

Design of a Component-based Plant Factory Management Platform

Aekyung Moon, Sangho Lee, Kyuhyung Kim

Embedded System Research Team, Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, KOREA
{akmoon, comman35, jaykim}@etri.re.kr

Abstract—In this study, we introduce convergence technologies of agriculture and IT that allow flexible development of greenhouse management services for different types of plant factory. There are various types of plant factory such as fully artificial light-type, sunlight-type and plastic house, etc. Currently, almost sunlight-type plant factory in Korea uses Priva as environment control system. However, it is expensive, but also difficult for user to change and integrate different sensors and actuators. In addition, most of the typical plant factory control systems have mostly been considered a specific type of plant factory. With this viewpoint, we propose a component-based methodology for providing various types of plant factory with flexible service development and deployment. A component-based platform developed by ETRI applies fully artificial light-type plant factory for monitoring and controlling microclimate of indoor environment. Finally, we develop a small plant factory prototype for a kind of fully artificial light-type plant factory in order to validate component-based service development methodology of plant factory. In the future, we implement sensor and actuators components for greenhouse of sunlight-type. We expect that the proposed service development methodology will be effective service deployment for various types of plant factory.

Keywords—plant factory; greenhouse; environment control.

I. INTRODUCTION

The plant factory is a facility that aids the steady production of high-quality vegetables all year round by artificially controlling the cultivation environment (e.g., light, temperature, humidity, carbon dioxide concentration, and culture solution), allowing growers to plan production. There are various types of plant factory such as fully artificial light-type, sunlight-type and plastic house, etc. In the last several decades, greenhouse environment control systems for plant factory have been greatly developed. However, most of the typical systems have mostly been considered a specific plant factory types. Not only plant factory have many different types of sensors and actuators, but also environment control services are becoming more and more sophisticated considering correlation of microclimate information and autonomous control of actuator. Furthermore, each their services are often not reusable even in slightly different environment monitoring applications through sensors such as temperature and humidity.

In this paper, we design a component-based architecture of plant factory management platform. A component is a reusable and replaceable software module that enables complex functions to be developed easily. The main focus of component-based development is concerned with the

assembly of pre-existing software components into larger pieces. Nevertheless, software reuse and component-based development are not yet state-of-the-art practice software development approaches in plant factory. We introduce a component-based methodology that can support to provide a variety of plant factory type with the flexible service deployment according to this platform. Component-based development is expected to shorten the development period, reduce maintenance costs, and improve program reusability and the interoperability of components.

The remainder of this paper is organized as follows. Section 2 presents the background of this research. Section 3 proposes system architecture and component-based service development methodology for plant factory. Section 4 describes case study of fully artificial light-type greenhouse. Finally, summary and concluding remarks are presented in Section 5.

II. BACKGROUND OF RESEARCH

There are various types of plant factory such as fully artificial light-type, sunlight-type and plastic house, etc. The fully artificial light-type plant factory also controls the closed indoor light environment using artificially light without the light of the sun as well as production cycle. The sunlight-type plant factory uses the light of the sun mainly, but supplements auxiliary lighting such as LED and fluorescent light.

The plant factory control system has a role to control the greenhouse environments; it gathers microclimate information around the crops from various sensors indoor parameters such as light, temperature, humidity, and CO₂ density as well as outdoor temperature, sun light, wind and rain. And it provides appropriate control functions. Figure 1 shows the example of sensors and actuators for plant factory. The most of the typical systems have mostly been considered a specific plant factory type. Not only plant factory have many different types of sensors and actuators, but also environment control services are becoming more and more sophisticated considering correlation of microclimate information and autonomous control of actuator.

TABLE I. EXAMPLE OF MAIN SENSORS AND ACTUATORS AS PLANT FACTORY (GREENHOUSE) TYPES

Type	Example of Sensor/ Actuators Functions	
Fully artificial light Plant factory	Sensors	Indoor: Temperature, Humidity, Light, pH, EC, CO ₂ density, Leaf Temperature, etc.
	Actuators	Cooling, Heating, Artificial light, CO ₂ dosing, Water supply, Irrigation,

type		Circulation fan, Ventilation fan, etc.
Sunlight type	Sensors	Indoor: Temperature, Humidity, pH, EC, CO ₂ density, Leaf Temperature, etc. Outdoor: Temperature, Sun light, Wind, Wind speed, Wind direction, Rain, Snow, etc.
	Actuators	Cooling, Heating, Supplemental lighting, CO ₂ dosing, Water supply, Irrigation, Ventilation(Roof Vents, Side vents), Contains(Shading, Keeping warm), Misting, etc.
Plastic house	Sensors	Indoor: Temperature, Humidity Outdoor: Temperature, Sun light, Wind, Wind speed, Wind direction, Rain, Snow, etc.
	Actuators	Heating, Side Vents, Contains(Shading, Keeping warm), Irrigation, Supplemental lighting

Many research projects are tried to improve of existing plant factory (or greenhouse) systems. However, most of the typical systems have mostly been considered a specific plant factory types. Only exception is IntelliGrow [1] and BipsArch [2] by Aaslyng. IntelliGrow have functions as an addition to a generic Environmental Control Computer (ECC). The communication between PC and ECC was handled with a systems integration interface called BipsArch. But, this research has no consideration methodology for component and service development by composing.

III. COMPONENT-BASED SERVICE DEVELOPMENT

A. System Architecture

One of the benefits of growing crops in a plant factory is the ability to control all aspects of the production environment. This greenhouse management system has following considerations:

First, to optimize the environment for crop growth automatically, it can gather environment context information from various sensors and can provide appropriate control functions using various actuators. Second, it has limited easy installation and extension ability because most of the typical systems have mostly been developed based on the target specific plant factory. As a result, maintenance cost is increased. In particular, almost sunlight-type plant factory in Korea uses Priva [3] as environment management system. It is expensive, as well as difficult for user to change and integrate different sensors or actuators. Third, the microclimate control in a plant factory is a complicated procedure since the related variables are several and dependant on each other [4]. For example, the regulation of temperature can use heating, ventilation, cooling and water fogging. And ventilation control is calculated by wind speed and direction when weather station or the related sensors are installed.

To meet above mentioned considerations, we design plant factory management platform that divided into three major systems, IMS (Integrated Management System for

greenhouse), GOS (Greenhouse Operation System) and GC (Greenhouse Controller) as shown in Figure 1.

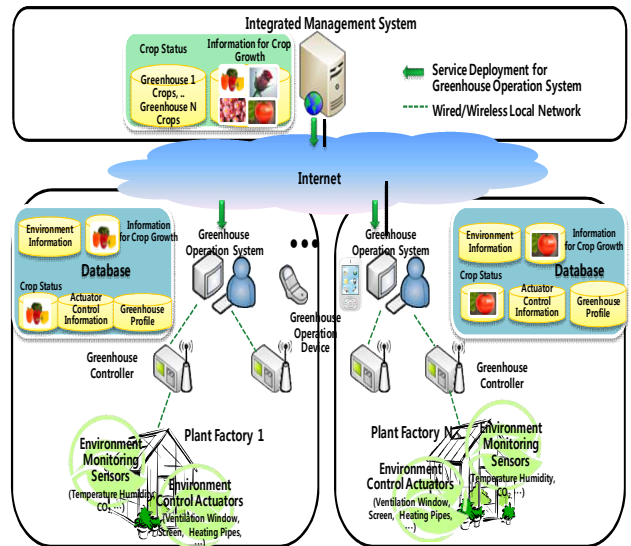


Figure 1. System Architecture

IMS deploys software components for them that sensors and actuators are installed in each actual plant factory. The communication of IMS and GOS uses TCP/IP protocol. IMS have databases to store crop growth environment information for a variety of crops. Database also stores crop status information received from GOS.

GOS includes the functions of internal and external environmental monitoring, life cycle management of sensor node and actuator node, and fault management and so on. For this purpose, GC gathers internal and external environmental contexts, sends control commands to the actuator. GOS installs services for agriculture and controlled horticulture received from IMS. In addition, it provides optimized environmental control services by feedback on the growth data of crops.

GC is consisted of embedded board with some control logics. GC supports the followings functions.

- Gathering contextual information from various sensors installed in plant factory environments. The communication of GC and sensors/actuators uses wireless protocols such as WiFi and Zigbee.
- Delivering microclimate information to GOS.
- Transferring control message from GOS to actuators
- Converting wireless protocol

B. Maintaining the Integrity of the Specifications

Plant factory management platform for a variety of operating systems has a role to install composed services according to sensors and actuators a installed on site. System provider / facility installation contractor consists of composed services using component registry of IMS as well as create new sensor / actuator components into component

registry. Figure 2 shows the component-based service development methodology.

Components for new facilities (Sensor or Actuators) are required for the development and the produced components are registered to component repository by system developer or facility installation contractor. Obviously, component repository is helpful if we can utilize existing components when making a new component in that this reduces the development time and errors that might occur when creating the component from scratch.

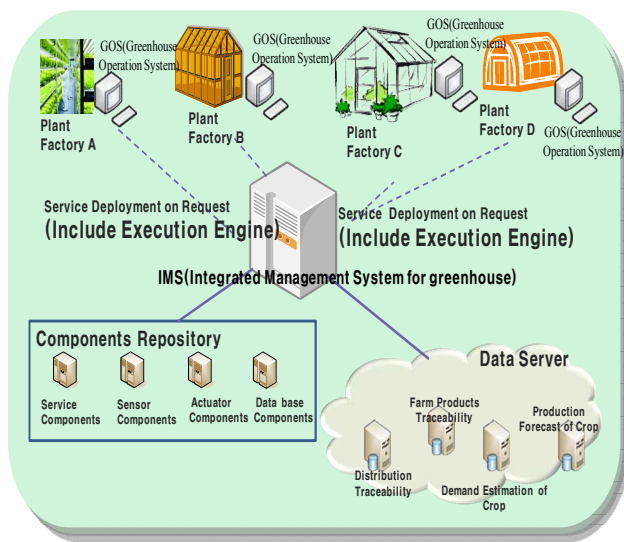


Figure 2. Component-based Service Development

IV. DESIGN OF COMPONENTS

We use OPRoS (open platform for robotic services) [5][6] in order to develop component. These tools run as plug-ins for the eclipse IDE style. OPRoS supports the full development lifecycle for robot software by providing a robot software component model, component execution engine, various middleware services, development tools, and a simulation environment. Not only do robots have different types of sensors and actuators, but also their services are becoming more and more sophisticated allowing them to run autonomously [7]. It is similar to plant factory. For this reason, we apply the OPRoS to develop the greenhouse management system.

A. Fully Artificial Light Type Plant Factory

For verify the possibility of OPRoS to develop the greenhouse management system, we develop small prototype for a kind of fully artificial light-type plant factory. Fully artificially light plant factory for testing consists of two separate individual chambers. The specification of chamber is 2000 (W) x 1900 (D) (mm). Each chamber is installed temperature sensor, humidity sensor, light sensor, CO₂

sensor, pH sensor and EC sensor for collecting environment context. And for microclimate control of indoor environment, it is installed heater, cooler, ventilation fan, CO₂ generators, humidifiers and dehumidifiers. The light sources are LED and fluorescent light, respectively.

The entire system switches automatically according to the manual and automatic switch of control box. In the case of automatic, actuator can be controlled via the 485 communication protocol. The GOS is deployed software which is developed by component methodology and can control the two chambers; as well as a GUI made by component is installed to monitor microclimate of this chamber by several sensors.

B. OPRoS based Component Development

We use two tools provided by OPRoS that run as plug-ins for the eclipse IDE (Integrated Development Environment) style [5]. One is component authoring tool. The user needs to specify the port interfaces, callback functions, and a component profile when making an atomic component. The component authoring tool helps users to add implementations of callback functions and user-defined codes without any concern regarding various relationships between port interfaces and conformances defined in the component model, for example. The component authoring tool runs as a plug-in into the eclipse C/C++ development tools (CDT). It supports the GCC and Microsoft Visual C++ compilers. Figure 3 shows OPRoS based component that we designed. And we make components using component authoring tool.

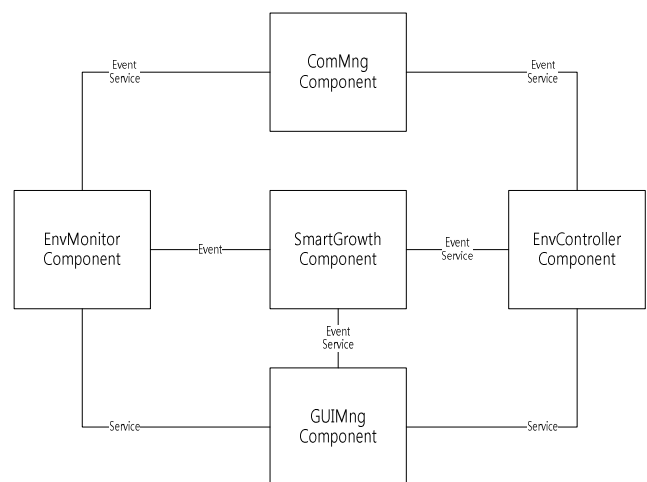


Figure 3. Design of OPRoS based Components

- SmartGrowth component refers to values of sensor node and controls actuators according to setting values for the regulation of the indoor environments in chambers. It is focus on creating the most appropriate microclimate for the maximization of crop growth.

- ComMng component has a role to communicate sensor node/actuate node with RS-485 protocol. We use wired communication protocol.
- EnvController component provides abstracted services of the actuator node to control a variety of environment.
- EnvMonitor component provides abstracted services of the sensor node to monitor a variety of environment.
- Finally, GUIMng component provides graphic user interfaces on monitoring services for indoor microclimate environment and outdoor environment as well as controlling services for regulation of indoor microclimate environment.

C. Component Composer

After developing components (SmartGrowth, ComMng, EnvController, EnvMonitor, GUIMng) for test plant factory using component authoring tool, the component composer is used building applications by composing components. It has a local repository to store components and imports component package for the component authoring tool [5]. A composite component can also be created by putting individual components into the composite component and connecting their ports to those of the composite component. The application developer drags and drops components onto

the main diagram and connects components to build an application, as shown in Figure 4. Applications are composed with combinations of components according to message flow. Components communicate with each other via connections. A connection is established from a port of a sending component to a port of a receiving component. OPRoS provides inter-component communication for sending or receiving three types of information: method invocation, data, and events.

A service port allows other components to invoke its methods. It has an interface definition of a set of methods. A service port is either a provided or required type. A provided service port provides method services to other components. A data port is for exchanging data. It is either for input or output. An output data port sends data to input data ports of other components. Both the input and output should be of the same data type for a data exchange. An event port is for transmitting events. Although data ports and event ports are similar in that they transmit structured data, events are processed immediately. Figure 4 shows connection of components. The circle represents service ports and the rhombus represents event ports.

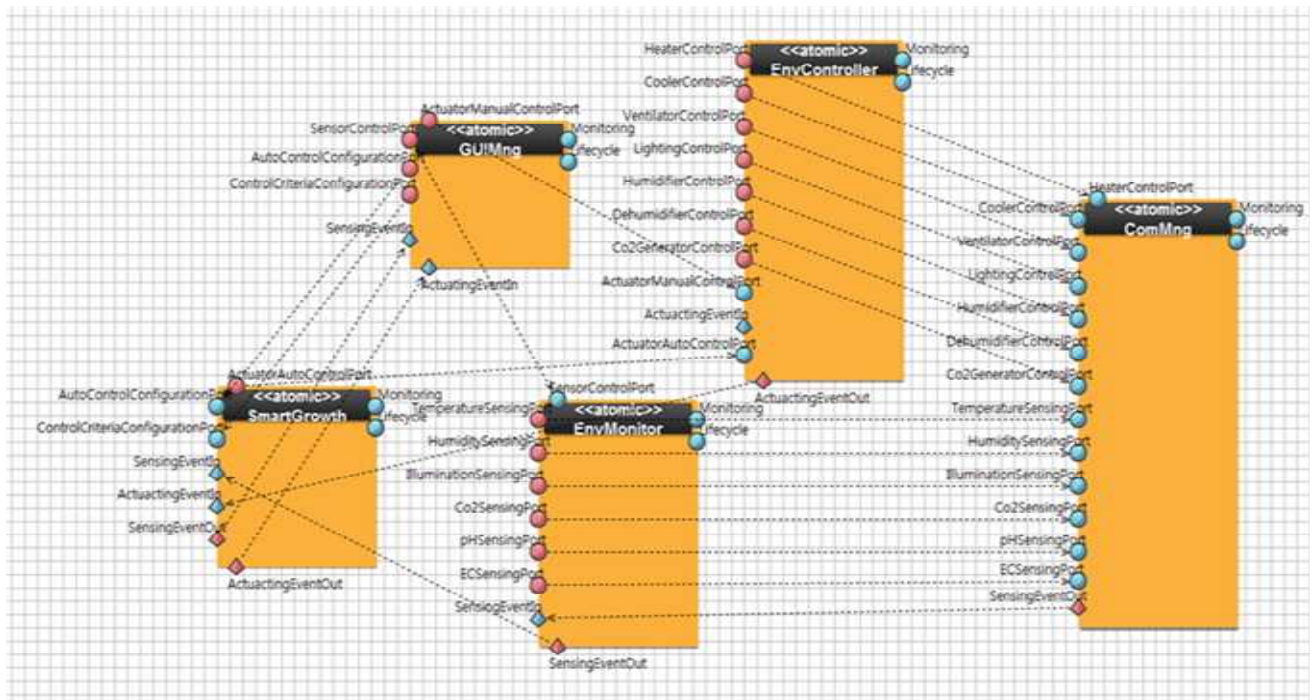


Figure 4. Component Composer for Application

The composed application will deploy the target the operating system after build. Finally the application profile (.xml) and components (.dll) are packaged and deployed to component execution engine on target operating system of plant factory via a network.

D. Experiment Results

Developed applications using environmental monitoring and control components are installed in the fully artificially light plant factory previous mentioned.

EnvMonitor component gathers temperature, humidity, illumination, CO₂ environmental information from sensor in test plant factory through ComMng communication component. And environmental information is stored in database. SmartGrowth component controls actuator based on information collected environment according to the service logic. Consequently, by adjusting temperature / humidity / photoperiod / CO₂, we grow lettuce (scientific namen : *Lactuca sativa*).

V. CONCLUSION REMARKS

We proposed a component-based methodology for providing various types of plant factory with flexible service development and deployment. And component-based platform developed by OPRoS applies fully controlled artificial light-type plant factory for monitoring and controlling of microclimate indoor environment. Finally, we develop small plant factory prototype for a kind of fully artificial light-type plant factory. Developed applications using environmental monitoring and control components are installed in the fully artificially light plant factory. It is shown that potential capability of OPRoS component-based software platform. In the future, we also implement sensor and actuators components for plant factory of sunlight-type. Eventually, we expect that the proposed service development methodology will be effective service deployment for various types of plant factory.

REFERENCES

- [1] J. Aaslyng, J. Lund, N. Ehler, and E. Rosenqvist, "IntelliGrow: a greenhouse component-based climate control system," *Environmental Modeling & Software*, 18, 2003.
- [2] J. Aaslyng, N. Ehler, and L. Jakobsen, "Climate Control Software Integration with a Greenhouse Environmental Control Computer," *Environmental Modeling & Software*, 20, 2005.
- [3] D. Kolokotsa, G. Saridakis, K. Dalamagkidis, S. Dolianitis, I. Kaliakatsos, "Development of an Intelligent Indoor Environment and Energy Management System for Greenhouses," *Energy Conversion and Management*, pp. 155-168, 2010
- [4] Priva, <http://www.priva-international.com/en/>, May, 2012
- [5] C. Jang, S. Lee, S. Jung, B. Song, R. Kim, S. Kim, and C. Lee, "OPRoS: A New Component-Based Software Platform," *ETRI Journal*, 32(5), 2010
- [6] OPRoS, <http://www.opros.or.kr>, May, 2012
- [7] C. Jang, S. Kim, M. Roh, and B. Seo, "Issues and Implementation of a URC Home Service Robot," *16th IEEE Int. Conf. Robot Human Interactive Commun.*, 2007, pp. 570-575