# MARIoT: a Framework for Creating Customizable IoT Applications with Mobile Augmented Reality

Meral Kuyucu
*Department of Computer Engineering*
*Istanbul Technical University*
email: korkmazmer@itu.edu.tr

Gökhan İnce
*Department of Computer Engineering*
*Istanbul Technical University*
email: gokhan.ince@itu.edu.tr

***Abstract*—The programming overhead related to the implementation of intelligent environments is a major obstacle keeping non-technical technology enthusiasts from harnessing the advantages of bringing such technologies into their lives. Commercially available alternatives lack the customization angle that makes these technologies desirable in the first place. In this study, we propose a Low-Code/No-Code (LCNC) Mobile Augmented Reality for Internet of Things (MARIoT) framework that alleviates the requirement of programming expertise otherwise necessary in both the Internet of Things (IoT) and Augmented Reality (AR) fields. With the MARIoT framework, users can create customized AR interfaces to be used for interacting with devices and sensors in a smart home environment. The usability of this framework with participants of different educational and programming backgrounds was tested. Experimental results show that the proposed framework facilitates the intuitive integration of the two technologies.**

***Index Terms*—AR; IoT; Smart Home; LCNC.**

## I. Introduction

Big data, Artificial Intelligence (AI), Augmented Reality (AR), the Internet of Things (IoT), cloud computing, and autonomous robots are among today's cutting-edge technologies. Brought together, these technologies can make smart decisions instead of people in various environments including households, factories, hospitals, transportation, and cities amongst others [1].

AR is the superimposition of digital information onto the real-world view of users to create a richer, compound representation of the surrounding environment [2]. IoT can be loosely described as the concept of connecting everything to the internet [3]. In this sense, the term "everything" encompasses things that can collect data (i.e., sensors), things that can perform actions based on input commands (i.e., home appliances), or things that can do both.

Two technologies can be fused if they are cooperative and complement each other [4]. The fusion of cooperative and complementary technologies often creates an opportunity to harness the advantages of both technologies and mitigate the weaknesses which arise when using the technologies separately. In this study, we motivate the fusion of AR and IoT.

The IoT often exists in the form of ubiquitous devices in our environment [3]. That is, they exist in 3D space. In such a data-driven environment, users must be able to visualize data and communicate with devices seamlessly and intuitively. We believe that in order to achieve this intuition, the interaction surface must also be in 3D.

To better illustrate, consider a scenario in which a user sitting in the living room wishes to observe the setting of the thermostat across the room. With AR-enabled IoT, it will be possible for him to simply point his mobile phone towards the IoT device in question. The context-aware AR interface will then adapt and present the user with a thermostat reading and controller. Using the virtual slider controller, he can adjust the temperature and turn his mobile device towards the light. The context-aware interface will then adapt again and display the light control options.

This seamless context-aware integration of AR and IoT can potentially revolutionize the already progressive concept of smart homes. However, many challenges lie in the way of achievement. Often, people do not have the necessary knowledge and skills to create end-to-end personalized solutions. Moreover, commercial solutions are not customized to fit the needs of different individuals, defeating the purpose of having a customized smart home.

In this study, a Low-Code/No-Code (LCNC) Mobile AR for IoT (MARIoT) framework for customized smart home applications is proposed. We assemble a framework consisting of an IoT network, an AR interface template generator, a dynamic mobile AR application that creates an interface based on the template in real-time, and a messaging protocol between the AR interface and IoT sensors and devices. The main goal of the framework is to provide necessary abstractions so that users can build customized applications by taking advantage of technologies that require coding knowledge without actually having to write code. This framework was tested to answer the following research questions:

1) Will tech-savvy but not necessarily code-savvy users find the suggested framework helpful when creating customized AR-enabled smart environments?
2) Will users find an AR interface for interacting with IoT devices intuitive?

There are three main contributions of this study. First, an end-to-end framework integrating AR and IoT using open source technologies is presented. Second, an LCNC framework that provides users with necessary abstractions so that they can create a personalized smart home application is established.

 ISBN: 978-1-61208-870-9 

Lastly, this study assesses the usability of the framework using an application generated with this framework.

The rest of this paper is organized as follows. Section II presents a review of the literature. The design of the proposed framework is given in Section III. Experiments and results are discussed in Section IV. Finally, Section V presents concluding remarks.

## II. Literature Review

Numerous studies have focused on applications generated by the fusion of AR and IoT. Studies have targeted medical assistance [5], energy management [6], technical instruction [7], crop monitoring [8], machine fault diagnostics [9], and even military applications [10]. The attitude toward domestic AR was investigated in a study conducted by Knierim et al. [11]. In light of semi-structured interviews at the end of a technology probe, they conclude that participants are interested in using AR within a domestic environment for its ability to enhance perception, provide on-demand information and assistance, and augment devices.

Especially in recent years, many implementation-level studies have been published in this field. These works present direct applications integrating AR and IoT in varying domains. Jang and Bednarz suggest head-mounted AR for visualizing data from various sensors. The output of their work is an application that communicates data from sensors in the network to the AR application [12]. Ankireddy et al. propose a mobile AR application promoting interaction with devices. They also design and implement a low-cost plug-and-play smart glass that can be used to control devices [13]. Fredericks et al. argue that energy consumption is inherently invisible in nature and can be made visible by employing augmented reality to raise awareness and promote energy-conserving behaviors [14]. Another study by Mahroo and colleagues investigates the possibility of communicating with household devices using Microsoft HoloLens [15]. Blanco-Novoa and colleagues set forth an open-source AR-IoT framework facilitating the utilization of AR and IoT in real-time [16]. Wright et al. suggest a cross-platform open-source mixed reality framework capable of communicating with headless IoT devices [17]. Marques et al. propose a context-aware AR application to configure and create uninterrupted management and control of smart home sensors and devices [18]. Mishra et al. describe an AR framework for home automation and telemetry application with IoT. They implement a prototype with which users can control household appliances such as a fan or a light [19]. The common shortcoming of all of the aforementioned studies is that no investigation of usability has been conducted.

Studies that assess the usability of prototype applications also exist in literature. Ullah et al. suggested a prototype that uses AR to control home appliances [20]. Test results suggest that most users would consider using AR interfaces. Jo and Kim introduce a framework for visualizing IoT data with augmented reality and conduct a usability test with 10 participants [21]. Experimental results show that most participants preferred AR-based interaction over Graphical User Interface (GUI) based interaction. Putze et al. design a head-mounted AR Brain-Computer Interface (BCI) which can be controlled by the user's eye movements [22]. Results of their usability tests also show that users are highly in favor of AR interfaces over standard GUI-based modes of interaction. In these studies, users were given a prefabricated interface capable of controlling a fixed set of IoT elements. Users were expected to use the interface and assess usability. We are curious about what happens when the user wishes to create an interface that can control a set of user-determined IoT elements.

Most of the studies found in this field are at the implementation level. Studies that do conduct usability tests do not aim for LCNC solutions. Our study differs from these works because we suggest a framework that can be used by non-programming users who wish to create personalized applications. A smart home was selected as the domain of our study because it appeals to many potential users. Yet, the framework suggested in this study can be generalized to a variety of disciplines.

## III. An AR-IoT Enabled Interactive Framework for Smart Home Use

In this section, we present our LCNC framework for creating personalized AR-IoT enabled applications called MARIoT. Figure 1 is an illustration of the system.

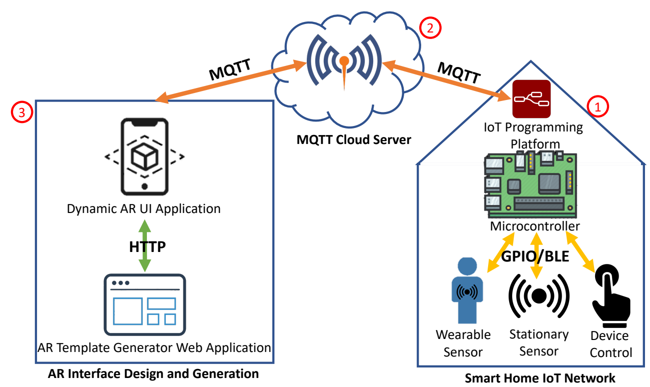



Fig. 1: Overview of the proposed MARIoT framework.

The main components of the system can be divided into three. A smart home IoT network (Figure 1①) consists of different types of sensors and actuators connected to a microcontroller via General Purpose Input Output (GPIO) pins of the controller or Bluetooth Low Energy (BLE). Sensors (i.e., photoresistor) collect data which can be displayed to the user, analyzed, or used in a machine learning application to make smart decisions. For the purposes of this study, we have chosen the name actuator for devices to which commands can be issued by the user via an interface (i.e., a desk lamp). The microcontroller is programmed to send collected data to a cloud server and listen to the server for device input commands. This is done utilizing a visual (flow-based) development tool. A set of configurable generic flows helps the user easily bring up the IoT end of the system. The cloud

 ISBN: 978-1-61208-870-9 

server (Figure 1②) relays messages between the AR interface and the IoT network. The third and final constituent of the system is the AR interface template generator and mobile AR application (Figure 1③). First, the template generator web application can be utilized to create a template for an AR interface. The output is utilized by the mobile AR application to dynamically and automatically create the user-designed AR interface at runtime. Data from IoT sensors will be visualized with text labels and users will be able to input a number of discrete-valued input commands via several GUI buttons. These components can be utilized to create personalized AR applications for interacting with smart homes.

### A. Design of MARIoT framework

Figure 2 illustrates a layered architecture of the proposed framework. The IoT elements, cloud server, and AR Engine lie in the lowest layer. These components all require the user to have programming expertise. The second layer consists of the abstractions we added to the base layer so that users can benefit from the base layer technologies without having to write code. Customizable flows enable the user to create IoT scenarios and the template generator assists the user in designing a personalized AR interface to interact with IoT sensors and devices. User-generated applications lie at the top layer of the architecture.

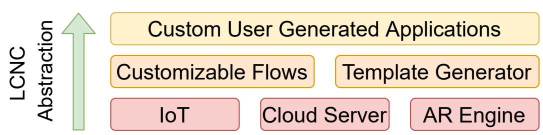



Fig. 2: Layered architecture of the proposed framework.

*1) Customizable Task Flow Generator for IoT Devices:* This component is responsible for IoT management. Processes in the IoT system are carried out in the form of flows, modular sequential ways of representing more complex tasks.

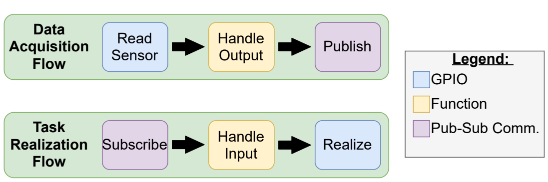



Fig. 3: Set of customizable task flows.

In order to manage IoT processes, we recommend two generic customizable task flows. To configure the IoT system, users can simply import these flows into their workspaces and configure only the device and server-specific aspects. The set of essential flows is illustrated in Figure 3 and detailed as follows:

- **Data Acquisition Flow:** The flow is triggered by a sensor reading which occurs when there is a change in the environment. In the second step, the digital output received from the sensor is converted to a message determined by the user. For example, a photoresistor reading of "1" can be converted to "Light is on". The user-defined message is then combined with a timestamp to indicate the freshness of the data. The third step consists of sending this data to the cloud server.
- **Task Realization Flow:** The IoT device listens to the server for a message from the AR interface. The flow is triggered when a message arrives. In the second step, the message content of the command is parsed. The necessary control signals are applied to the device in the third step. For example, an "OFF" message sent to a desk lamp can be converted to a "0" digital signal to turn off the lamp.

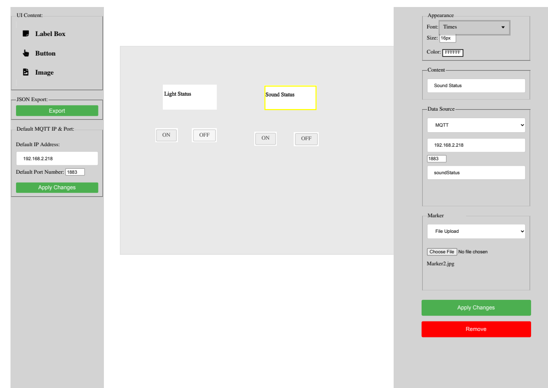



Fig. 4: Template generator web application.

*2) Template Generator for Designing Augmented Reality Applications:* To facilitate the task of creating an AR user interface for non-programmer users, we introduce a template generator. This web application allows users to drag-and-drop User Interface (UI) elements onto a canvas to design an interface. These elements can be customized in terms of appearance, content, and data source. Additionally, image markers can be added for each UI element to support context awareness.

The template generator application with an example template can be seen in Figure 4. The sidebar on the left contains the UI elements that can be placed onto the canvas. These elements include label boxes, images, and buttons. When an element on the canvas is clicked on, it becomes active and the edit form appears from the right. The user can modify the appearance (i.e., font size and color), content (i.e., text placeholder), data source (i.e., the server it connects to for sending and receiving data), as well as the marker image for the UI element using this form. Completed templates can be exported to be used by the mobile application to dynamically create an AR interface.

*3) Pub-Sub Messaging Communication Protocol Between IoT Task Flow and AR Interface:* Perhaps the backbone of the suggested framework is the communication of elements in the system. We chose to implement communication using the Message Queuing Telemetry Transport (MQTT) protocol. MQTT is a lightweight, publish-subscribe network protocol

 ISBN: 978-1-61208-870-9 

that transports messages between devices. Two types of clients exist in a pub-sub network. The first of these is the publishing client, which sends messages under a topic. Secondly, the subscribers listen for messages under specific topics.

In our system, sensors act as publishers, collecting environmental data and publishing it to the network. Actuators subscribe to user commands and are controlled based on incoming commands. AR interface elements that display data (i.e., labels) are subscribers. They subscribe to data coming from the IoT sensors and update the displayed data in real-time. AR interface elements that take inputs from the user (i.e., buttons) publish user commands.

### B. Implementation of MARIoT framework

In this study, a photoresistor and microphone were chosen as the sensors and a desk lamp and desk fan were chosen as the actuators of an IoT network. A Raspberry Pi computer was preferred as the microcontroller to which the sensors and actuators were connected. We adopted Node-RED, a flow-based visual development tool with a web-based flow editor for controlling IoT devices. The customizable generic task flows we propose were designed using Node-RED. The flows were composed so that only the GPIO information, output message contents, and server-specific information must be configured by the user. Mosquitto MQTT was chosen for the messaging protocol due to its speed and significantly low overhead when compared to the standard HTTP protocol. The AR template generator is a web application created using HTML and JavaScript (JQuery UI). The AR interface is created by a mobile application employing the Vuforia AR library. This application takes the output of the template generator and creates a corresponding mobile AR interface dynamically at runtime.

## IV. Experiments and Results

In this section, we illustrate the experimentation procedure in detail. The experimental results and a discussion of the outcomes follows.

### A. Experimental Framework

We conducted usability tests for the proposed framework at Istanbul Technical Univerity's UX Lab. The experiment setup is depicted in Figure 5. Two image markers and the selected set of devices and sensors are placed on the table. Additionally, a computer to be used for IoT configuration and AR interface design is provided.

### B. Usability Tests

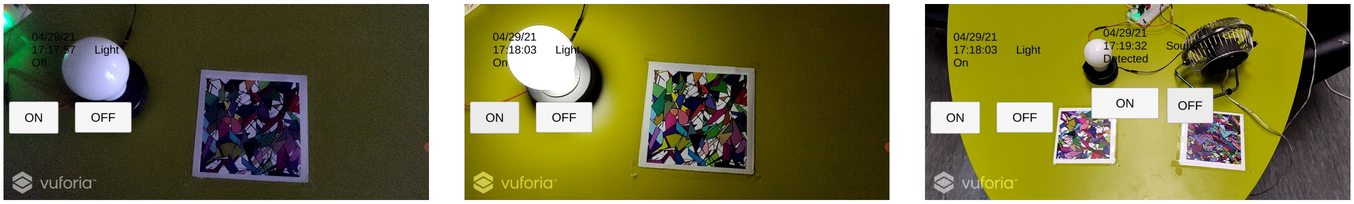
*1) Participant Demographics:* Due to the COVID-19 pandemic, access to participants was rather limited. After an initial acquaintance with the facilitator, potential participants were asked to complete a screening survey. Participants were selected based on a combination of AR experience, IoT experience, smart home experience, and daily time spent in front of a computer or mobile device. We preferred participants who used computers and mobile devices for a good part of the day and who were familiar with one or more of the technologies utilized in the study. Participants were familiar with IoT devices such as smart locks, robot vacuums, smart scales, and security cameras. They had some AR experience with games, simulators, and applications that employ camera filters.

After the screening procedure, usability tests were conducted with 9 participants (3 male, 6 female) ranging from 18 to 26 years in age. Selected participants had varying educational backgrounds such as computer engineering, textile engineering, and genetics engineering. All selected participants were either college graduates, or students. They were required to have acquired some level of higher learning due to the concentration necessary to complete the experiment. Of the 9 participants, 4 were undergraduate students and 1 was a graduate student in the department of computer engineering. The rest of the participants were from different disciplines of engineering (i.e., textile and genetics engineering). Although they had varying levels of programming knowledge, none had coding experience with Node-RED, MQTT, or AR prior to the experiment.

*2) Test Materials:* Throughout the test, the information provided to the participant by the facilitator was read to the participants from a script to maintain consistency throughout experiments. The script can also be considered a guide for the facilitator throughout the experiment. Moreover, a carefully constructed script can remove otherwise unnoticed bias from the words of the facilitator.

*3) Pre-Training Materials:* We prepared a video presentation that defines AR and IoT and discusses how the two technologies will be used in the scope of this study. The video was designed to provide information to a person who knows nothing about these two technologies. First, the primary definitions of AR and IoT were given. Then, the motivation of using these two technologies together in a smart home domain was built. Afterward, each component of the system was explained to the participants so that the connection between AR UI elements and IoT sensors and devices was clear to the participant. A discussion of Figures 1 and 2 have been presented in the training video. When the participants finished watching the video, they were permitted to ask questions about the system before resuming the experiment. We also prepared two pre-training videos demonstrating the process of importing and customizing the Node-RED task flows.

*4) Test Procedure and Metrics:* We conducted two pilot studies with expert users. The test materials (script and videos) were revised based on the insight gained from the pilot studies. Due to the lengthy and incremental nature of the system, we decided to fix the time to complete a task. The maximum amount of time to complete the experiment tasks was determined based on the outputs of the pilot studies. Allowing participants to roam prior to completing the test may cause them to lose track and interest in the task at hand. Users were given opportunities to ask questions prior to the test. During the test, users were not provided with assistance. After completing the revisions, usability tests were conducted




(a) Task 1: Turning on the light. (b) Task 2: Observing sensor output. (c) Task 3: Visualizing both markers.

Fig. 5: AR application generated automatically based on template designed by the user.

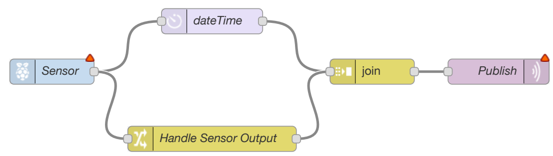
with selected participants.

Our tests were designed so that users can assess both the usability of the end-to-end framework as well as the usability of an AR interface for interacting with IoT sensors and devices. Throughout the tests, users interact with the framework in three consequent tasks:

- Task 1) Importing and customizing two generic flows:
  - Task 1.1) Data Acquisition Flow
  - Task 1.2) Task Realization Flow
- Task 2) Using the template generator to create a template of a mobile AR interface:
  - Task 2.1) Configuration of server settings
  - Task 2.2) Adding labels to template
  - Task 2.3) Adding buttons to template
- Task 3) Using the dynamically and automatically created mobile AR interface:
  - Task 3.1) Observing sensor output
  - Task 3.2) Controlling a device

First, users were briefly informed about the overall system. Afterward, a training and practice session for Node-RED was provided. In the pre-training session, participants were asked to watch a training video demonstrating the usage of the customizable generic task flows. Then, they were given a chance to practice importing and customizing a Data Acquisition flow and a Task Realization flow in the training session. Because Node-RED is an existing development tool, we felt it necessary to provide training because we are not interested in the users' assessment of Node-RED. Rather, we would like to find out how well the entire system consisting of the three components in Figure 1 works as a whole. We did not train the users with the template generator or the mobile AR interface because we are interested in how intuitive users find them to be. Users interacted with the template generator and AR interface for the first time during the test.

The first task (Task 1) constitutes the configuration of the sensors and actuators. The customizable generic task flows users interacted with are given in Figure 6. In Task 1.1, users were asked to import and configure a Data Acquisition flow for a sensor as shown in Figure 6(a). The "Sensor Data" node should be configured so that the pin to which the sensor is connected is read from. Then, the "Handle Sensor Output" node should be configured with the messages to be displayed depending on digital sensor outputs. Finally, the "Publish" node should be configured with the server details




(a) Data Acquisition Flow


(b) Task Realization Flow

Fig. 6: Customizable task flows.

and a topic to publish messages under. In Task 1.2, users were requested to import and configure a Task Realization flow for an actuator. The Task Realization flow is given in Figure 6(b). First, the "Subscribe" node should be configured with server details and a topic to listen for messages. For this flow, the "Handle Message" node does not need modification because currently, the system only supports "ON" and "OFF". Lastly, the "Device" node should be configured so that the pin to which the device is connected is sent a signal.

The objective of Task 2 was to design a template for an AR interface using the template generator. In Task 2.1, users were instructed to configure the server details to which the AR interface elements will be connecting to. This way users do not have to enter sensor details for each dropped UI element. Only the topic should be added. In Task 2.2, users were asked to place two labels onto the canvas. The labels should be configured to subscribe to the topics assigned to the sensors in Task 1.1. In Task 2.3, users were asked to add 4 buttons to the canvas and to configure the buttons as On/Off buttons for two actuators. The buttons should be configured to publish to the topic to which the actuators were subscribed in Task 1.2. A marker image was uploaded for each UI element. An element becomes visible when its assigned marker is in camera view. Once the user finishes designing a template, it can be exported.

The user-designed template from Task 2 is automatically generated by the mobile AR application in Task 3. In this final task, users are asked to hold the mobile device against different markers, make sensor observations, and control devices with buttons.

The metrics used in evaluation can be seen in Table I. We requested users to complete an After Scenario Questionnaire (ASQ) after tasks 1 and 2. Finally, users were invited to

TABLE I: SELECTED EVALUATION METRICS

| Usability Dimensions | Evaluation Metrics | Units | Investigation Techniques |
|---|---|---|---|
| *Effectiveness* | Completion Rate | Percentage (%) | Direct Observation |
| | Number of Errors | Number | |
| *Efficiency* | Task Completion Time | Seconds | Direct Observation |
| *Satisfaction* | After Scenario Questionnaire | 1-5 Likert Scale | Questionnaire |
| | System Usability Scale | | |

evaluate the end-to-end system with a System Usability Scale (SUS).

*5) Walkthrough of Test Procedure:* Throughout the IoT training and test phases, four flows were configured for the two sensors and two actuators in the experimental setup. Flows for the photoresistor and desk lamp were configured in the IoT training session. Flows for the microphone and desk fan were configured during Task 1.

In Task 2, the template in Figure 4 was designed by the users. The "Light Status" label was configured to subscribe to the same topic that the photoresistor is publishing to. This way, the photoresistor readings will be displayed on the label in real-time. The button pair under the label are publishing On/Off messages to the same topic assigned to the desk lamp. The photoresistor and the buttons for the desk lamp shared a marker. The microphone and the buttons for the desk fan shared a second marker.

In Task 3, participants were asked to launch and use the application they designed. Figure 5 illustrates the mobile AR interface resulting from the template generated in Figure 4. First, the lights are turned off, and the user is requested to hold the mobile device against a marker and observe the sensor status (Figure 5(a)). When the marker related to the desk lamp and light sensor is in camera view, the controls of the desk lamp and the status of the light are displayed. The user is then instructed to turn the desk lamp on and observe the change in the sensor output (Figure 5(b)). After the button press, the "ON" message is published. The subscriber (desk lamp) receives the message and the lamp is turned on (Figure 5(b)). The user is then instructed to hold the mobile device against both markers (Figure 5(c)). The controls for both devices as well as the information from both sensors become visible.

Once participants were finished with all three tasks of the experiment, they were given the opportunity to roam. Some participants chose to change the layout and appearance of the AR interface. Other participants experimented with changing the marker images of different UI elements.

## C. Test Results

The results of Tasks 1 and 2 can be seen in Table II. Participants were given 180 seconds to complete Task 1.1. Most users were able to complete the task with little training. In Task 1.1, two participants were unsure about choosing a topic to publish messages under and took time to look back into the documentation provided during the training phase. The time given to complete Task 1.2 was 120 seconds. Only one participant was unable to complete this task within 120 seconds. Four participants were unsure about connecting to the same server (from Task 1.1) as a publisher. The overall satisfaction with Task 1 as given by the ASQ is 4.81/5.00.

TABLE II: TASK 1 AND 2 RESULTS

| Evaluation Metrics | Tasks | | | | |
|---|---|---|---|---|---|
| | Task 1 | | Task 2 | | |
| | Task 1.1 | Task 1.2 | Task 2.1 | Task 2.2 | Task 2.3 |
| *Avg. Completion Rate* | 89% | 89% | 89% | 67% | 100% |
| *Avg. Number of Errors* | 0.23 (± 0.44) | 0.45 (± 0.53) | 0.12 (± 0.33) | 0.56 (± 0.73) | 0.34 (± 0.71) |
| *Avg. Task Time (sec.)* | 119.56 (± 40.46) | 75 (± 38.39) | 28.44 (± 14.3) | 182.89 (± 45.45) | 211.67 (± 31.34) |

Because Task 2.1 required very few steps to complete, participants were given 45 seconds to complete the task. This task was fairly simple and only one participant misunderstood the task initially. Participants were expected to complete Task 1.2 in 180 seconds. Three participants were unable to finish in time and the average time was slightly higher than the allotted time. The participants who did not finish on time failed to distinguish the topic and the content of a label. Because Task 2.3 required the insertion and customization of four buttons, users were given 420 seconds to complete it. Most participants did not even need half of the allotted time to complete the task. The errors were due to confusion between the topic and the content of a button. Even though users saw the web application for the first time and no training was provided for this task, most users were able to complete the task within the given amount of time. Unsuccessful users were allowed to continue if they wished. Although they were unsuccessful, these users were also eventually able to complete the tasks with extra time. The overall satisfaction with Task 2 was 4.74/5.00.

When asked to hold the mobile device against a marker in Task 3, participants saw that the related UI elements appeared. They were interested in controlling devices–most participants attempted to turn on the devices without being asked to do so. Participants tested turning the light on and off and observing the changes in the state of the photoresistor. The sound created when the fan was turned on triggered the state of the microphone and was noted by the participants. After using the application, most participants commented that they found the concept of topics and pub-sub messaging more intuitive. Participants were also keen to notice that the interface was updated as the mobile device was turned to face different markers. The SUS summarizes the results of Task 3. In general, users indicated that they were satisfied with the system, however, some did think that the system required users to learn. The overall SUS score was 81.9 (A).

## D. Discussion

During unstructured conversations with the participants, 6 of the 9 participants indicated that they would like to use these technologies in their homes without directly being asked the question. Participants 4 and 7 mentioned that they felt accomplished and satisfied after completing the test. Most participants were excited to suggest scenarios in which the proposed framework can be used. For example, participant 2 commented that she would like to use this system to monitor and control her kitchen appliances such as her coffee machine. Participant 2 also mentioned that this would not only be useful in the home but also in the workplace. Participant 8 commented on the potentials of integrating wearable technologies into this framework for an even richer, more continuous interaction experience. He suggested integrating textile-based sensors as wearable components of the system. Participant 6 mentioned that different methods of visualizing data such as graphs could be integrated to better present information to the user. Participant 7 mentioned that even though this was completely new material, she was very comfortable and found it intuitive even. She mentioned the concept of pub-sub messaging was similar to a biological process. Participant 9 stated that she was hesitant to participate in the experiment because she did not think she was tech-savvy enough but after testing the system, she found it to be simple to use.

An important angle to consider when evaluating the results of this experiment is the performance comparison of participants with computer engineering training and participants who come from different backgrounds. Table III emphasizes these differences. For each metric, the "CE" column presents the results for participants with computer engineering experience. The "NCE" column depicts the results of participants who do not have computer engineering knowledge. It can be observed from Table III that the CE group was more successful in Task 1.1 and the NCE group outperformed in Task 1.2. While the CE group had a higher completion rate in Task 2.1, the NCE group completed the task in fewer seconds. That is to say, both groups had similar performance throughout the experiment.

TABLE III: PERFORMANCE COMPARISON

| | **Completion Rate** | | **Completion Time** | | **Num. of Errors** | |
|---|---|---|---|---|---|---|
| **Tasks** | **CE** | **NCE** | **CE** | **NCE** | **CE** | **NCE** |
| *1.1* | 100% | 75% | 103.2 | 114.65 | 0 | 2 |
| *1.2* | 80% | 100% | 76.5 | 56.75 | 4 | 0 |
| *2.1* | 100% | 75% | 29 | 18 | 0 | 1 |
| *2.2* | 80% | 50% | 150.25 | 174 | 2 | 3 |
| *2.3* | 100% | 100% | 193.2 | 234.75 | 0 | 3 |

While there are no major usability problems with the suggested framework, the wording in the template generator application can be modified to help users distinguish between the content and the topic of a UI element. The targeted audience for this study was a tech-savvy population. This is why tests were conducted with younger individuals. Because there is a significant amount of cognitive load involved with configuring the system end-to-end, elder adults who may not be as computer literate may have difficulties with this system.

In the future, we plan on re-designing the template generator so that it can collect information regarding the flows on Node-RED and make suggestions to the user to reduce this cognitive load. This way, users do not have to remember the topic they entered in the IoT configuration phase. Once the user determines a topic in Node-RED, the same topic can be presented to the user in a drop-down menu refreshed in real-time. Numerous functionalities can be added to the architecture so that users can take complete advantage of the system. The set of supported UI elements available to the user are to be expanded to include various types of graphs, and different methods for inputs (i.e., slider input). Next, support for head-mounted gears will be integrated. Finally, multiple methods of context-awareness will be added to the system so that the user-defined apps can be context-aware in a way that is most convenient to the user.

## V. CONCLUSION

In our study, we introduced an LCNC framework that detaches coding from the process of creating personalized AR-IoT enabled applications. This way, people with little knowledge regarding coding can easily set up and control a custom smart environment. We conducted usability experiments and found that even participants with no programming background can easily create an end-to-end AR-enabled IoT system using our framework. In the light of experimental results, it can be concluded that tech-savvy users were able to use the suggested framework to communicate with IoT sensors and devices using mobile AR. Moreover, participants were excited to suggest ways in which they can use mobile AR in their own homes to interact with their devices.

## ACKNOWLEDGMENT

In this work, Meral Kuyucu has been supported by the Turkcell Academia-ITU Research Grant Program.

## REFERENCES

[1] G. Lampropoulos, K. Siakas, and T. Anastasiadis, "Internet of things in the context of industry 4.0: An overview," *International Journal of Entrepreneurial Knowledge*, vol. 7, pp. 4–19, 06 2019.

[2] J. Carmigniani and B. Furht, *Augmented Reality: An Overview*, 07 2011, pp. 3–46.

[3] K. Patel, S. Patel, P. Scholar, and C. Salazar, "Internet of things-iot: Definition, characteristics, architecture, enabling technologies, application & future challenges," 05 2016.

[4] J. Tidd and T. Sammut-Bonnici, *Technology Fusion*, 2015.

[5] F. Ghorbani, M. Kia, M. Delrobaei, and Q. Rahman, "Evaluating the possibility of integrating augmented reality and internet of things technologies to help patients with alzheimer's disease," in *2019 26th National and 4th International Iranian Conference on Biomedical Engineering (ICBME)*, 2019, pp. 139–144.

[6] K. Cho, H. Jang, L. W. Park, S. Kim, and S. Park, "Energy management system based on augmented reality for human-computer interaction in a smart city," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, pp. 1–3.

[7] P. Hořejší, K. Novikov, and M. Šimon, "A smart factory in a smart city: Virtual and augmented reality in a smart assembly line," *IEEE Access*, vol. 8, pp. 94 330–94 340, 2020.

[8] P. Phupattanasilp and S.-R. Tong, "Augmented reality in the integrative internet of things (ar-iot): Application for precision farming," *Sustainability*, vol. 11, p. 2658, 05 2019.

[9] V. Rajan, N. Sobhana, and R. Jayakrishnan, "Machine fault diagnostics and condition monitoring using augmented reality and iot," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 910–914.

[10] S. Jalui, T. Hait, T. Hathi, and S. Ghosh, “Advanced military helmet aided with wireless live video transmission, sensor integration and augmented reality headset,” in *2019 International Conference on Communication and Electronics Systems (ICCES)*, 2019, pp. 123–127.

[11] P. Knierim, P. W. Woundefinedniak, Y. Abdelrahman, and A. Schmidt, “Exploring the potential of augmented reality in domestic environments,” in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–12. [Online]. Available: https://doi.org/10.1145/3338286.3340142

[12] J. Jang and T. Bednarz, “Holosensor for smart home, health, entertainment,” in *ACM SIGGRAPH 2018 Appy Hour*, ser. SIGGRAPH ’18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3213779.3213786

[13] M. A. Ankireddy, A. Rajath, M. Ruthwik Ganesh, and M. Anuradha, “Augmented reality rendered for iot applications,” in *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1–4.

[14] A. D. Fredericks, Z. Fan, and S. I. Woolley, “Visualising the invisible: Augmented reality and virtual reality as persuasive technologies for energy feedback,” in *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2019, pp. 1209–1212.

[15] A. Mahroo, L. Greci, and M. Sacco, “Holohome: An augmented reality framework to manage the smart home,” in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis and P. Bourdot, Eds. Cham: Springer International Publishing, 2019, pp. 137–145.

[16] O. Blanco-Novoa, P. Fraga-Lamas, M. Vilar-Montesinos, and T. Fernández-Caramés, “Towards the internet of augmented things: An open-source framework to interconnect iot devices and augmented reality systems,” in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 42, no. 1, 11 2019, p. 50.

[17] D. Wright, R. Skaggs-Schellenberg, and S. Tayeb, “Networked mixed reality framework for the internet of things,” in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2020, pp. 1–7.

[18] B. Marques, P. Dias, J. Alves, E. Fonseca, and B. S. Santos, “Configuration and use of pervasive augmented reality interfaces in a smart home context: A prototype,” in *Human Systems Engineering and Design II*, T. Ahram, W. Karwowski, S. Pickl, and R. Taiar, Eds. Cham: Springer International Publishing, 2020, pp. 96–102.

[19] A. Mishra, S. Karmakar, A. Bose, and A. Dutta, “Design and development of iot-based latency-optimized augmented reality framework in home automation and telemetry for smart lifestyle,” *Journal of Reliable Intelligent Environments*, vol. 6, no. 3, pp. 169–187, 2020. [Online]. Available: https://doi.org/10.1007/s40860-020-00106-1

[20] A. M. Ullah, M. R. Islam, S. F. Aktar, and S. K. A. Hossain, “Remote-touch: Augmented reality based marker tracking for smart home control,” in *2012 15th International Conference on Computer and Information Technology (ICCIT)*, 2012, pp. 473–477.

[21] D. Jo and G. J. Kim, “Ariot: scalable augmented reality framework for interacting with internet of things appliances everywhere,” *IEEE Transactions on Consumer Electronics*, vol. 62, no. 3, pp. 334–340, 2016.

[22] F. Putze, D. Weiß, L.-M. Vortmann, and T. Schultz, “Augmented reality interface for smart home control using ssvep-bci and eye gaze,” in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 2812–2817.


 ISBN: 978-1-61208-870-9