10, 5 (2020), 1855. doi:10.3390/app10051855 Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[46] Dimitris Mourtzis, Vasilios Zogopoulos, and Fotini Xanthi. 2019. Augmented reality application to support the assembly of highly customized products and to adapt to production re-scheduling. 105, 9 (2019), 3899–3910. doi:10.1007/s00170-019-03941-6

[47] Fabio Paternò. 2022. End-User Creation and Control of Daily Automations in Intelligent Environments. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, 1–3. doi:10.1145/3491101.3503752

[48] Francesco Porpiglia, Enrico Checcucci, Daniele Amparore, Federico Piramide, Gabriele Volpi, Stefano Granato, Paolo Verri, Matteo Manfredi, Andrea Bellin, Pietro Piazzolla, Riccardo Autorino, Ivano Morra, Cristian Fiori, and Alex Mottrie. 2020. Three-dimensional Augmented Reality Robot-assisted Partial Nephrectomy in Case of Complex

Tumours (PADUA ≥ 10): A New Intraoperative Tool Overcoming the Ultrasound Guidance. 78, 2 (2020), 229–238. doi:10.1016/j.eururo.2019.11.024

[49] Jannish A. Purmaissur, Praveer Towakel, Shivanand P. Guness, Amar Seeam, and Xavier A. Bellekens. 2018. Augmented-Reality Computer-Vision Assisted Disaggregated Energy Monitoring and IoT Control Platform. In *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)* (2018-09). 1–6. doi:10.1109/ICONIC.2018.8601199

[50] Louis B. Rosenberg. 1993. Virtual fixtures as tools to enhance operator performance in telepresence environments. In *Telemanipulator Technology and Space Telerobotics* (1993-12-21), Vol. 2057. SPIE, 10–21. doi:10.1117/12.164901

[51] Mohammad Sahal, Emma Dryden, Mali Halac, Sara Feldman, Terry Heiman-Patterson, and Hasan Ayaz. 2021. Augmented Reality Integrated Brain Computer Interface for Smart Home Control. In *Advances in Neuroergonomics and Cognitive Engineering* (Cham, 2021), Hasan Ayaz, Umer Asgher, and Lucas Paletta (Eds.). Springer International Publishing, 89–97. doi:10.1007/978-3-030-80285-1_11

[52] Angela Sanguinetti, Beth Karlin, and Rebecca Ford. 2018. Understanding the Path to Smart Home Adoption: Segmenting and Describing Consumers across the Innovation-Decision Process. *Energy Research & Social Science* 46 (Dec. 2018), 274–283. doi:10.1016/j.erss.2018.08.002

[53] Ronny Seiger, Romina Kühn, Mandy Korzetz, and Uwe Aßmann. 2021. HoloFlows: modelling of processes for the Internet of Things in mixed reality. 20, 5 (2021), 1465–1489. doi:10.1007/s10270-020-00859-6

[54] Erich Stark, Erik Kučera, Oto Haffner, Peter Drahoš, and Roman Leskovský. 2020. Using Augmented Reality and Internet of Things for Control and Monitoring of Mechatronic Devices. 9, 8 (2020), 1272. doi:10.3390/electronics9081272 Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[55] Panagiotis Stavropoulos, Lydia Athanasopoulou, Alexios Papacharalampopoulos, Ioannis Kanellopoulos, and Dimitris Mourtzis. 2022. Augmented Reality tool for the facilitation of the UAV flight inspection process during aircraft de-icing as a maintenance procedure. 55, 10 (2022), 976–982. doi:10.1016/j.ifacol.2022.09.480

[56] S Sureshkumar, C P Agash, S Ramya, R Kaviyaraj, and S Elanchezhiyan. 2021. Augmented Reality with Internet of Things. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)* (2021-03). 1426–1430. doi:10.1109/ICAIS50930.2021.9395941

[57] Markus Tatzgern. 2015. Situated Visualization in Augmented Reality.

[58] Athanasios Vourvopoulos, Octavio Marin-Pardo, Meghan Neureither, David Saldana, Esther Jahng, and Sook-Lei Liew. 2019. Multimodal Head-Mounted Virtual-Reality Brain-Computer Interface for Stroke Rehabilitation. In *Virtual, Augmented and Mixed Reality. Multimodal Interaction* (Cham, 2019), Jessie Y.C. Chen and Gino Fragomeni (Eds.). Springer International Publishing, 165–179. doi:10.1007/978-3-030-21607-8_13

[59] Tao Wang, View Profile, Kangkang Zhang, View Profile, Wei Chen, View Profile, Wensheng Dou, View Profile, Jiaxin Zhu, View Profile, Jun Wei, View Profile, Tao Huang, and View Profile. 2022. Understanding device integration bugs in smart home system. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. 429–441. doi:10.1145/3533767.3534365

[60] X. Wang, A. W. W. Yew, S. K. Ong, and A. Y. C. Nee. 2020. Enhancing smart shop floor management with ubiquitous augmented reality. (2020). https://www.tandfonline.com/doi/full/10.1080/00207543.2019.1629667 Publisher: Taylor & Francis.

[61] Sean White, View Profile, Steven Feiner, and View Profile. 2009. SiteLens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1117–1120. doi:10.1145/1518701.1518871

[62] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. 2017. Embedded Data Representations. 23, 1 (2017), 461–470. doi:10.1109/TVCG.2016.2598608

[63] Lefan Zhang, Cyrus Zhou, Michael L. Littman, Blase Ur, and Shan Lu. 2023. Helping Users Debug Trigger-Action Programs. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4 (Jan. 2023), 196:1–196:32. doi:10.1145/3569506

[64] Mengya Zheng, Xingyu Pan, Nestor Velasco Bermeo, Rosemary J. Thomas, David Coyle, Gregory M. P. O'hare, and Abraham G. Campbell. 2022. STARE: Augmented Reality Data Visualization for Explainable Decision Support in Smart Environments. *IEEE Access* 10 (2022), 29543–29557. doi:10.1109/ACCESS.2022.3156697

[65] Xiaoyan Zhou, Benjamin Lee, Francisco R. Ortega, Anil Ufuk Batmaz, and Yalong Yang. 2024. Lights, Headset, Tablet, Action: Exploring the Use of Hybrid User Interfaces for Immersive Situated Analytics. *Proc. ACM Hum.-Comput. Interact.* 8, ISS (Oct. 2024), 547:517–547:539. doi:10.1145/3698147