*Corresponding Author's Email: przemyslaw.mazurek@zut.edu.pl
Proceedings of the International Conference on Modern Research in Social Sciences
Vol. 3, Issue. 1, 2026, pp. 22-31
DOI: https://doi.org/10.33422/icmrss.v3i1.1958
Copyright © 2026 Author(s)
ISSN: 3030-0630 online

A Distributed System for Studying Multimodal Interaction Using IoT-based Devices

Arkadiusz Gajda, Szymon Kostrubiec, Ewelina Łazęcka, Zuzanna Piwko, Adam Suchomski, Filip Wojno, Kacper Zawada, Jaroslaw Fastowicz, and Przemyslaw Mazurek*
Department of Signal Processing and Multimedia Engineering,
West Pomeranian University of Technology in Szczecin, Szczecin, Poland

Abstract

The study of human interactions with the environment is an important task in behavioral research. As we grow older, numerous barriers emerge that limit interactions with the environment. This limits the acquisition of new knowledge, especially when physical interaction is required. To study these phenomena, a distributed computer system with IoT devices was developed. These devices, called widgets, do not perform specific functions, nor do they have typical shapes to avoid suggesting or restricting interactions. A centralized communication solution using MQTT transmission was proposed for a diverse set of widgets (various combinations of sensors and actuators), allowing interaction sessions with interaction measurements. From the perspective of communication and application, it is an IoT system. The task of a human or group is to search for interactions, which, in a typical scheme, are based on the stimulation of unknown sensors, which is rewarded by the work of actuators (light, sound, motion). Interaction sessions can be created with random relations and implemented using Markov chains (transition matrix), allowing for the implementation of deterministic and stochastic sequences of desired interactions, which are then explored by the human.

Keywords: IoT, Distributed Systems, Embedded Devices, Human Interaction, Education

1. Introduction

One of the most important topics in human behavior is the analysis of interactions, both with the environment and with other people. This changes with age for every person and is strongly linked to many external factors: cultural, emotional, intellectual limitations, etc. (Dourish, 2001). Young children's exploration of the world is based on a wide range of methods of interacting with their surroundings, which allows for broad-based development. The primary senses are touch and taste, with sight and hearing gradually increasing. Adults, on the other hand, often limit their exploration of the world to passive visual and auditory observation. Barriers emerge, such as fear of the new, fear of ridicule, and fear of not understanding objects or phenomena. This problem is particularly evident in science centers/museums around the world (Steier et al., 2015; Duranti et al, 2016), where children actively explore their surroundings and conduct experiments, while adults are blocked.

Studying these phenomena is important not only through observing individuals, but also by comparing the interactions of similar scenarios in the same environment by other individuals to gather comparative knowledge. For this purpose, the concept of 'widgets' was developed. This word denotes a specific object, but without defining its purpose. Currently, this word usually refers to various (an umbrella term) graphical UI (User Interface) elements.

The use of tangible interaction, which connects digital information with objects, is known from the literature (Hiroshi, 2007) (Van den Hoven et al., 2024) (Burzio & Ferraro, 2026).

The idea behind using widgets in behavioral research is to present subjects with objects whose shapes are not familiar, typical shapes. The presentation of familiar shapes determines the study's load. For this reason, they should be abstract. Each widget is designed to contain sensors, unknown to the subject, as well as actuators. The relationship between sensors and actuators, i.e., the operating algorithm, is also unknown. This necessitates experimental exploration using various methods.

This approach can be scaled to a group of different widgets, allowing for a much broader study. Because widgets can communicate with each other, sensors in one widget can influence the behavior of other widgets, including actuators. An example distributed interaction scenario might be one in which detecting the lifting of one widget triggers the illumination of others. Scenarios can be diverse, allowing for the measurement of sequences of human interactions with a set of widgets. This type of research is important in the context of extended cognition (Clark, 2008).

The multimodality of the system means the ability to use various interaction methods, both through the availability of various sensors and actuators and by defining their relationships.

2. Methods

Researching multimodal interaction using widgets requires the development of structures with custom housings and electronic systems with various sensors and actuators. Real-time communication between widgets is also necessary to implement more advanced scenarios and not be limited to a single widget instance.

2.1 Autonomous vs. Centralized Solutions

The main conceptual challenge is the need for interaction between widgets. The direct solution involves implementing peer-to-peer communication, assuming full autonomy for the widgets. In this solution, each widget must execute its own algorithm, interact with others, and simultaneously respond to commands from others. This solution is then an implementation of autonomous agents. While this approach is very attractive from an implementation perspective, the relationships between widgets should be defined and controlled by a central unit **Error! Reference source not found.**. This comes from the need for scenario repetition and the possibility of scenario modification.

The second solution is a centralized one, where communication takes place between each widget and a central unit **Error! Reference source not found.**. This significantly simplifies the process of controlling scenarios and storing interaction-related data. The advantage is that the algorithms for individual widgets can be located in the central unit, eliminating the need to reprogram the microcontrollers in the widgets each time. This significantly simplifies the development process and eliminates the need for over-the-air programming. This type of solution was chosen for the implementation of a distributed system.

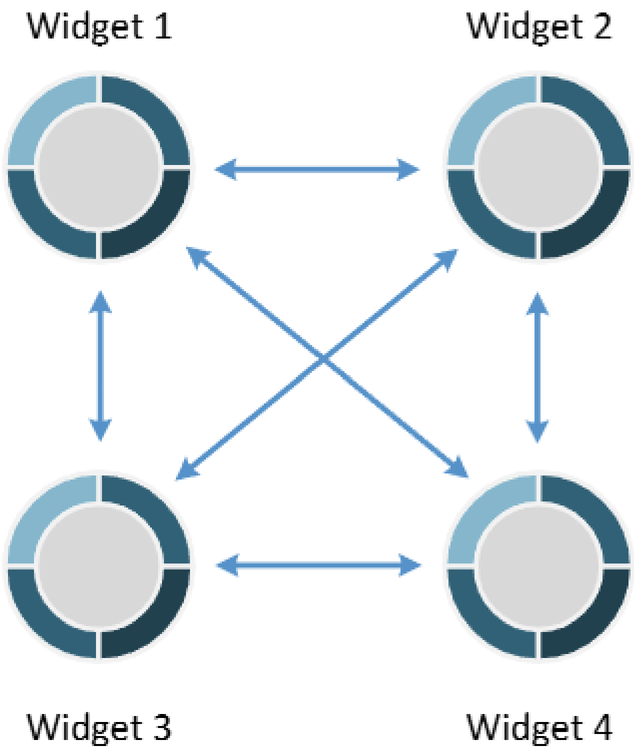

Figure 1: Autonomous solution for widgets.

Source: own elaboration

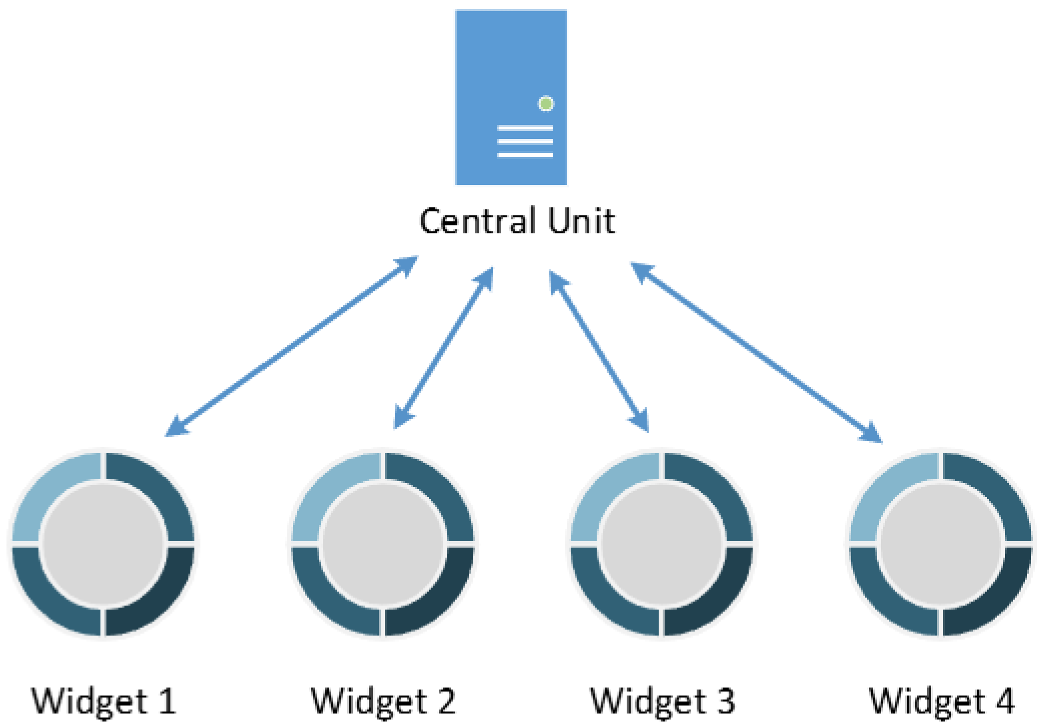

Figure 2: Centralized solution for widgets.

Source: own elaboration

From the perspective of the person participating in the interaction, there is no difference, as the central unit is physically hidden from view and is not the subject of interaction. This means that widgets implement simple measurement and actuator functionality. Low-level control is performed by a microcontroller, which receives control commands and sends formatted measurement data to the central unit. This type of implementation allows one to transfer, for example, blinking, pulsing, and other LED operations to the widget. The hardware and software architecture within the widget is also encapsulated.

2.2 Communication Interface

There are many possible solutions for wireless communication between widgets and the central unit. We decided to use a 2.4GHz WiFi network because it is well-supported by relatively inexpensive and well-documented modules from the ESP32 (Kurniawan, 2019) and RPi families. Furthermore, it is possible to use the ESP32 as a communication module for microcontrollers that do not support WiFi.

The AP (Access Point) functionality is located in the central unit, which uses the NVidia Orin board. This approach allows for the implementation of a system using proven technology. Communication takes place between widgets and a central unit. A potential problem is human obstruction of the direct line of sight between the AP antenna and the widget. However, 2.4GHz WiFi technology works well in such conditions because the distances are not large (on the order of meters).

This solution also allows communication (in the future) with a control/monitoring tablet application.

2.3 MQTT Communication Protocol

The MQTT protocol (Hillar, 2017) was used to implement main communication. MQTT is a lightweight publish/subscribe communication protocol designed for IoT (Internet of Things) systems (Smart, 2020), resource-constrained devices, and networks with low bandwidth or high latency.

MQTT is based on three main elements:

- Broker – a central server that mediates communication (central unit),
- Publisher – a client that sends messages to the broker (widget),
- Subscriber – a client that receives messages from the broker (widget).

Clients do not communicate directly with each other – all messages pass through the Broker.

Typical widget implements Publisher and Subscriber functionality, which are sensors and actuators, respectively **Error! Reference source not found.**. A special type of Publisher measures the energy of the battery that powers the widget and transmits it to the Broker.

The central unit contains the Broker and a process implementing widget interaction algorithms (Publisher and Subscriber - sending widget control and receiving sensory data, respectively).

The amount of transmitted data is not large, therefore it is possible to use QoS at level 2, which allows for the highest reliability.

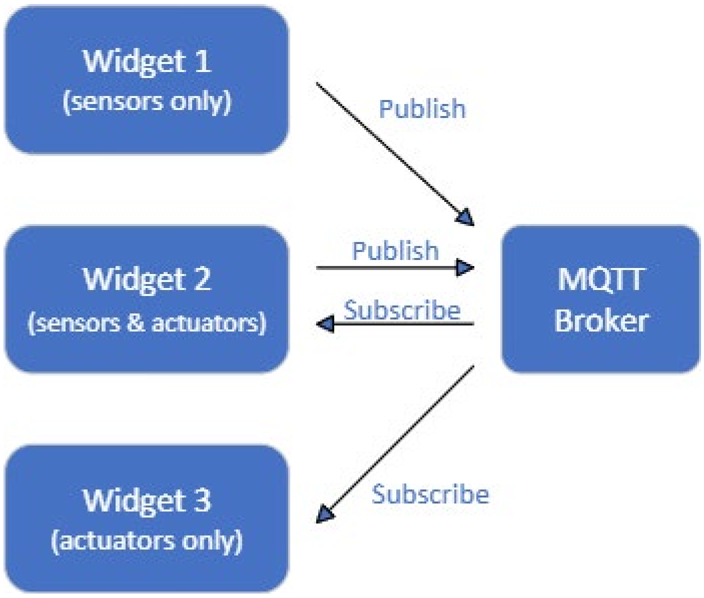

Figure 3: MQTT transmission model.
Source: own elaboration

2.4 Energy Control

One of the identified issues with the widgets' design is power supply. Because widgets can be lifted and moved during interaction, they cannot be powered by wires. A battery is required, and since each widget consumes electricity differently, it must be continuously measured to prevent the widget from shutting down during the interaction session. Li-ion batteries were used for power. The charge level is measured by measuring the voltage, which is the simplest method and is entirely sufficient in terms of quality. Using a voltage divider, the battery's output lines can be connected to the microcontroller's analog-to-digital converter.

The voltage data are sent to the Broker and read by the main process controlling the widgets.

2.5 Widget Discovery

The turning on of the system requires determining the available widgets and registering them in the central unit. The REST (Representational State Transfer) (Gaitatzis, 2019) protocol over HTTP (Hypertext Transfer Protocol) was used to retrieve initial states and widget configurations.

2.6 Basic Fault Tolerance

A significant issue for the system is fault tolerance. The simplest case of a failure is when a widget runs out of power during a session, due to the use of a motor, a large number of LEDs, etc. If the session is continuous (or if it is not expected to end), it is necessary to ignore the widgets in the state machine (algorithm) of the central unit. Another case is where a widget is removed from the work area, for example, due to a mechanical failure. In this case, the widget can be disabled using a switch.

To increase resilience to such situations, dynamic control of each widget's availability is necessary. While measuring the supply voltage allows for prediction of such situations, in other cases, regaining control in terms of interaction logic through activity measurement is necessary.

For this purpose, the Heartbeat/Keep-Alive mechanism and data reception acknowledgment are used. The time set for this is 300 seconds, after which the widget is removed from the list of available widgets.

2.7 User Interface

The server implements functionality using WebSockets (Lombardi, 2015) for control and monitoring via a user interface (for example, using a tablet). This functionality is currently being exploited (development of an application with a user interface is necessary).

2.8 Sensors

Each widget was equipped with different sensors, depending on the knowledge and decisions of its designers. This is due to the educational nature of the project. Therefore, a single set of sensors was not specified for each widget. Moreover, similar functionalities can be implemented in different widgets using different sensor models.

Typical sensors include accelerometers, microphones, and touch sensors for mechanical interaction with widgets. This type of interaction is most challenging for people with internal blockages. Touching, changing orientation, or lifting objects of unknown functionality is rewarded by the work of actuators within a given widget or another.

Different sensors are connected to different microcontroller boards via different pins, and different libraries are used. Therefore, the microcontroller's task is to encapsulate these tasks, eliminating the need for low-level control by the central unit.

This makes it necessary to define a more complex pool of observations on the widget actuators. The simplest solution would be to send raw data from the sensors to the central processing unit, but this would introduce significant bandwidth overhead, especially with certain sensors such as a microphone. Instead, general action information interpreted by the widget could be sent—high-intensity sound, or no sound at all; lifting the widget, tilting the widget, etc.

2.9 Actuators

Like sensors, actuators require encapsulation. This means that to achieve more advanced interaction methods, there is no need to, for example, turn LEDs on and off via the central unit or resolve things like blinking, which is a low-level operation. Commands like "LED on" and "LED off" are expanded to include blinking, flashing, and other more sophisticated commands, such as controlling larger groups of LEDs and color support (animations).

Important actuators are those that provide mechanical interactions. Vibration generators (a motor with an eccentric) and loudspeakers are used, and it is also possible to change the object's shape. The use of wheel-driven motors was abandoned because they are conceptually problematic (the viewer then perceives a 'robot' or 'car' due to the wheels) and, if moved too far, could cause an uncontrolled fall from the table.

2.10 Control Algorithm

The most important part of the system is the control algorithm. Since centralized implementation was chosen, the control mechanism is simplified because it can be synchronous. In the autonomous variant, it is a set of asynchronous algorithms, or synchronous if synchronization is used. The ability to centralize the algorithm simplifies implementation, particularly testing, and does not impact the user, similarly to centralization.

There are two key issues regarding the control algorithm. The first is whether it should be deterministic or stochastic. The second is the level of complexity of the interactions implemented by the algorithm. The simplest variant is an algorithm implemented as a deterministic state machine, where each state is the state of the actuators, and the edges determining state changes are the activation or activations of sensors. In this case, the time aspect is omitted. Implementing this type of task is algorithmically simple because it involves a unidirectional directed graph with a fixed initial state (the beginning of the interaction session) and a fixed final state (the end of the interaction, achieving success/winning, as in a simple game). It is worth noting that from a human perspective, the interaction resembles the mechanism most commonly used in 'escape room' games (Lyman, 2021).

Since the system's primary goal is to enable the study of human interactions with widgets, the algorithm must be freely definable. Therefore, using a Markov matrix (Privault, 2018), (Van Der Post & Bisette, 2025) is a good option, as it allows the implementation of deterministic, stochastic, and mixed algorithms (where some transitions are deterministic and others are stochastic). Enabling stochastic transitions provides alternative paths within the session and the possibility of creating loops.

The use of Markov transition matrices to describe the work enables mathematical analysis of the session based on Markov chains **Error! Reference source not found.**. This allows for comparative studies. Successful or unsuccessful interactions (transitions) can also be described using Markov chains. In addition, the recording can include time measurements, providing important research data.

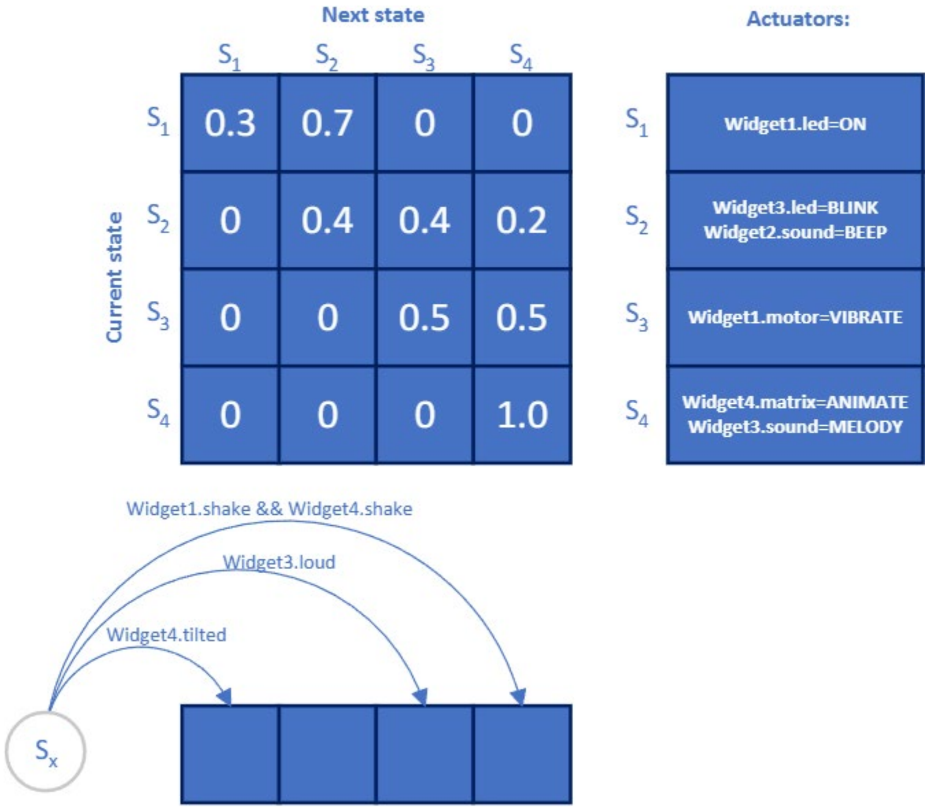

Figure 4: Example of session with stochastic matrix.
Source: own elaboration

3. Results

As part of the project, several widgets were independently created using the same ESP32-S3 microcontroller module, allowing for the same wireless communication. Individual widgets have dedicated code for sensors and actuators, simplifying design and control. The sample configurations are presented in the Table 1, and the widgets are shown in **Error! Reference source not found.**.

Figure 5: Example widgets.
Source: own elaboration

3D printing technology was used for the enclosures, using various textures and colors. The designers of the widget chose abstract shapes so that the enclosures did not suggest their intended use. Furthermore, actuators, particularly those that use LEDs, are usually hidden within the enclosure so that the individual LEDs are not visible.

Table 1: Exemplary widgets hardware.

No.	Sensors	Actuators	CPU
	electret microphone with MAX9814 APDS-9930 (digital proximity and ambient light sensor) PAJ7620 (gesture sensor)	vibration motor MT73 LED WS2812B ring	ESP32-S3
1.	electret microphone with MAX9814	vibration motor MT73	ESP32-S3
2.	2× APDS-9930 (digital proximity and ambient light sensor)	2× speaker 3W vibration motor MT73 RGB stripe WS2812B 60 LEDs	ESP32-S3
3.	AHT20 (temperature and humidity)	WS2818B 16×16 LED matrix	ESP32-S3
4.	modGY-87 -10DoF module DFRobot Gravity Capacitive Touch Sensor	vibration motor MT73 RGB stripe WS2812B 60 LEDs 2× speaker 3W	ESP32-CAM

No.	Sensors	Actuators	CPU
5.	modGY-87 (10 DoF module) DHT22 (temperature and humidity sensor) 3× APDS-9930 (digital proximity and ambient light sensor)	speaker 3W RGB stripe WS2812B 60 LEDs Serwo SG-90	Raspberry Pi Zero 2 W
6.	ZeroCam OV5647 5MP Fisheye Camera Module GL5616D Light Dependent Resistor modGY-87 (10 DoF module)	RGB stripe WS2812B 60 LEDs 2× speaker 3W	Raspberry Pi Zero 2 W

Source: own elaboration

4. Discussion

The problem of interacting with the unknown, seeking new experiences, and exploring these processes is made possible by the designed system. The widgets were presented at several open events, particularly for young people.

Only a small number of people decided to interact with the widgets, and only after being explained what they were and what they were used for. None of them engaged in any spontaneous interaction. This means that these groups are completely blocked from exploring the world through interaction with the unknown. These needs are met via smartphones and reels.

One of the observed behaviors was that a group of young people moved away from the widgets so as not to be the center of attention or so that someone from that group was not selected for interaction.

A noticeable problem was that people interacting with wdigets expected to be instructed on what to do. There was a lack of spontaneity in exploration.

5. Further Works

The system will be developed into other designs and use cases. A study is planned in groups of younger children to determine whether or not blockages occur. The development of the system is also planned in the field of control and monitoring application using a tablet. The cases are made using 3D printing technology, but they can be covered with various materials to obtain a greater variety of widgets.

Acknowledgment

Projekt finansowany ze środków budżetu państwa, przyznanych przez Ministra Nauki i Szkolnictwa Wyższego w ramach Programu „Studenckie koła naukowe tworzą innowacje” SKN/SP/631122/2025.

The project is financed from the state budget, allocated by the Minister of Science and Higher Education under the Programme "Student science clubs create innovations" SKN/SP/631122/2025.

References

Burzio, G. & Ferraro, V. (2026). *Physicalizing Data: Design Approaches for Human-Data Interaction*. https://doi.org/10.1007/978-3-032-05322-0

Clark, A. (2008). *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195333213.001.0001

Dourish, P. (2001). *Where the action is: The foundations of embodied interaction*. MIT Press. https://doi.org/10.7551/mitpress/7221.001.0001

Duranti, D., Spallazzo, D., & Trocchianesi, R. (2016). *Tangible interaction in museums and temporary exhibitions: embedding and embodying cultural heritage*. 6th International Forum of Design as a Process Systems & Design: Beyond Processes and Thinking, https://doi.org/10.4995/IFDP.2016.3322

Gaitatzis, T. (2019). *Learn REST APIs.* United States : BackupBrain Press.

Hillar, G. C. (2017). *MQTT Essentials - A Lightweight IoT Protocol.* Packt Publishing.

Hiroshi, I. (2007). "Tangible User Interfaces". The Human-Computer Interaction Handbook. pp. 495-514, https://doi.org/10.1201/9781410615862-35

Kurniawan, A. (2019). *Internet of Things Projects with ESP32.* Packt Publishing.

Lombardi, A. (2015). *WebSocket: Lightweight Client-Server Communications.* O'Reilly Media.

Lyman, P. E. (2021). *The Do-It-Yourself Escape Room Book.* Skyhorse Publishing.

Privault, N. (2018). *Understanding Markov Chains.* Springer Singapore. https://doi.org/10.1007/978-981-13-0659-4

Smart, G. (2020). *Practical Python Programming for IoT.* Packt Publishing.

Steier, R., Pierroux, P., & Krange, I. (2015). *Embodied interpretation: Gesture, social interaction, and meaning making in a national art museum*. Learning, Culture and Social Interaction, 7, 28–42. https://doi.org/10.1016/j.lcsi.2015.05.002

Van den Hoven, E. D., Mazalek, A., & Margetis, G. (2024). *Tangible and embodied interaction*. W C. Stephanidis & G. Salvendy (Eds.), Interaction techniques and technologies in human-computer interaction (pp. 290–324). CRC Press. https://doi.org/10.1201/9781003490678-11

Van Der Post, H. & Bisette, V. (2025). *Random Processes and Markov Chains.* Independently published.