MARIoT: An Authoring Framework for Creating IoT Applications with Mobile Augmented Reality

Meral Kuyucu, Gökhan İnce

User Experience Laboratory
Department of Computer Engineering
Istanbul Technical University (Turkey)

Contact email: korkmazmer@itu.edu.tr
Meral Kuyucu received her BSE and MS degrees from the Department of Computer Engineering at Istanbul Technical University (ITU) in 2017 and 2021 respectively. She received the ITU-Turkcell academic grant throughout her graduate studies. She will continue to pursue a PhD degree at ITU where she is currently working as a Research and Teaching Assistant at the User Experience Lab. She will continue to conduct research on AR-enabled IoT systems.
Introduction

- With the launch of the internet, components of the digital age such as AR and IoT took off.

Technology fusion:
  - Cooperative
  - Complementary
  [Tidd, 2013]

Harness advantages
Mitigate weaknesses

We propose the fusion of AR and IoT

IoT Exists in 3D Space
+ 
Need Intuitive Interaction

3D Surface of Interaction

Kuyucu | 3/34
Proposed System

- MARIoT: Mobile Augmented Reality for the Internet of Things
  - IoT Network
  - Communication Server
  - AR Authoring Tool and Application Generator

- Research Questions
  - Will tech-savvy but not necessarily code-savy users find the suggested framework helpful when creating customized AR enabled smart environments?
  - Will users find an AR interface for interacting with IoT devices intuitive?
Literature Review

• Many studies fuse AR and IoT at the implementation level.

• Most studies do not conduct usability tests [Marques et al., 2019].

• Studies that do, do not aim for Low Code/No Code (LCNC) solutions.

Although many studies in literature investigate End-User Development (EUD) for IoT and AR separately, to our knowledge, there was no research conducted on authoring tools for the integration of AR and IoT.
System Components of MARIoT

- Components:
  - Wearable Sensor
  - Stationary Sensor
  - Device Control
  - Microcontroller
  - IoT Programming Platform
  - Dynamic AR UI Application
  - HTTP
  - AR Template Generator Web Application
  - AR Interface Design and Generation

- Network:
  - MQTT Cloud Server
  - Dynamic AR UI Application
  - AR Template Generator Web Application
  - Smart Home IoT Network

- Protocols:
  - MQTT
  - HTTP
Architecture of MARIoT

- Custom User Generated Applications
- Customizable Task Flow Generator
- AR Template Generator
- IoT
- Cloud Server
- AR Engine
Customizable Task Flow Generator for IoT

**Data Acquisition Flow**
- Read Sensor
- Handle Output
- Publish

**Task Realization Flow**
- Subscribe
- Handle Input
- Realize

**Legend:**
- GPIO
- Function
- Pub-Sub Comm.
Template Generator for AR Applications
Pub-Sub Messaging Communication Protocol

- **Backbone of System = Communication of Elements**

- **MQTT**: Lightweight, pub-sub network protocol that transports messages among devices.
  - **Publisher**: Broadcasts messages with topics
  - **Subscriber**: Listens for messages with specific topics
Software

- Dynamic AR UI Generator
- AR Template Generator
- Customizable Task Flow Generator
- Pub-Sub Broker
Hardware

Actuators:
- Relay Switch
- Fan

Microcontroller:
- Raspberry Pi 3B+

Sensors:
- Microphone
- Photoresistor
- Photoresistor

Photoresistor
Pub-Sub Communication Architecture

- **Device ON Button**: Publish Topic: Device
- **Device OFF Button**: Publish Topic: Device
- **Sensor Data Label**: Subscribe Topic: Sensor
- **MQTT Broker**: Publish Topic: Sensor
- **Device**: Subscribe Topic: Device
Customizable Task Flow Generator

Data Acquisition Flow

Sensor → DateTime → Handle Sensor Output → join → Publish

Task Realization Flow

Subscribe → Handle Message → Device
Experimental setup
Two pilot studies were conducted with expert users.

Determine maximum time to complete a task.

Changes in template generator interface to ensure consistency.

Participant Demographics:
- Aged 18 – 26
- Computer, Genetics, Textile Engineering
- Some experience with AR, IoT and Smart Home Automation Systems
### Metrics

<table>
<thead>
<tr>
<th>Usability Dimension</th>
<th>Evaluation Metrics</th>
<th>Units</th>
<th>Investigation Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Effectiveness</strong></td>
<td>Completion Rate</td>
<td>Percentage (%)</td>
<td>Direct Observation</td>
</tr>
<tr>
<td></td>
<td>Number of Errors</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>Task Completion Time</td>
<td>Seconds</td>
<td></td>
</tr>
<tr>
<td><strong>Satisfaction</strong></td>
<td>After Scenario Questionnaire</td>
<td>1-5 Likert Scale</td>
<td>Questionnaire</td>
</tr>
<tr>
<td></td>
<td>System Usability Scale</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiment Schema

General Introduction to the System

Selection of Participants

Pre Training

IoT → AR

Training

IoT

Task 1: IoT System Configuration

Task 1.1 → Task 1.2 → ASQ

Task 2: AR System Configuration

Task 2.1 → Task 2.2 → Task 2.3 → ASQ

Task 3: Interaction with IoT using MAR UI

Task 3.1 → Task 3.2 → SUS

Legend

- Pre-Test Stage
- Training Step
- Test Task
- Subtask
- Evaluation

Task Breakdown

- Task 1: IoT System Configuration
  - Task 1.1: Data Acquisition Flow
  - Task 1.2: Task Realization Flow
- Task 2: AR System Configuration
  - Task 2.1: Default IP Port Configuration
  - Task 2.2: Insertion of Labels
  - Task 2.3: Insertion of Buttons
- Task 3: Interaction with IoT Using MAR UI
  - Task 3.1: Visualizing Sensor Output
  - Task 3.2: Controlling Devices
Pre-Training and Training

**IoT Pre-Training:**
- Data Acquisition Video
- Task Realization Video

**AR Pre-Training**
- Description of how UI elements should interact with IoT elements.

**IoT Training:**
- Import and configure a Data Acquisition flow.
- Import and configure a Task Realization flow.

**AR Training**
- No training provided for AR.
- Aim to measure how *intuitive* participants found the system.
Overview of Test 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) Insertion of labels to template
   Task 2.3) Insertion of 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
Task 1.1
Task 1.2
Task 2.1

1. Drag elements to create a UI.

- Label Box
- Button
- Image

2. Configure MQTT IP and Port:
   - Default IP Address: 192.168.2.218
   - Default Port Number: 1883

3. Export your UI design.
Task 2.2
Task 2.3
Task 3.1
Task 3.2

04/29/21
17:18:03
Light
On

ON  OFF

vuforia™
Task 3
## Results

<table>
<thead>
<tr>
<th>Metric</th>
<th>Task 1.1</th>
<th>Task 1.2</th>
<th>Task 2.1</th>
<th>Task 2.2</th>
<th>Task 2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Completion Rate</td>
<td>89%</td>
<td>89%</td>
<td>89%</td>
<td>67%</td>
<td>100%</td>
</tr>
<tr>
<td>Average Number of Errors</td>
<td>0.23</td>
<td>0.45</td>
<td>0.12</td>
<td>0.56</td>
<td>0.34</td>
</tr>
<tr>
<td>Average Completion Time</td>
<td>119.56</td>
<td>73.7</td>
<td>28.44</td>
<td>182.89</td>
<td>211.67</td>
</tr>
<tr>
<td>Given Maximum Time (sec.)</td>
<td>180</td>
<td>120</td>
<td>45</td>
<td>180</td>
<td>420</td>
</tr>
<tr>
<td>Average ASQ</td>
<td>4.82</td>
<td></td>
<td></td>
<td></td>
<td>4.74</td>
</tr>
</tbody>
</table>

SUS Score: 81.9 (A)

Kuyucu | 30/34
Participant Feedback

• Showed **interest** in the system.
• Inquired about using the system in their own **homes**.
• Felt **accomplished** after using the system.
• Found it **intuitive**.
• Suggested different **scenarios for the use of this system**.
• Made **suggestions to improve** the system.
## Performance Assessment wrt. Background

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Completion Rate</th>
<th>Completion Time</th>
<th>Number of Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CE</td>
<td>NCE</td>
<td>CE</td>
</tr>
<tr>
<td>Task 1.1</td>
<td>100%</td>
<td>75%</td>
<td>103.2</td>
</tr>
<tr>
<td>Task 1.2</td>
<td>80%</td>
<td>100%</td>
<td>76.5</td>
</tr>
<tr>
<td>Task 2.1</td>
<td>100%</td>
<td>75%</td>
<td>29</td>
</tr>
<tr>
<td>Task 2.2</td>
<td>80%</td>
<td>50%</td>
<td>150.25</td>
</tr>
<tr>
<td>Task 2.3</td>
<td>100%</td>
<td>100%</td>
<td>193.2</td>
</tr>
</tbody>
</table>
Conclusion

• An end-to-end framework integrating AR and IoT using open source technologies is presented.

• A LCNC framework which provides users with necessary abstractions so that they can create a personalized smart home application is established.

• The usability of the framework has been tested using an application generated with this framework.
Future Work

- Different methods of **data visualization**.
- Different methods of **context awareness**.
- Different **devices for AR** (head-mounted gear).
- Seamless **communication with Node-RED** to reduce cognitive load.
Thank you for watching.
MARIoT: An Authoring Framework for Creating IoT Applications with Mobile Augmented Reality

Meral Kuyucu, Gökhan Înce

User Experience Laboratory
Department of Computer Engineering
Istanbul Technical University (Turkey)

Contact email: korkmazmer@itu.edu.tr