Design and Implementation of a Digital Twin Platform in Vertical Farming Systems

Tae-Hwan Ko, Hea-Min Lee, Dong-Hee Noh, JuHwan, Choi*, Sung-Woo Byun†
*IT Application Research Center, Jenbuk Regional Branch, Korea Electronics Technology Institute (KETI), Jeonju, Republic of Korea
{kth1147, lee10849, dhee.noh, netside}@keti.re.kr
†SMEs Innovation Support Center, Jenbuk Regional Branch, Korea Electronics Technology Institute (KETI), Jeonju, Republic of Korea
swbyun@keti.re.kr

***Abstract*—This paper proposes the framework design of a digital twin system on the real scale of a vertical farm. Specifically, in this paper, we design the system as the three-part of modules, 1) node of a real vertical farm, 2) gateway of a real vertical farm, 3) virtual world of the vertical farm as a digital twin server. We also choose to use the Eclipse Ditto framework, which is commonly used as an open-source in the digital twin. Finally, we implement the prototype of a digital twin system and tested it from 1 Jan to 31 Dec 2021.**

***Index Terms*—Digital Twin, Vertical Farm, Simulation, Eclipse Ditto, IoT**

I. Introduction

Currently, in the agricultural field, IoT devices are being used in many areas, such as sensors that can measure temperature, humidity, CO2, etc., and nutrient solutions that can be controlled remotely in order to advance to a smart farm [1]–[4]. If this is implemented as a digital twin, data collected through large-scale sensors is expressed in a virtual environment and managed in a virtual environment at the same level as the real one.

This paper describes the design and implementation of the digital twin framework applicable to smart farms. especially, the digital twin-related technology trends and the key technologies and related solutions mainly used in the design of this digital twin framework, etc. The Implemented system has been tested in a real greenhouse, which has located in Gyeonggi-do, and data for about a year from January 1 to December 31, 2021. The data was collected through Raspberry Pi. The real-world data collected in this way was visualized as digital twin information through a web application built through a client implemented separately within the framework.

The remainder of this paper is organized as follows: Section 2 presents the Implemented design system of digital twin in vertical farming system. Sections 3 explains an the results of Implementation and Section 4 concludes this work.

II. System Design

A. Smart Farm Twin Data Configuration

In this paper, a prototype digital twin system was developed that collects environmental data through a Raspberry Pi installed separately at a plant factory in Gyeonggi-do and visualizes it in a virtual environment. At this time, the main

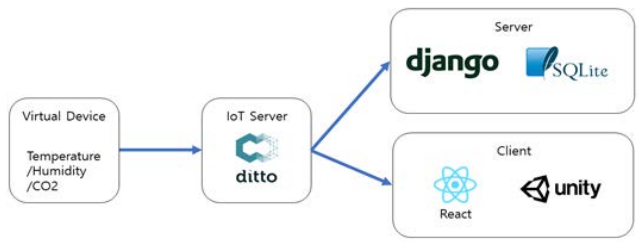

Fig. 1: The Proposed Overall Structure.

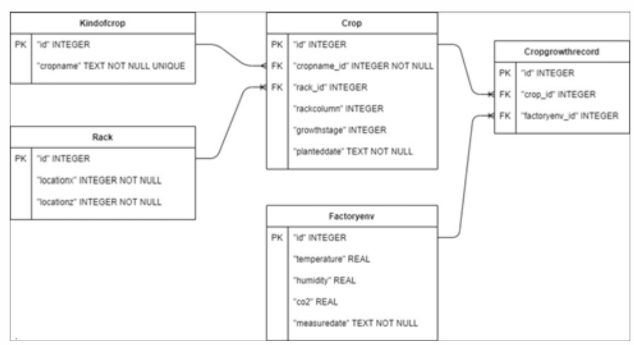

Fig. 2: The ERD Structure of Proposed System Design.

sensor data collected through the Raspberry Pi to implement this system are indoor temperature, indoor humidity, and indoor CO2.

As for the Database structure for the collected data, as shown in Table 1, ERD (Entity-Relationship Diagram) was designed with the following structure (Figure 2).

B. Smart Farm Twin Client

In this paper, the framework for expressing environmental information and facility information was collected in an arbitrary demonstration environment (household gardening greenhouse, vertical smart farm, etc.) A virtual environment identical to the real one was subdivided into a client, server, and gateway modules and implemented.

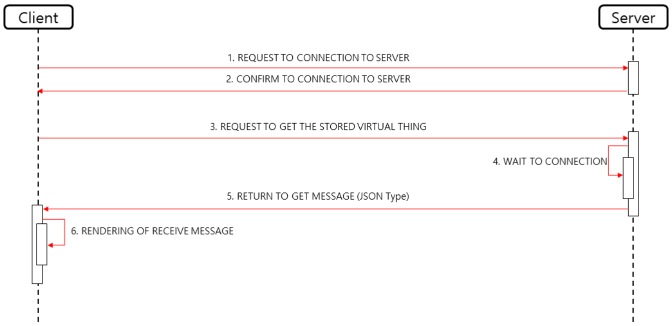

Fig. 3: The Sequence Diagram : Communication.

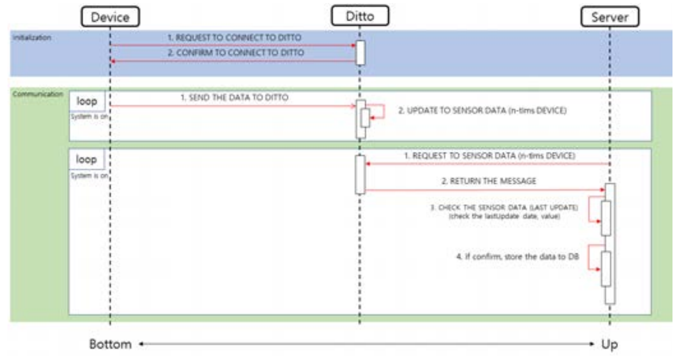

Fig. 4: The Sequence Diagram : Sending and Receiving the Data.

Among the frameworks of this system, **Smart Farm Twin Client** is responsible for objectifying data collected through sensors. We define environmental information, growth information, and facility-related information, which is real-world data information transmitted to the server, and develop an interface to communicate with the server. Through this, data is objectified precisely as it is in reality and then implemented as a digital twin. Through this, the primary users who use this system, farmers and major managers who operate plant factory-type farms, can make real-life plant factory smart farms even if they do not go directly to the site to provide convenience and accessibility to management. For this purpose, in this paper, by adding a web function module to the client, it is designed so that users can use this service by accessing it through the web.

The protocol design for real-time synchronization of data between physical and virtual objects was developed to collect physical information in the real world using WebSocket. There is currently no established case of data communication standard in the plant factory environment, but the data communication standard for the standard greenhouse was established through PG426, so the interface was designed with reference to it [5]. The main communication procedures are the initialization procedure and the client, gateway, and server after initialization. It consists of a sequential data transfer procedure between each other.

First, as an initialization procedure, the process of initializing the web application by first executing it was designed as an interface, and the situation at this time is expressed as a sequence diagram as follows (Figure 3). As the web application runs, virtualized objects stored in the Smart Farm twin server are requested, and when the return is made to the client through the server, based on the object, the 3D image rendered in advance is rendered in real-time through WebGL after being placed in the virtual environment.

Second, as the transmission/reception procedure for environmental data, the procedure for collecting environmental data in an actual greenhouse through IoT and transmitting the collected data through a server is shown below (Figure 4).

It receives environmental data values such as temperature, humidity, and CO2 through a JSON-based GET request at a predefined time through the smart farm twin server. At this time, the server is designed to operate according to the design rules of the REST API, stores the data received from the server, always maintains communication with the client through WebSocket, and sends the environmental data so that the environmental data can be inquired in real-time. In particular, in the case of a gateway, it is a platform that is used universally as a digital twin hardware open framework and is responsible for updating the data received from the sensor with the latest information.

C. Smart Farm Twin Gateway

Among the frameworks of this system, the **Smart Farm Twin Gateway** is responsible for managing multiple IoT devices connected to the plant factory digital twin system. Currently, the "Eclipse Ditto" platform, an open hardware platform commonly used in digital twins, which was used. developed. Eclipse Ditto is a framework that collects, registers, and manages IoT devices in the real world. To support the connection between multiple IoT devices, Eclipse Ditto mainly uses 5 protocols (HTTP, MQTT, CoAP, AMQP, custom) It can be provided as a module, so it is easy to manage the device.

Eclipse Ditto utilizes the five protocols described above for communication with IoT devices and abstracts processes such as control and data collection between multiple IoT devices so that IoT devices can be easily handled through API requests from clients or servers. Eclipse Ditto expresses connected IoT devices in the real world with Thing and expresses the parts that IoT devices can handle with features. In this paper, IoT devices load measured environmental information, such as temperature, humidity, and CO2, into a separate database, define each IoT device based on this, and register and manage devices according to the following format to register them in Eclipse Ditto.

D. Smart Farm Twin Server

Smart Farm Twin Server has a function to store data collected from IoT Gateway. Multiple data collected through multiple devices and the Ditto hardware platform is expressed in units of virtualized objects expressed in software from the client to the user through a visualization tool. The basic architecture of the server implemented in this paper is to load and manage data collected from multiple devices to design a database that can map data appropriately to the client, and support it, it is implemented with Django based on CRUD logic implementation.

In order to implement a digital twin, it is necessary to determine where the virtualized software object should be placed within the virtualized greenhouse. So, in the case of a smart farm, you must have the coordinates of the rack, and each crop must have the value of the coordinates of the rack and the number of tiers. In addition, the attribute values that crops should have (type of crop, growth stage, planting date, etc.) should also be included in the DB. The database designed to manage the above-mentioned contents is built-in to Django, and it is designed with SQLite, a file-based database that can be easily accessed through the ORM.

III. Implementation Results

In this paper, we designed a framework for realizing a digital twin system for plant factories and developed it in the form of a prototype. It was verified whether or not was visualized. For this system design and development, in this thesis, an empirical test for data collection was conducted at a plant farm in Gyeonggi-do, and the main data collection period was about one year from January 2021 to December 2021. The mainly collected data were environmental data such as temperature, humidity, and CO2. The IoT device used for collection used a general-purpose hardware platform, Raspberry Pi, and the development language was C and javascript.

In this study, a digital twin framework that can be used in plant factories was designed, and an interactive system was developed by linking pre-rendered virtual images to virtual objects according to real-time coordinates for digital twins. The system screen for the digital twin that collects data in real time and expresses it on the web through the client is shown in the figure below (Figure 5).

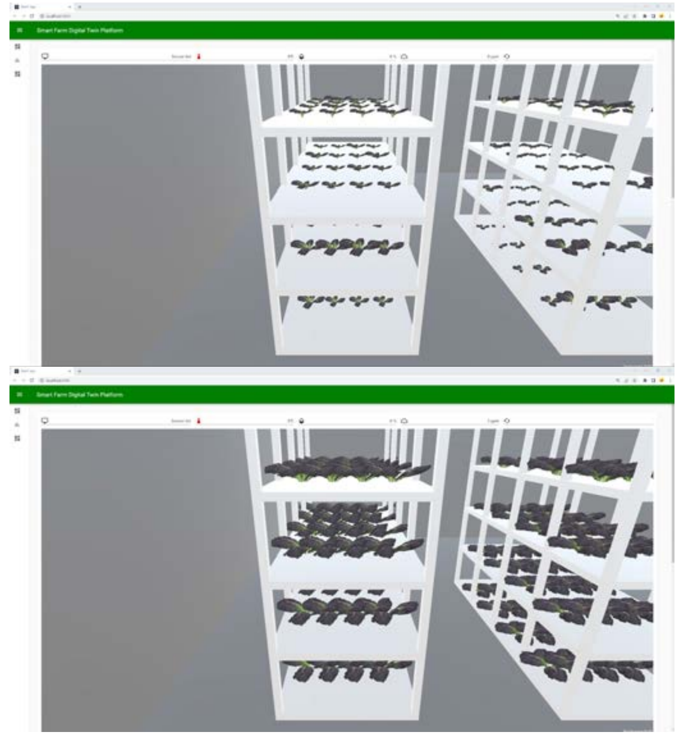

Fig. 5: The Implemented Results of Digital Twin.

IV. Conclusion

In this paper, when implementing a smart farm digital twin, virtual space and virtualized software objects are implemented in Unity before connecting to the real world, and even the stage of distributing them to a web platform for easy access. In particular, we researched the framework necessary for the development of the smart farm digital twin system, designed and developed the interface necessary for communication between each module within the framework, and collected data from a plant factory located in Gyeonggi-do based on Raspberry Pi and linked it to the actual object.

In future research, we plan to conduct research on the digital twin framework advancement plan. In particular, to overcome the asynchronous limitation of Django used for server implementation, we plan to implement a server based on Node.js, and Eclipse used as an IoT gateway. We plan to develop the system and verify the performance by applying Eclipse Hono, which is additionally provided along with the enhancement of functions for Ditto [6].

Acknowledgment

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) through Smart Farm Innovation Technology Development Program Project, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA), Ministry of Science and ICT(MSIT) and Rural Development Administration(RDA) (421034-04-1-CG000)

References

[1] Rafael Gomes Alves at, el, "A digital twin for smart farming", 2016.12.27. 2019 IEEE Global Humanitarian Technology Conference (GHTC), Oct. 2019.

[2] Melanie Jans-Singh at, el, "Digital twin of an urban-integrated hydroponic farm", Data-Centric Engineering, vol.1, 2020

[3] A. Ghandar, A. Ahmed, S. Zulfiqar, Z. Hua, M. Hanai and G. Theodoropoulos, "A Decision Support System for Urban Agriculture Using Digital Twin: A Case Study With Aquaponics," in IEEE Access, vol. 9, pp. 35691-35708, 2021

[4] D. Mitchell et al., "Symbiotic System of Systems Design for Safe and Resilient Autonomous Robotics in Offshore Wind Farms," in IEEE Access, vol. 9, pp. 141421-141452, 2021

[5] noh at, el, Communication protocol between smart farm greenhouse integrated controller and sensor-Actuator integrated node (TTAK.KO-10.0943), 2016.12.27.

[6] Eclipse HONO, https://www.eclipse.org/hono/