Smart Museum

Software Engineering for Autonomous Systems

Beatrice Tomassi (matr. 295351)

Lorenzo Rocchi (matr. 294418)

Table of Contents

Introduction 3
System Requirements 4
1. Functional Requirements 4
2. Non Functional Requirements 4
System Components 6
1. Sensors and Actuators 6
2. Adaptation Goals 7
MAPE-K Loop 8
Conclusion 13

Introduction

Preserving priceless works of art and ensuring an optimal visitor experience are two of the fundamental challenges that museums face nowadays. The proposed Smart Museum system aims to approach these problems by operating autonomously, responding to environmental changes and human activity in real-time.

Given the context, let's go deeper into each of the primary goals:

1. **Preservation of artworks through environmental monitoring:**

Artworks, especially ancient and fragile pieces, are extremely sensitive to environmental fluctuations. Temperature, humidity, and light intensity must remain within specific thresholds to prevent irreversible damage such as fading, warping, or mold development.

The proposed system continuously monitors these parameters and dynamically adjusts conditions.

2. **Visitor management:**

Museums are public spaces that aim to provide an enriching experience to visitors, yet overcrowding can threaten both the experience and the safety of the exhibits.

To mitigate this risk, the system incorporates sensors to assure that no room is overcrowded.

System Requirements

To achieve the predefined scenario, the requirements explicated in the following sub-sections have been implemented.

1. Functional Requirements

RF1: Environmental Monitoring (Monitor Phase)

The system should collect real-time data on temperature, humidity, light intensity, and air quality in each room of the museum.

RF2: Presence Monitoring (Monitor Phase)

The system should collect real-time data on the number of visitors present in each room of the museum.

RF3: Data Analysis (Analyse Phase)

The system should analyse collected data by comparing it against predefined thresholds to detect anomalies or deviations.

RF4: Environmental Management (Plan & Execute Phase)

The system should automatically regulate temperature, humidity, lighting, and air quality to maintain optimal conditions for artwork preservation.

RF5: Visitor Flow Management (Plan & Execute Phase)

The system should trigger an alert to notify museum staff when the number of visitors in a room exceeds the allowed capacity.

RF6: Knowledge Management (Knowledge Phase)

The system should store collected environmental and visitor data in a database and use it to make analysis.

2. Non Functional Requirements

NFR1: Usability

The system should provide an intuitive interface for museum staff to view real-time and historical data.

NFR2: Scalability

The system should support the addition or removal of rooms without requiring significant architectural changes or manual reconfiguration.

NFR3: Data Reliability

The system shall ensure that stored data remains available and accessible for future analysis.

System Components

1. Sensors and Actuators

- **Temperature Sensors - HVAC System (Air Conditioning/Heating) (1 per room)**

  The temperature sensors measure the ambient temperature in each room of the museum. Numerical values are generated within the range of 8°C to 35°C, with a higher probability of values falling within the range of 18°C to 25°C (non-critical range).

- **Humidity Sensors - HVAC System (Air Conditioning/Heating) (1 per room)**

  The humidity sensors measure the air humidity level. Numerical values are expressed as a percentage, with a higher probability of values ranging between 30% and 60%.

- **Light Sensors - Adaptive Lighting System (1 per room)**

  The light sensors measure the light intensity within a room. Each one detects values ranging from 0 to 300 lux, with higher probability of generating data in the range 50 to 200 lux.

- **Air Quality Sensors - Ventilation System (1 per room)**

  The air quality sensors measure the CO_2 concentration in the air of a room, generating numerical values in ppm (parts per million) in the range 400 ppm to 2000 ppm, with a higher probability of values between 500 ppm (fresh air) and 1000 ppm (stale air).

- **Presence Sensors - Alarm System and Door System (1 per room)**

  The presence sensors detect the number of people in each room. In the simulation, *x* people will be added or removed from the total count of each room, with *x* being a randomly generated number between 0 and 5.

Simulated sensor readings are generated every 5 seconds.

2. Adaptation Goals

Goal	Sensor-Actuator involved	Evaluation Metric
Mantain optimal room temperature.	Temperature Sensors - HVAC System	If the temperature is too high (> 25°C) or to low (<18°C), activate the HVAC system.
Maintain stable humidity levels to prevent artwork deterioration.	Humidity Sensors - HVAC System	If humidity is too high (> 60%) or too low (< 30%), activate the HVAC system.
Maintain adequate lighting for visibility and artwork preservation.	Light Sensors - Adaptive Lighting System	If the light intensity is too bright (> 200 lux) or too low (< 50 lux), adjust the Lighting system.
Ensure good air quality.	Air Quality Sensors - Ventilation System	If the CO_2 concentration exceeds 1000 ppm or falls below 500 ppm, regulate the Ventilation system.
Ensure museum rooms are not overcrowded.	Presence Sensors - Alarm System and Door System	If the number of people in a room exceeds a certain limit (that depends on the room size), the alarm system notifies staff and the door of the room is closed.

MAPE-K Loop

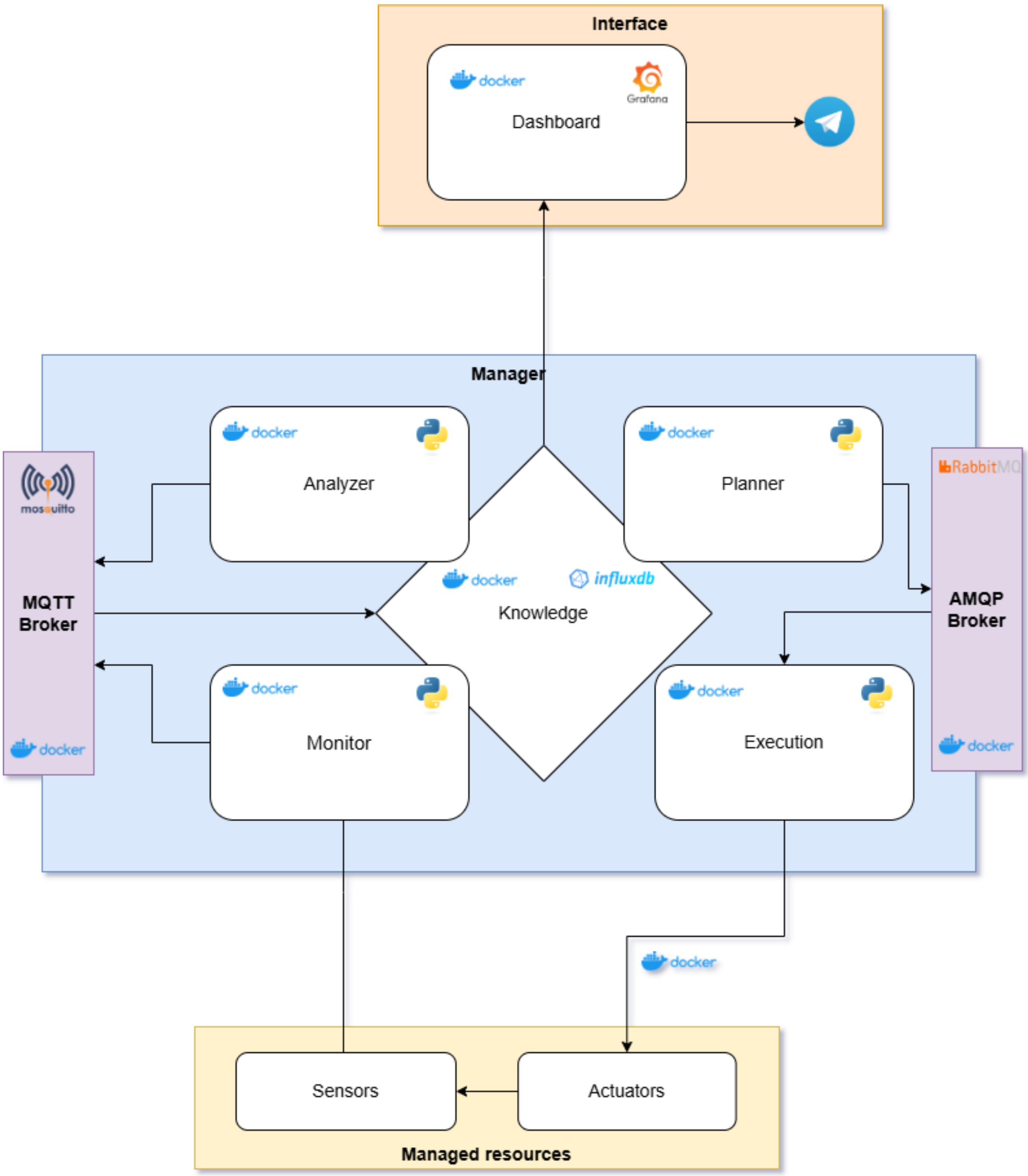

Figure 1 - MAPE-K Loop of the Smart Museum

The figure gives an overview of how the MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) loop is implemented in the museum monitoring system. The system continuously collects sensor data, processes it to detect anomalies, plans corrective actions, and executes interventions using actuators or notifications.

MQTT Broker

The component, thanks to Telegraf (that is properly configured to forward all the data received to InfluxDB's *SmartMuseum* bucket) effectively stores the data in the Knowledge component.

Monitor

The component gathers real-time data from the sensors placed in the museum and saves them in the Knowledge component.

In particular, in this simulation, is the Monitor itself that generates new sensor data every 5 seconds. This data is then published to the MQTT Broker for each room using specific topics that follows this notation: *room/{room_name}/{metric}*.

From the Broker the data is stored in the DB.

<u>Note</u>: In order to simulate the environment, the sensor data are generated depending on the state of the corresponding actuator.

Analyzer

The component takes the data stored by the Monitor in the Knowledge and analyses them to identify anomalies (e.g., excessive humidity or overcrowding in a room). Specifically, it retrieves the most recent data (the last four measurements) for each metric and computes the average. Once the average is calculated, the system compares it against thresholds defined in the adaptation goals.

For the **temperature, humidity, light and air quality** metrics, the system checks if the average value exceeds or falls below the predefined min and max thresholds:

- If the value is below the minimum threshold, the status is set to -1 (**critical condition: below min threshold**);
- If the value is within the limits, the status is set to 0 (**regular condition**);
- If the value is above the maximum threshold, the status is set to 1 (**critical condition: above max threshold**).

Concerning the **presence** metric, we established that the number of visitors cannot exceed half of the room's size (2 square meters per person). So, the status is determined based on the room's size and the number of people present:

- If the average value is exceeds the threshold, the status is set to 1 (**critical condition**);

- In the other cases, the status is set to 0 (**regular condition**).

The results of this analysis, including both the calculated average value and the room's status, are then published back to the MQTT broker with a specific topic for each room and metric: *analysed/room/{room_name}/{metric}*.

From the Broker the data is stored in the DB.

Planner

The component decides how to respond to possible anomalies by planning interventions to restore optimal conditions.

Specifically, it retrieves from the Knowledge the status for each metric and from a Docker volume the state for each actuator, aggregating them into a data structure.

Then, it generates plans based on the sensor and actuator states.

The logic for generating plans for the **HVAC**, **lighting** and **ventilation systems** is as follows:

- If the sensor state is 0, no action is taken.
- If the sensor state is 1 and the actuator state is 0, the plan is set to -1.
- If the sensor state is 1 and the actuator state is -1, the plan is set to -2.
- If the sensor state is 1 and the actuator state is greater than or equal to 1, the plan is set to 0.
- If the sensor state is -1 and the actuator state is 0, the plan is set to 1.
- If the sensor state is -1 and the actuator state is 1, the plan is set to 2.
- If the sensor state is -1 and the actuator state is less than or equal to -1, the plan is set to 0.

<u>E.g. for Temperature and HVAC System</u>:

> If the temperature is regular, no action is taken.
>
> If the temperature is below the minimum threshold and the HVAC is not active, then the HVAC is planned to be activated and set to mode 1, meaning that it is not so urgent to increase temperature; however, if the temperature continues to be below the minimum threshold while the HVAC is active, then it is set to mode 2, meaning that the temperature should increase urgently.

Same goes for the other cases.

Instead, the logic for generating plans for the door system is:

- If the sensor value is 1, the plan is set to 1.
- If the sensor value is 0, the plan is set to 0.

This means that if a room is overcrowded the door is closed.

The complete plan is formatted into a dictionary (*plans[room][metric] = state*), converted to a JSON format and sent to a RabbitMQ queue.

Executor

The component, based on what the Planner decides, activates the actuators.

It does so by listening to the RabbitMQ queue. When a plan is received, it updates the Docker volume that stores the status of each actuator in every room. This volume is then read by the Monitor, which, according to the stored values, generates new metric readings that are compliant to the plan.

Knowledge

The component is responsible for storing both the raw data collected by the Monitor and the analyzed data produced by the Analyzer. In particular, the system stores everything in the *SmartMuseum* bucket.

The stored data is then used by Grafana to generate visualizations and dashboards.

Grafana

The component is responsible for visualizing real-time data collected from all sensors in the museum. It provides a dashboard that displays:

- Time series graphs for environmental metrics (temperature, humidity, light and air quality).
- Gauge visualizations for presence detection.

In addition, Grafana is also used to simulate the alarm system for presence management. If a room becomes overcrowded, Grafana triggers an alert by sending a Telegram message to notify the staff, ensuring their intervention.

Figura 2 - Dashboard

Conclusion

The proposed Smart Museum system uses a the MAPE-K loop to ensure optimal environmental conditions and visitor safety. Through a combination of sensor data collection, automated analysis, and adaptive planning, the system effectively regulates temperature, humidity, light, and air quality while also monitoring room occupancy.

All the steps needed to set the system up are contained in the README.md file of the system's repository.