UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Software Engineering for The Autonomous Systems

SMART AND ADAPTING ANOMALY DETECTOR FOR GREENHOUSE

EDISS MASTER'S PROGRAMME

Erasmus+

Submitted by:

Debayan Bhattacharya

Kabita Adhikari

Francisco Javier Macias Villaecija

Submitted to:

Professor Davide Di Ruscio

Date: 06/05/2024

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Table of content

1. Introduction

1.1. Project Description

The Smart And Adapting Anomaly Detector For Greenhouse project represents a transformative approach to modern agriculture, harnessing advanced technologies to optimize crop cultivation and resource utilization. By integrating IoT sensors, MQTT communication protocol, InfluxDB database, and machine learning algorithms, the system enables real-time monitoring, anomaly detection, and feedback mechanisms crucial for maintaining ideal growing conditions. This report provides an overview of the smart greenhouse system's architecture and functionalities, highlighting its role in enhancing crop yield, minimizing resource consumption, and facilitating sustainable agricultural practices.

1.2. Goal of Project

The implementation of the smart greenhouse system addresses the challenges faced by traditional farming methods by offering precise control over environmental parameters such as temperature, humidity, light, and soil moisture. This project aims to adapt the situation and threshold for the values in the sensors in Greenhouse according to the changing environmental conditions. Through continuous data collection and analysis, growers can make informed decisions to adjust conditions as needed, ensuring optimal plant growth and health. Additionally, the incorporation of anomaly detection algorithms enhances the system's capability to detect and respond to deviations from normal patterns, mitigating risks and maximizing productivity. Overall, the smart greenhouse system represents a significant step towards revolutionizing agricultural practices, promoting efficiency, sustainability, and resilience in food production.

1.3. Functional Requirements

The functional requirements for this systems are :

1.3.1. **Real-time Monitoring:** The system must collect and process sensor data from various environmental parameters such as temperature, humidity, light intensity, and soil moisture in real-time.

1.3.2. **Anomaly Detection:** It should include anomaly detection algorithms to identify deviations from normal conditions and trigger alerts or corrective actions when necessary.

1.3.3. **Feedback Mechanism:** The system must provide a mechanism for receiving feedback from execution component to analyzer component to make a proper feedback loop.

1.3.4. **Action Execution:** It should be capable of executing planned actions based on analysis results, such as adjusting ventilation, turning on fan

1.4. Non-functional Requirements

The non functional requirements are:

1.4.1. **Scalability**: The system should be designed to handle a scalable number of sensors and devices, accommodating the growth of the greenhouse operation without significant performance degradation.

1.4.2. **Reliability**: The system should have high uptime and fault tolerance, ensuring that critical operations such as data collection, analysis, and control are not compromised even in the event of failures or network disruptions.

2. System Components

2.1. Sensors and Effectors

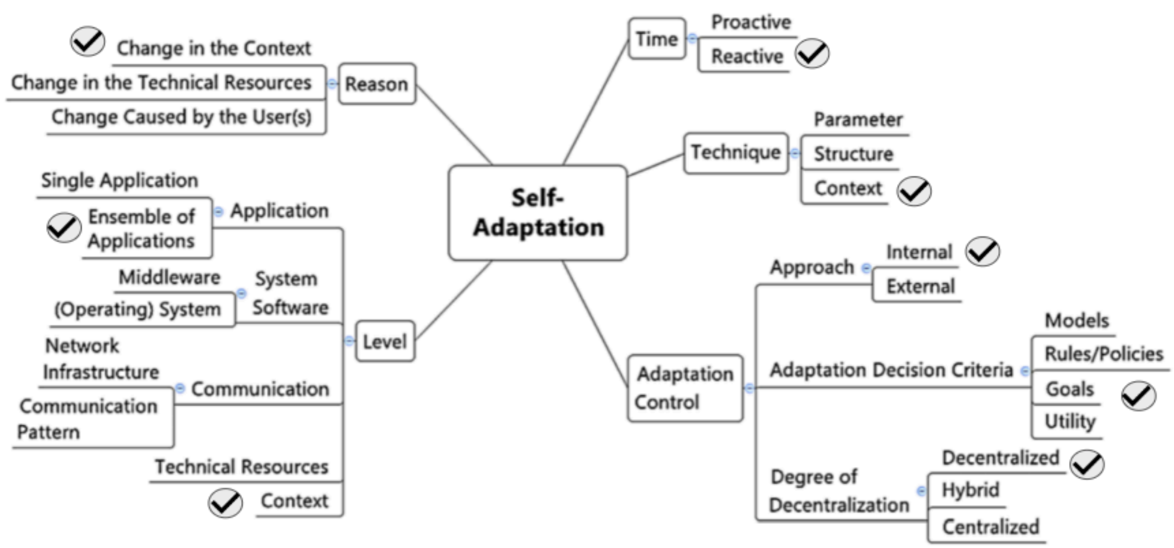
There are sensor simulators for the generating data such as temperature which are sending temperature data to monitor continuously.

There is a simulator for generating weather forecast data to plan continuously.

The effectors facilitates the execution of actions given by planner in greenhouse

2.2. Autonomic Manager

2.2.1. Architectural pattern



Self-Adaptation

- **Reason**
  - Change in the Context: Adaptation occurs when the forecasted weather condition and anomaly detected by the model.
- **Level**

- Application
  - Ensemble of Applications: The system is based on containerized services which are communicating through a message broker.
- Context: The system adapts to the change in weather forecasted from the simulator and anomaly detected in the data(temperature)

- **Time**
  - Reactive: The analyzer module adapts to the change in the action as there is anomaly in the temperature data or in the weather forecast
- **Technique**
  - Context: The system keeps the temperature data coming from the sensors and continually updates as the new temperature is recorded in the sensor.
- **Adaptation control**
  - Approach:
    - Internal: The adaptation logic is linked with the managed resources with adaptation initiated in response to changes within these resources.
  - Adaptation:
    - Goals: Goals drive the adaptation
  - Degree of Decentralization
    - Centralized: Adaptation logic is implemented in the planner component which takes the weather forecast and model.

2.2.2. Adaptation Goal

Adaptation goals of the Smart GreenHouse

Goal	Description	Evaluation metric
Detect Anamoly in Temperature	Detect anomalies in current temperature conditions	$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$ *
Adjust GreenHouse Conditions	Implement adaptive measures to adjust the condition	1. If there is no anomaly (1) no action is required. 2. If an anomaly is detected (anomaly score = -1) and there is a forecast for rain ("yes") then windows should be closed, and fans turned off. 3. If an anomaly is detected (anomaly score = -1) and there

		is no forecast for rain ("no") then adjust the ventilation, and fans turned on.
Sense Operations and adapt the conditions	Check if measures could be adapted or else send feedback to adjust with new data	If executor action failed then feedback = failure. Now for this current normal temperature adjust my anomaly detection model, so that new temperature (due to adjustment failure) is not treated as anomaly.

* Anomaly Score is calculated

Where:

- x is the data point.
- n is the number of samples.
- $E(h(x))$ is the average path length of x over a collection of trees.
- $c(n)$ is the average path length of unsuccessful search in a Binary Search Tree (BST) and is used for normalization.

If s(x,n) is close to 1, the data point x is more likely to be an anomaly. If it's much smaller than 0.5, x is likely to be normal. Values close to 0.5 generally indicate insufficient information to determine the nature of the data point.

2.2.3. Decision Function

The decision function of this autonomic manager is based on a hybrid approach which consists of rule based and AI based methods. The rules define specific temperature for the greenhouse to be in stable condition and an AI based method consists of Isolation Forest which detects the anomaly of the temperature and gives a decision to planner according to the data. This hybrid approach allows us to combine and explore the advantages of the rule-based systems and AI based systems.

2.3. MAPE-K

2.3.1. Monitor

Collects data from various sensors installed in the greenhouse, such as temperature, humidity, and light intensity, to continuously monitor environmental conditions.

2.3.2. Knowledge

Maintains a knowledge base that stores historical data.

2.3.3. Analyzer

Utilizes machine learning algorithm Isolation Forest to analyze the collected data, identifying patterns, trends, and anomalies that may indicate issues and gives decision to planner to take further action.

2.3.4. Planner

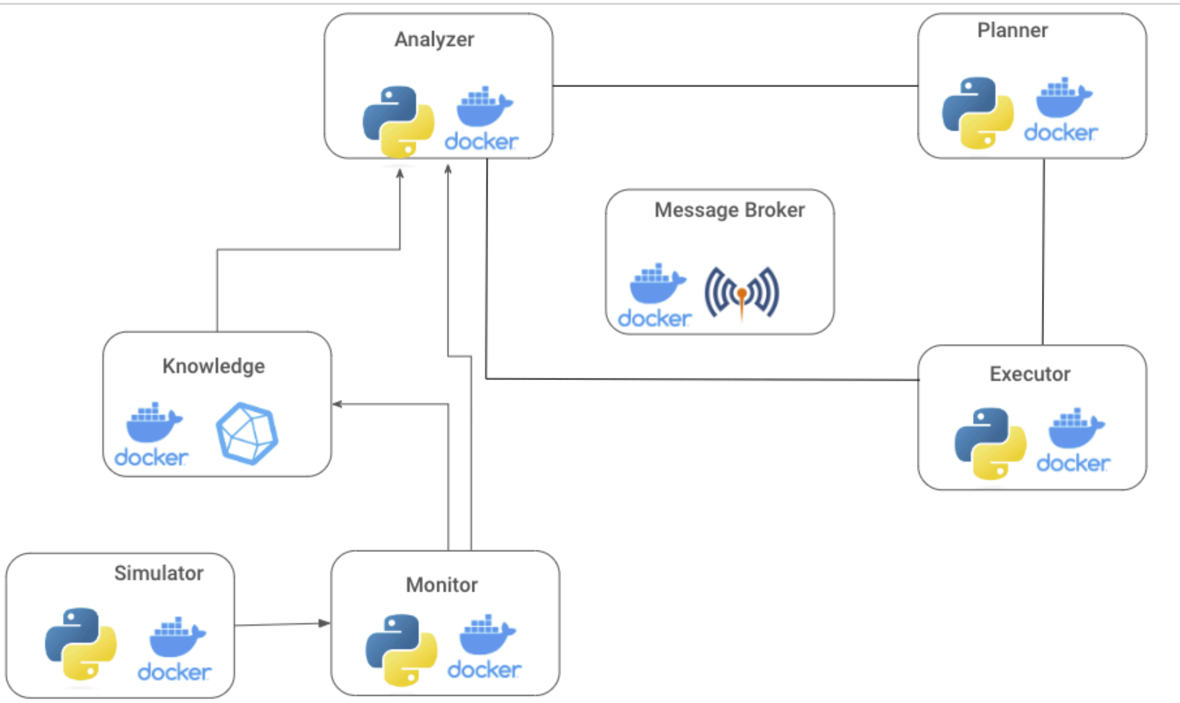
Based on the analysis results, generates actionable plans or strategies to optimize greenhouse operations, such as adjusting ventilation, or lighting settings.

2.3.5. Executor

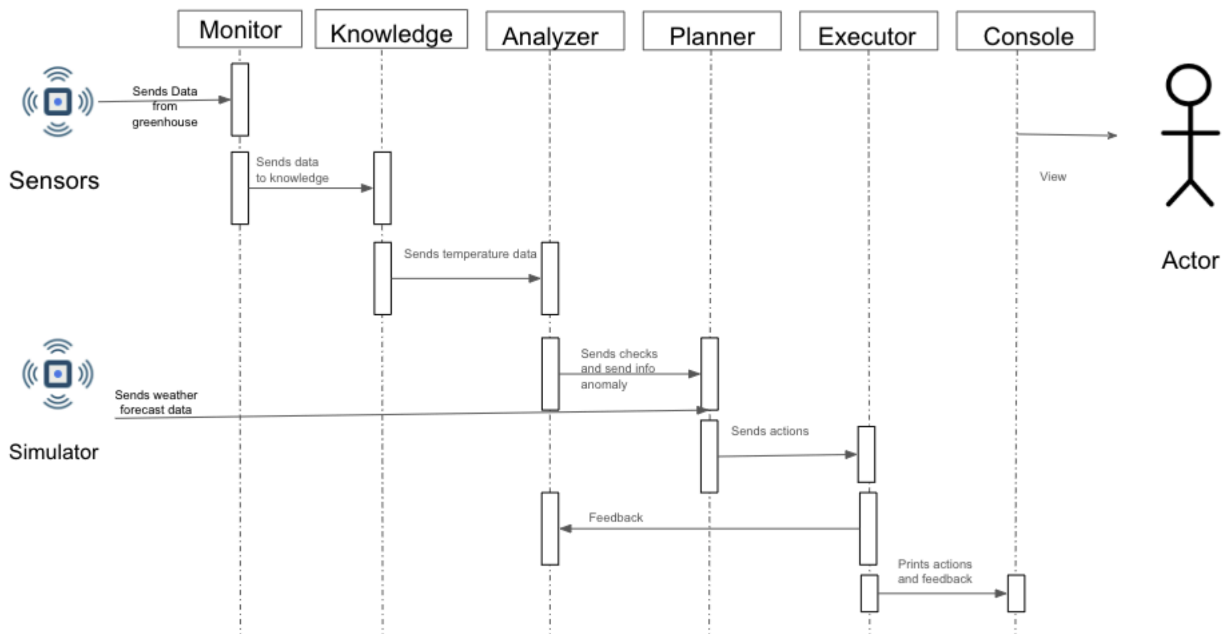
Implements the planned actions in the greenhouse environment, controlling actuators and devices to carry out tasks like adjusting ventilation systems and sends feedback to the analyzer if it was able to perform the action or not.

3. System Architecture

3.1. MAPE-K Architecture diagram



3.2. Sequence Diagram



3.3. Technologies

Docker

docker Docker facilitates the deployment and management of autonomous systems by providing a standardized way to encapsulate and deploy applications and services. Docker containers ensure that dependencies are isolated, promoting autonomy and consistency in the execution of complex systems

Python

Python is a popular choice for the development of autonomous systems due to its versatility and ease of use. It is used for implementing control algorithms, interacting with hardware, processing data, and developing interfaces all of which are crucial in autonomous systems.

MQTT Broker

In autonomous systems, the MQTT broker serves as the central component enabling efficient communication between different modules or autonomous devices. It facilitates the transmission of data and commands, which is essential for real-time coordination and decision-making.

InfluxDB

It is a tool for the storage, analysis and visualization of time series data, providing scalability, performance and flexibility to manage large volumes of data generated by these systems in real time.

4. Instructions

Git repository link: *https://github.com/srdebayan/greenhouse_anamoly_model.git*

Steps to run the project:

1. Clone the project using the link provided above
2. In terminal: docker-compose up