Complex Event Processing Methods for Greenhouse Control

Yunsong Jia and Xiang Li *

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China; jia.yw@cau.edu.cn
* Correspondence: cqlixiang@cau.edu.cn; Tel.: +86-1561-136-2365

Abstract: In the greenhouse environment, the application of complex event processing (CEP) technology can effectively tackle the problem of recognition of the complex patterns that appeared in greenhouse conditions. In the existing research, few scholars have proposed a scheme to integrate complicated scenes within the greenhouse environment with high efficiency, convenience, and low coupling. Therefore, in order to solve the problem of hard recognition and fusion of complex patterns in the greenhouse environment, based on the characteristics of the greenhouse, this paper proposes a complex event processing method for greenhouse control. Our method has high applicability and high expansibility, including 13 types of event processing agents and 21 types of typical events involved in greenhouse automatic control. This method has the advantages of low information coupling and multi-domain integration, which can be directly used by agricultural experts and related workers and is of great significance to promote the extensive application of CEP technology in the greenhouse field. Our experiment successfully realized a greenhouse intelligent control system based on CEP technology is successfully realized in our experiment. The experimental statistics shows that the structure of the control system was accessible and effective.

Keywords: greenhouse control; complex event processing; mixed automata

1. Introduction

With the development of smart agriculture, a new generation of sensor technology has been further integrated with agricultural production [1]. Consumers in urban cities have an increasing demand for fresh vegetables [2]. As a result, the greenhouse, serving as production equipment, is playing an increasingly momentous role in modern agriculture. There are many influencing factors in the greenhouse environment, which makes the greenhouse have the characteristics of nonlinearity, strong coupling, large inertia, and multi-disturbance [3]. In addition, the greenhouse environment and crops will change with the seasons and the market, with uncertainty, so many control parameters in the greenhouse still need to be dynamically identified [3]. Therefore, it is necessary to collect and summarize knowledge from the agricultural field and combine the agricultural expert system with the greenhouse automatic control technology to realize automatic reasoning and determine the control parameters and control strategies suitable for crop growth [3] in order to further realize the intelligent automatic control of greenhouses.

To facilitate the process from decision-making to control, it is necessary to use computers to describe greenhouse scenes [4]. A greenhouse is a typical complex system [3] with a variety of data types. It is difficult to describe complex scenes in a greenhouse using the traditional threshold control method with a single parameter expression. Only multi-parameter fusion expressions of multi-parameter fusion can meet the needs of planting personnel [5]. For example, the opening and closing of the skylight, air conditioner, and wet curtain need to be considered in the regulation of temperature. These controllers will also affect other factors such as carbon dioxide, humidity, and light in the greenhouse, and then the temperature, humidity, and light outside the greenhouse also need to be considered. To realize the adjustment of complex greenhouse scenes, a reasonable approach is to separate
the execution of scene detection from the response decision algorithm [6]. Therefore, we know whether the expected complex scene in the greenhouse has occurred before the response decision, and then drive the response decision and execution. When describing the above scenario, it is very necessary to find composite data with numerical, temporal, and logical significance in the data stream obtained by the sensor.

In traditional multi-variable control systems, since the weak connection between variables is often ignored, the use of multiple subsystems for separate control [7] will lead to obvious control oscillation and other problems. Multivariable control systems based on fuzzy control [8] and neural networks [9] often attempt to calculate multiple types of variables to calculate at the same time, resulting in difficulties in maintenance and modification and poor interpretability. Traditional multivariable control system based on production rules [10,11] realizes greenhouse control by explicitly defining control rules, which has the advantage of strong interpretability [11], but it is difficult to design and modify rules due to the common coupling between rules [12]. Rules will also become very complex due to the necessity of expressing complex scenarios through simple logic such as production and or not. For example, when the external sensor detects an increase and subsequent decrease in air temperature, a steep decrease and subsequent increase in air pressure, and a sharp increase in humidity, a hailstorm is expected [13], and the greenhouse skylight needs to be closed. If the production rule is used to judge whether this scene occurs, it is necessary to determine whether the values of multiple sensors show these directional change characteristics at the same time, calculate the times when the values of each sensor begin to change, and judge the sequence of these times to know whether the complex scene occurs. An event-based rule system uses formal language to standardize the relationship between events, has a strong ideographic ability [14], and can greatly simplify the expression of rule logic [15]. It is also effective at analyzing the complex scene of photosensitive sensors mentioned above. An event-based rule system can judge whether the value of each sensor has the characteristics of directional change asynchronously and in real-time, resulting in events such as “rapid rise in external humidity”. The tracking of the above scenarios can then be completed through an event filtering rule similar to “(outside temperature rises) before (outside temperature drops) and ((outside air pressure rises rapidly) after (outside air pressure drops rapidly)) and (outside humidity rises rapidly)”. An event processing system using the event itself as the carrier for information transmission also has the characteristics of high cohesion and low coupling, which makes most modules reach the low coupling degree of stamp coupling [12], and produces defined rules that are easier to modify and improve.

The occurrence of an event corresponds to the change of some states of the physical world [6]. Whether an event occurs can be judged through data flow filtering, timing analysis, and logic analysis. To monitor complex scenes, a feasible method is to regard the emergence of complex scenes in a greenhouse as the occurrence of an event, record the data flow characteristics of the scene according to expert experience, detect the event in the existing data flow, and then guide greenhouse control. However, the conditions for the occurrence of the above events are very complicated. It is difficult to detect their occurrence directly through the original data stream. It can be assessed only by further testing the timing and logic relationships between a series of simple events. This method of detecting complex events through simple event groups is called event aggregation. As an efficient event stream processing technology, complex event processing technology can effectively realize the monitoring and aggregation of complex events.

Complex event processing (CEP) is an active data processing technology born in the field of databases [16]. It regards a change in a single data state as an atomic events and the emergence of complex scenarios as the occurrence of complex events. It detects the occurrence of complex events by matching the occurrence of a large number of atomic events with predetermined complex event patterns and logical calculations, which trigger further actions [17]. CEP technology can make full use of information from different approaches to make decisions on a particular situation. It offers strong asynchronous
decoupling and situation analysis ability and is an important means to realize real-time processing and automation. Therefore, it is an important method that can solve the problem of multi-temporal event flow processing [18]. It has been widely used in internet of things scenarios such as industrial production lines [19], smart homes [20], intelligent logistics [21]. However, so far, the research on agricultural-oriented CEP has remained at the basic application of the Internet of Things and the expansion of theoretical operators. It has not yet formed a complete and versatile method for constructing a CEP system.

This paper applies CEP to intelligent control in the greenhouse environment and builds an intelligent system for greenhouse control. This paper takes the intelligent control of the greenhouse as the core of the research, analyzes the information generation role and transmission characteristics, and encapsulates the main information in the greenhouse environment. A theoretical scheme of greenhouse CEP system with high applicability and high expansibility has been obtained to solve the key difficulties in the application of CEP theory in the greenhouse.

The main contributions of this work are as follows:
1. This article innovatively applies CEP technology to the intelligent control of agricultural greenhouses. By analyzing the commonalities and differences of many agricultural issues, we propose an effective, greenhouse-oriented complex event processing system structure. It is of great significance to promote the extensive application of CEP technology in the greenhouse field;
2. We have improved the automata algorithm for event aggregation, which allows CEP technology to be used more flexibly in complex greenhouse scenarios. Our algorithm includes 13 types of event processing agent types, 21 types of typical events involved in greenhouse automatic control, which can correspond to the seven most common specific agricultural problems;
3. This work incorporates knowledge in related greenhouse research fields such as greenhouse planting schemes. It gives full play to the advantages of various control methods to deal with the different control characteristics that need to be solved in the greenhouse environment.

2. Related Work

Since the 1940s, FW Went presided over the construction of the world’s first plant artificial climate chamber, which brought a revolutionary change to the field of facility agriculture. In the early period, greenhouse control algorithms often simply used threshold control based on farming experience [22]. In the traditional threshold control system, due to the large coupling of various factors in the greenhouse environment, the regulation of a single environmental factor often affects other factors in the same closed space. For example, to reduce the temperature, the optional control strategy is to open the sunshade, but this measure will correspondingly reduce the number of lights [23]. Limited by the weak representation ability of the threshold control method, it is difficult to manage complex scenes and the interaction between multiple environmental factors. Therefore, an algorithm with stronger regulation ability is essential.

Over the years, the regulation methods to multiple environmental factors in complex scenes are basically divided into two kinds: the control strategy based on the physical model and the intelligent control algorithm based on experience and data [24]. The essence of using the physical method is to establish a physical model of a greenhouse through mechanism analysis and use a multivariate decoupling method to achieve control [25]. More specifically, control strategies include robust controls proposed by Linker [26] and Bennis [27], nonlinear controls used by Pasgianos [28] and Wu Xiuhua [29], adaptive controls used by Cunha [30], Guzman [31], Ahmed [32], etc., and model prediction control, etc. used by Camacho [33], Ramirez [34], Berenguer [35], etc. However, the greenhouse is a complicated system that has nonlinear multi-inputs and multi-outputs [36]. Factors near the greenhouse will affect the physical and chemical properties of its internal environment. Additionally, some of these factors are difficult to accurately measure or model. Besides,
different physical parameters in the greenhouse will also fluctuate due to the aging of the equipment, resulting in deviations between the parameters calculated by the decoupling control and the actual ones [37]. Therefore, it is hard to establish a long-term effective physical model suitable for a greenhouse environment. All in all, for the traditional linear parameter model, even if it performs well in a specific greenhouse, it is difficult to apply effectively in a wide range of greenhouses [38].

The greenhouse intelligent control algorithm based on experience and data combines the analysis of historical data and the experience of expert planting. It mainly includes control algorithms based on fuzzy logic and control algorithms based on machine learning. This algorithm is not limited to the complete physical model of a specific greenhouse when it is defined.

The fuzzy logic control algorithm was first proposed by Zadeh et al. [39]. This algorithm solves the problem of uncertain reasoning in the control system by transforming the expert construction language information into control strategies [40]. However, when applied to the time-varying nonlinear system of the greenhouse, the fuzzy control algorithm must define very clear control rules in advance in order to achieve a good control effect [41]. As the controlled process of a nonlinear system has the disadvantages of nonlinearity, high order, and time variability, it is therefore intractable to use the existing control theory and analysis methods to design a fuzzy control system [42].

Machine learning [43] is a type of algorithm based on statistical theory that can directly learn relevant laws from data and is usually used in regression and pattern recognition [44]. In recent years, machine learning algorithms have also achieved good results in greenhouse control. Whether directly using artificial neural networks [9] or combining it with PID control technology [45], WDNN technology [46], machine learning methods have achieved amazing achievements in the field of temperature, disease, and pest control in greenhouses [47]. However, it needs to collect a large amount of data in the same scene for training, and the output results are not scientifically interpretable. Therefore, it is difficult to maintain satisfactory results when directly migrating to a similar greenhouse scene.

CEP techniques were first emerged from the active database domain [48]. It can properly describe the connection between the monitored event and the execution command, so as to more accurately locate the conditions under which a specific response action occurs [49]. With the development of active databases, researchers have gradually realized that the logical relationships between events are not only limited to simple “and”, “or”, and “not”, but also intricate as temporal relationships. In 1992, Gatziu S. etc. proposed the SAMOS Active Database System [50], and operators related to timing logic [51]. In 2008, D. Gyllstrom et al. proposed the SASE+ improvement system and applied the green closure operator for repeated events [52]. In the same year, Magid Y. et al. also strictly defined a series of operators, applying scenario description methods to complex event models [53]. With the vigorous development of computer technology and the exponential growth of data, event detection methods emerged one after another [18]. Among them, CEP technology based on rule description stands out because of its excellent multi-data processing function [54]. It has achieved remarkable success in many fields such as industry [55] and commerce [56]. In current practice, CEP technologies often use graph structure model [57], automata model [58], matching tree [59], Petri net [60], event pattern language [61], and other methods for event pattern matching. An automatic machine to realize event pattern recognition is the most representative and excellent way to realize a high-performance CEP system [3].

In the agricultural field, Bertha et al. [62] proposed the application of complex event processing engines to build peripheral systems. Li et al. [63] conducted a detailed analysis of the spatiotemporal event model in farmland. Deng et al. [64] also made relevant attempts for the use of non-CEP structured time automata in the greenhouse environment. Li et al. [14] propose a space-time event model for the processing of complex events of the farmland Internet of Things and extends the logical relationship of complex events to the definition of spatial topology relationship. Nie Juan et al. [5] proposed a new type of CEP
engine called PUCEP, which can calculate the probability of complex events. It improves the real-time performance of the system by improving the matching efficiency of uncertain complex events, reducing computing time and resource consumption. Mazon et al. [62] proposed the RE-CEP structure for precision agriculture and applied it to the intelligent irrigation system of the banana test area in Macara, Ecuador, which confirmed the feasibility of CEP technology in agricultural production. However, the above-mentioned research mainly revolves around the model design and feasibility analysis of complex event processing technology. Research on the overall control of greenhouses is scarce.

3. Complex Event Processing Algorithm for Greenhouses

3.1. Overall Introduction

To realize intelligent greenhouse control, the CEP system needs to regard the fluctuation of different types of data in the greenhouse as the most basic atomic events at first. Additionally, it should continuously match and calculate to generate more complex events by analyzing the logical relationship between basic events. Some of the complex events contain the control commands and complex scene information needed in the greenhouse environment. The goal of our design is to provide a CEP system with simple data analysis capabilities and a complex event processing system framework that can yield control instructions based on expert commands or automatic control schemes. The core work is to design and implement a complex event processing engine.

Figure 1 shows the connection between the system and other objects in the greenhouse. These objects are sensors, feedback units, experts and other analysis systems, control strategies, and actuators in addition to the CEP system and greenhouse. Among them, different functional sensors include air humidity sensors, soil sensors, etc. The feedback unit and actuator can work on Internet of Things software platforms such as Fiware and ZigBee network facilities to transmit the data of sensors and controllers. Other analysis systems could be PhenoArch or other phenotype analysis platforms, which can generate meticulous and ample data information. The control strategy could be a simple switching operation or a more complex timing control mechanism. When a control strategy needs to be executed, the state of the controllers could be modified according to the preset instructions.

In detail, the sensors are responsible for collecting the physical and chemical information in the greenhouse and transmitting the original data to the feedback unit, agricultural experts, and other analysis systems to achieve preliminary processing. The feedback unit packs environmental data into events (atomic events) and transmits them to the CEP system. Agricultural experts and other analysis systems conduct preliminary analysis based on actual conditions. The feedback unit, agricultural experts, and other analysis
systems package the analysis results into events and send them to the CEP system. When the CEP system receives this data and the event containing equipment status information fed back by the actuator, it will be transmitted to different processing modules (event processing agent, EPA) in the CEP engine through the event distribution middleware. EPA will analyze the input events in time sequence and logic according to the predetermined CEP rule set, so as to get high-density complex scene information. The identified complex scenes can eventually be fed back to agricultural experts and other analysis systems and mapped to the corresponding control strategies. In this way, the closed-loop control of the greenhouse is realized by the whole system.

For the convenience of description, according to the functions of the above objects, they are divided into four major information objects: CEP system (left, dark gray), controller (above, red), worker (middle, blue), and environment (below, green). In the CEP system, all input and output information will be packaged into events and delivered through event distribution middleware. The CEP engine runs according to the designed CEP rule set, including filtering, aggregating, and generating events. Nowadays, middleware technology represented by Kafka [65] has matured, so this article focuses on the layout of CEP rules to provide reliable assistance for agricultural workers to use CEP technology in greenhouses.

As mentioned above, to build a CEP rule set, we need to stipulate the category of events, define atomic events, complex events, and the specific input and output of each type of EPA that can realize the event aggregation function, and finally constitute a complete event aggregation structure. This can reduce the repetition and incidental loss of intermediate information, thereby achieving information decoupling and determining whether complex scenes occur, and scheduling control strategies.

The following will present the general structure of the greenhouse-oriented CEP engine, the structural definition of various events, and the functional definition of EPA. It provides a general implementation of the greenhouse-oriented CEP system for agricultural experts and related workers. Subsequent experiments will be based on this structure for concrete function realization and feasibility verification.

### 3.2. Complex Event Processing Logical Relationships

In the process of judging complex scenes, it is necessary to analyze the logical relationship in time and space between low-level simple events to determine whether high-level complex events have occurred. Then, the complex event flow is synthesized by calculating the relevant information of the high-level events. The process of simple event flow evolving into complex event flow is called event aggregation. The logic between events that has been set is called event mode. The unit that performs matching and event generation based on event patterns is called an event processing agent (EPA). Through layer-by-layer aggregation, complex events are finally generated. These events can trigger control commands, get more valuable judgment information, and complete the reasoning process.

This article proposes 13 types of EPA and 21 types of typical events involved in greenhouse automatic control. The aggregation relationship between EPA and events is shown in Figure 2. The circular nodes in the figure represent events. The color of the nodes represents the information source and destination of the event (as shown in Figure 1 in the above section). Gray nodes represent the output of the event by the system. The square nodes correspond to the classes of EPA of the CEP engine, which can aggregate the input events to obtain the output events. In particular, the dashed arrows in the figure represent that one EPA module directly modifies the parameters of another EPA module. Each EPA module can be built using the existing CEP technology or can be replaced with any event processing module with the same type of event input and output (e.g., using machine learning for event classification filtering [66]). This makes the CEP system have the characteristics of multi-field and multi-technology integration.
Figure 2. The general structure of a complex event processing engine facing a greenhouse.

Each event flow from the beginning to the end corresponds to a specific farming problem. The system needs to use the existing coping strategies to build an event processing unit, analyze the current situation according to the event sequence, and transmit the analysis results to other units. The specific agricultural problems contained in the system are as follows:

In terms of environment, it is necessary to observe the changing trend of the greenhouse during the regulation (EPA2), find the most favorable greenhouse environment state for crop growth according to the crop planting plan (EPA5, EPA6, EPA3), and consider the mutual influence relationship existing in the greenhouse refinement regulation target (EPA7). The search for the most favorable greenhouse environment can be further refined into looking for the impact of the environment on crop growth (EPA5), the determination of crop growth status and the adjustment of control targets (EPA6), and the combination of control targets to judge the current state of the greenhouse (EPA3).

In terms of control, the system needs to monitor whether the connection and operation of the controller are normal (EPA8), consider the impact of the environment on the controller
effect (EPA1, EPA4), how to dispatch the controller and generate early warning information (EPA9, EPA10, EPA11, EPA12), and whether the instruction execution is smooth (EPA13).

For the convenience of description, we have given the name of the EPA that solves the corresponding agricultural issues in parentheses. In the following, we will introduce the corresponding types and definitions of each event and the implementation methods and specific functions of each EPA one by one. In the experiment, we will build an example of the CEP system based on common solutions to agricultural problems.

3.3. The Event Definition

In this paper, the state change of the complex system is called “event”. To make the structure shown in Figure 2 work smoothly, it is necessary to strictly define the type and basic structure of events transmitted between each EPA module and system input and output.

Backus Naur Form (BNF) paradigm is a formal syntax representation method, which is often used to describe program syntax [67] and data structure [68]. This paper will use the following extended BNF paradigm to define the structure of events:

- Angle brackets are used to separate strings that are the names of grammatical elements. If a triple with “e.g.,” as the second element appears in the angle brackets, the triple is a type of terminal symbol with a specific meaning. The first element is the category name of this type of terminal symbol, and the third element is some examples of this type of terminal symbol;
- The operator used for definition. Represents that the syntax element on the left can be replaced by the string appearing on the right;
- The elements in square brackets can appear 0 or 1 times;
- The vertical line indicates that the element on the left can be replaced by the element on the right.

JSON is a lightweight data exchange format. It can describe the basic structure of data through key-value pairs [69]. This article will wrap a specific data structure with a pair of braces. Each data structure is composed of multiple key-value pairs similar to “key: value”, separated by commas.

The following will use the BNF paradigm to specify the specific structure of the above events in the form of JSON string and list some specific examples of all terminations.

Most events have a unique number, type, name, timestamp, and other basic event information. Some special events may also have location and source information. Therefore, to simplify the subsequent expression, first define the basic event structure (BES) syntax as follows:

\[
\text{<BES>::=\langle\text{ID}, \langle\text{TYPE}, \langle\text{NAME}, \langle\text{TIME}, \langle\text{LOCATION}\rangle, \langle\text{SOURCE}\rangle\rangle\rangle\rangle}\]

\[
\langle\text{ID}\rangle::=\langle\text{id}\rangle: \langle\text{event unique id, e.g., } "E00121"\rangle
\]

\[
\langle\text{TYPE}\rangle::=\langle\text{type}\rangle: \langle\text{event type, e.g., } "Art"\rangle
\]

\[
\langle\text{NAME}\rangle::=\langle\text{name}\rangle: \langle\text{full name, e.g., } "temperature"\rangle
\]

\[
\langle\text{TIME}\rangle::=\langle\text{timestamp}\rangle: \langle\text{timestamp of the event, e.g., } "2001/11/20 16:25:58"\rangle
\]

\[
\langle\text{LOCATION}\rangle::=\langle\text{location}\rangle: \langle\text{location of the incident, e.g., } "outside"\rangle
\]

\[
\langle\text{SOURCE}\rangle::=\langle\text{source}\rangle: \langle\text{EPA-ID}[, \langle\text{EPA-STATE}\rangle[, \langle\text{CAUSE}\rangle]]\rangle\]
\]

\[
\langle\text{EPA-ID}\rangle::=\langle\text{EPA id}\rangle: \langle\text{ID of the EPA that generated the event, e.g., } "A001"\rangle
\]

\[
\langle\text{EPA-STATE}\rangle::=\langle\text{EPA state}\rangle: \langle\text{current state of the EPA that generated the event, e.g., } "S1"\rangle
\]

\[
\langle\text{CAUSE}\rangle::=\langle\text{cause}\rangle: \langle\text{the ID of the cause event, e.g., } "E00118"\rangle
\]

For most of the information content types of the control system, there are three main categories, namely numeric class, state class, and information class. They appear in a single numeric, enumeration, and string type. The event wrapping of these three types of information is relatively simple, just add the necessary information content to the underlying event structure. The structure is as follows:
For some special information, we need to tailor it to:

- Environmental trends, which mainly reflect environmental changes, are usually obtained by differential or forecasting of environmental information and need to indicate the start and end times, the specific structure of which is as follows:

```
<ETD>::=<BES><PERIOD><T-VALUE>
<PERIOD>::="period": ["start": <trend start time>, "end": <trend end time>]
<T-VALUE>::="value": <expected difference value | expected difference status, e.g., -5 | RISE>
```

- Control method, which simply distinguishes the type of method called by name, as follows:

```
<CMT>::=<ID><TYPE><METHOD-NAME><TIME><SOURCE>
<METHOD-NAME>::="method": <the name of the calling method, e.g., "Adjust-tem30">
```

- System configuration information, which includes executable commands and some descriptions, as follows:

```
<SMT>::=<BES><M-VALUE><DESCRIBE>
<DESCRIBE>::="describe": <explanation of the purpose of execution, e.g., "update">
<M-VALUE>::="script": <executable script>
```

For greenhouse scenes, the event is divided into three levels.

1. Input information (the simplest atomic event). This information originates from other information objects and acts on the control system and is the basis for the calculation of control system reasoning, as shown in Table 1.

2. Intermediate information (intermediate-level complex events). This message is not directly related to the input and output of the control system, but is used as a carrier of reasoning results in the control system, as shown in Table 2.

3. Output information (highest level complex events). This information originates from the control system, acts on other information objects, determines the control action of the controller, and makes it easy for workers to know about the greenhouse, which is an important part of the realization of automatic control, as shown in Table 3.

### Table 1. Input information table.

| Information Class          | Type     | Symbol | Introduction                                                      | Examples                          |
|----------------------------|----------|--------|-------------------------------------------------------------------|-----------------------------------|
| Environment information     |          |        |                                                                  |                                   |
| Air information             |          | Ari    | The system is input by sensors, prediction systems, and storage devices | Temperature, humidity, CO₂, O₂ concentration |
| Lighting information        |          | Lgi    | The system is input by sensors, prediction systems, and storage devices | Light Intensity                   |
| Soil information            |          | Soi    | The system is input by sensors, prediction systems, and storage devices | Moisture and nutrient content     |
| Weather information         |          | Wti    | The system is input by sensors, prediction systems, and storage devices | Rainfall, wind speed, wind direction |
| Control information         |          |        |                                                                  |                                   |
| The connection status       |          | Cni    | Information that tells the system controller that it is connected correctly | Air conditioning (dropped)       |
| The device status           |          | CsI    | Inform the system controller of the turn-on situation            | Air conditioning (on)            |
### Table 1. Cont.

| Information Class | Type         | Symbol | Introduction                                                                 | Examples                        |
|-------------------|--------------|--------|-----------------------------------------------------------------------------|---------------------------------|
| Worker information| Plant State  | Psi    | Used to describe the Plant state growth status of plants or disease        | The growth stage (rosette stage) |
|                   | Control Information | Cti    | Informs the system of an artificial change in the state of the controller  | Air conditioning (off to on)    |
|                   | System Configuration Information | Smi    | Inform the system of artificial modification of the system configuration    |                                 |

### Table 2. Intermediate information table.

| Type                                | Symbol | Introduction                                                                 | Examples                        |
|-------------------------------------|--------|-----------------------------------------------------------------------------|---------------------------------|
| Environmental State                 | Est    | Indicates the state of the current environmental factor                     | The temperature status is normal/high/low |
| Environmental trend                 | Etd    | Indicates the relative relationship between environmental factors and history and prediction | The temperature increases       |
| Environmental factors               | Eft    | Environmental factors related to controller control characteristics         | Temperature factors (indoor height/outdoor height) |
| Environmental Requirement           | End    | Indicates the regulatory needs of environmental factors                     | Temperature demand (rise/fall/none) |
| Control effectiveness               | Cet    | Indicates whether the instrument is currently operating effectively        | Air conditioning operation (invalid/valid) |
| Control Direction                   | Cdt    | The direction in which the controller controls environmental factors       | At this point, the fan temperature rises by 2 degrees |
| Control Method                      | Cmt    | The name of the method to call, as summarized from expert experience        |                                 |
| Control requirement                 | Cnd    | Control instructions that are prepared to be executed                      | Air conditioning (on)           |

### Table 3. Output information table.

| Information Class              | Type                  | Symbol | Introduction                                                                                           | Examples                        |
|--------------------------------|-----------------------|--------|--------------------------------------------------------------------------------------------------------|---------------------------------|
| Control command                | Action information    | Aci    | Control instructions generated by the system                                                           | Air conditioning (on)           |
| Environmental alert            | Emi                   |        | the worker that environment to stay away from the equilibrium state                                     | The temperature is too high.    |
| Instrument error message       | Cei                   |        | The information generated by the system tells the worker that the controller connection is down         | Wet curtain (disconnected)      |
| Plant state                    | Psi                   |        | The information generated by the system to inform workers of plant growth status                        | The growth stage (rosette stage) |

### 3.4. EPA Definition

This article uses automata technology to build EPA and uses transitions between states to identify the content of events and their relationships. When entering a new state, complete the corresponding instructions, modify local variables, and achieve event aggregation. There are three types of transition (or jump) between states, which are triggered by specific events, timers, and boundary conditions.

Transitions triggered by specific events can check the causal dependence between events, and whether data parameters and contextual relationships are in line with expectations. Its transitional rules include three parts: pre-transition state, post-transition state, and Boolean expression. Its Boolean expression can contain two types of parameters: the type of the received event and its parameters, and the local variables in the EPA. Event parameters include information such as the source of the event, the event ID, and timestamp. When a new event is received, the EPA will determine whether the pre-transition state in the jump rule triggered by the specific event is consistent with the current state. If it matches and the Boolean expression in the rule is true, it will transition to the post-transition state. Otherwise, no operation is performed. For all transitional rules with the same pre-transition state, the pre-transition state will change based on the current state, and the post-transition state will be determined according to the Boolean expression.
state, their Boolean expressions are mutually exclusive and should not be established at the same time.

Transitions triggered by the timer can check whether the time relationship between events is in line with expectations. The transitional rules include three parts: pre-transition state, post-transition state, and elapsed time. Among them, the elapsed time is the length of time elapsed after entering the pre-transition state. When the EPA is in a certain state, it will determine whether the pre-transition state in the transitional rule triggered by all the timers is consistent with the current state and if they match, the timer will be established according to the elapsed time. When the timer expires, the EPA will transition to the specified post-jump state. If the EPA has made a state transition before the timing is over, all current timers will be closed and no longer executed. For any state, if there are transitional rules triggered by multiple timers, the transitional rules will never be executed. Therefore, in a specific implementation, the pre-transition state of the transitional rule triggered by all timers in any EPA is different.

Transitions triggered by boundary conditions are mainly used in the EPA at the bottom of the system, and specific atomic events can be identified from the input of the system. The transitions triggered by the boundary condition will continuously check whether the input value meets the expected condition. The transitional rule includes three parts: pre-transition state, post-transition state, and the boundary condition. Among them, the boundary condition is a Boolean expression containing two types of variables, local variables, and input values in EPA. When the input values are updated, it will check whether the boundary condition is established. If it is established, the corresponding transition will be performed.

When the EPA transitions to the next state, a collection of actions required to be performed within the state is executed once. Among them, the action contains three kinds of modification of local variables, generating new events, and executing specific instructions using the event information just received.

As shown in Figure 3, an intermediate layer EPA triggers a jump without a boundary condition. The EPA has three states (State), T, L, and N, and when it enters the T state, a Cdt event is generated, with parameters Fan and RISE, the L state is the same, and the N state has no action to perform. The EPA has no arguments and there are two jump Boolean expressions. Wherein, the t-jump Boolean expression is used to determine whether the received event is an Eft type, the first argument is tem and the second argument is Inward condition, l jump takes 5 s, and if the EPA is in the N or T state for 5 s, it jumps to the L state.

![Figure 3. Example of middle EPA with boundary conditions triggering jump.](image)

For example, when the EPA is in state N and receives an Eft(tem, Inward) event, the EPA will check whether t is established. As the result is that t is established, the EPA jumps to the T state and generates a Cdt(Fan, RISE) event. Since there is a time jump l in the T state, a timer with a duration of 5 sec is established. When the timer expires after 5 s, the EPA will automatically jump from the T state to the L state and generate a Cdt(Fan, DECLINE) event. The subsequent process is the same.

As shown in Figure 4, it is an example of a bottom EPA with boundary conditions triggering a jump. It has four states, two parameters, two jump Boolean expressions, and two boundary conditions. The ra jump, ua jump, and all state actions are similar to the
middle layer EPA and will not be explained here. The EPA has two parameters from outside input. It will continuously observe the changes of these two parameters at a certain frequency. When the relationship between the two parameters matches the up or dn jump Boolean expression, the jump will be executed.

![Diagram of EPA](image)

**Figure 4.** Example of bottom EPA with boundary condition triggering jump (EPA1).

For example, when EPA is in state U, tem is 10.0, and tem_0 is 15.0, the external temperature sensor inputs a value, and tem_0 is updated to 5.0, and EPA will check whether the dn boundary condition is established. Obviously, tem_0 < tem is established at this time, so the EPA state will jump to D, and then an Eft (tem, Inward) event will be generated.

The transceiver unit for event transmission between EPAs can form a network of multiple EPAs to communicate with each other. Each EPA will respond to input events and pass the output events to other EPAs. This type of structure constitutes an EPA communication network called the event processing network (EPN). It is the key to the aggregation of complex events.

This chapter uses Kafka middleware with a publishing subscription pattern to deliver and receive events between EPA. For each type of event in the system, a message queue with the event name as the theme is established to hold all generated events of that type in the build order. When an EPA needs to send a specific type of event, it needs to find the message queue for the topic based on the event type and publish the event to this message queue. When the EPA needs to get a specific type of event, it subscribes to the message queue for the topic based on the event type and reads from the specified location in order, and then waits for the next event to enter the queue. When the system first starts, the message queue is emptied and all subscriptions are read from the header of the message queue.

The typical EPA for greenhouses is shown in Tables 4 and 5.

| Name | Function | Detailed Description |
|------|----------|----------------------|
| EPA1 | Calculate the environmental factors that affect the effectiveness of the controller | According to the characteristics of the controller, collect and calculate from the environmental information to obtain the environmental quantity that affects the effect of the controller, and generate the environmental factor event (Eft) to transmit the analysis result |
| EPA2 | Obtain the changing trend of environmental factors from environmental information | Record the current environmental event and the last environmental event processed, calculate the change value of the environmental factor and the time difference between the two events, and obtain the historical or future environmental factor change speed (change trend) |
| EPA3 | Gets the difference between the state of the environment and setting goals | Obtain the current physical and chemical characteristics of the environment through environmental information events (Ar/Hi/Lgi/Soi), then compare it to the established adjustment target, and package the comparison result into an Est event and send it out |

*Table 4. EPA function table.*
### Table 4. Cont.

| Name | Function | Detailed Description |
|------|----------|----------------------|
| EPA4 | Predict the effect of a certain control operation | According to EPA1's calculation results of environmental factors that affect the effect of the controller, the environmental impact and work efficiency of the controller during operation are estimated, and the control direction event (Cdt) transmission analysis result is generated. |
| EPA5 | Get the current crop growth stage according to the initial planting plan and current environmental conditions | According to the established planting plan, the time that has passed since entering the current stage and the changes in the environment, judge whether the crop enters the next growth cycle and whether there is disease. |
| EPA6 | Set the corresponding environmental goals through the crop growth phase | Obtain the information of the current growth stage of the plant according to the content of the plant state event, and then adjust the parameters in EPA3 according to the stage to achieve the purpose of setting the environmental regulation target suitable for the current growth stage of the crop. |
| EPA7 | Estimate the amount of adjustment that each factor in the greenhouse needs to be adjusted for | Through the status (Est), changing trend (Etd) of each factor that needs to be adjusted in the greenhouse, and the mutual influence relationship between environmental factors, the adjustment target is revised, and the environmental demand event (End) is sent to transmit the analysis result. |

### Table 5. EPA function Table 2.

| Name | Function | Detailed Description |
|------|----------|----------------------|
| EPA8 | Provide the worker with a controller connection status | According to the controller connection event (Cni) received by the system, an instrument error information event (Cdt) is generated to transmit the analysis result. |
| EPA9 | Provide the controller information needed to select the method, and judge whether a method can run normally | Filter the controller connection status and control direction information required by a method, and determine whether there are mutually exclusive methods running to determine whether a method is running well and can be performed smoothly. |
| EPA10 | Collect the environmental information required for the method and estimate whether the method is suitable for the current environmental conditions | Using the adjustment target event of the current environment, combined with the regulatory characteristics of the complete execution of the method, a series of parameters are obtained to judge the suitability of the method execution. |
| EPA11 | Score the best for each control method | By Clt and Elt events, determine which methods are suitable for running in the current environment, and consider the situation where the method cannot be executed at the same time and the worker participates in the regulation, give the serial number of the governance method that needs to be executed at present, and pass the method number to be called through the Cmt event. Method numbers are used to distinguish between different governance methods, and their values are defined in EPA12. |
| EPA12 | Perform specific control methods | Each contains a governance method that, by receiving control method events, determines whether it needs to start and stop the execution, generates a series of control instructions (Cnd) and environmental alert information (Ewi) when executing, and fine-tunes the control policy by controlling the information returned (Cet). |
| EPA13 | Receive the control request of the method, send the execution command to the control element, and ensure the correct execution of the command | Used to check the execution of a controller's control instruction, by generating an AcI event governance controller, listening to the Csi event to determine whether the instruction was executed successfully, and then feeding back the information of the success of the execution to EPA12 in the form of a Cet event, informing the worker that there is a problem with the control in the form of a Cei event when execution fails. |

### 4. Case Study

Nowadays, lettuce is the main representative of Commercial Vegetables in cities [2], also as a typical crop in North China Plain, central and Western China [70]. Therefore, lettuce was chosen as the planting crop in the experiment. The experiment includes the comparison of planting schemes, the construction of the structure, and the realization of an entire greenhouse control process through CEP. We collected and studied crop planting articles and web pages written by Shi, Sun, etc. [71–75]. We have formulated a comprehensive planting plan for Italian lettuce. The planting plan covers various greenhouse tuning indicators, which can achieve the goal of large crop yield and high quality. Some scenarios are shown in Table 6.
Table 6. Italian lettuce cultivation program.

| Stage                      | Parameter | Day Requirements | Night Requirements | Remark                                                                 |
|----------------------------|-----------|------------------|--------------------|------------------------------------------------------------------------|
| All                        | humidity  | About 70%        | About 70%          | O\(_2\) is in high demand                                             |
|                            | air       | ventilation      | ventilation        |                                                                        |
| Germination (about 10 days)| illumination | \(\approx 200 \mu \text{mol/m}^2\cdot\text{s}\) and 8 h |                    | The minimum temperature of germination is 4°C, and the germination rate of 25°C decreases |
|                            | temperature | 15~20°C         | 15~20°C            | Lack of water causes the outer lobe to grow poorly, too much easy to cause rotten roots |
|                            | moisture content | 60%          | 60%               |                                                                        |
|                            | CO\(_2\)     | \(\approx 1000\text{ pmm}\) |                    |                                                                        |

Below, through the definition of 13 types of EPA, the temperature-related control of the greenhouse is realized.

**EPA1** (Figure 4)—Take the fan control effect to calculate the internal and external temperature difference as an example: there are two types of internal and external temperature differences, which are internal low and external high, and internal high and external low. The exhaust fan is also divided into two modes: access system and disconnection. It needs to jump to generate exhaust fan control events through connection change events and temperature difference information.

**EPA2** (Figure 5)—Taking temperature as an example, there are three types of fluctuation trends in the actual scene: rising, falling, and steady. Three types of boundary conditions are defined here: the difference between the predicted value and the true value is greater than the threshold, the difference between the true value and the predicted value is equal to the threshold, and the absolute value of the difference between the predicted value and the true value is less than the threshold.

**EPA3** (Figure 6)—For most crops, for example, there are two states of optimal growth and normal growth, corresponding to the most suitable growth temperature range and normal growth temperature range, corresponding to four thresholds: the normal growth temperature nether, the most suitable growth temperature nether, the most suitable growth temperature upper bound, and the normal growth temperature upper bound.
EPA4 (Figure 7)—Take the temperature control effect of the exhaust fan affected by the temperature difference between inside and outside as an example: there are two situations. One is to increase the indoor temperature, and the other is to lower the indoor temperature. At the beginning of the system operation, the direction of temperature control by the exhaust fan can be judged by calculating the difference between the inside and outside of the temperature.

EPA5 (Figure 8)—Lettuce growth is divided into 5 growth stages. As the growth phases in the planting plan are all of fixed duration, the jump between the states of each growth phase is only determined by the staying time, and the jump is triggered by the timer. A new state needs to be defined here to indicate that the lettuce has not yet been planted, which is set by the worker.
EPA6 (Figure 9)—When the adjustment scene is the heading stage of Italian lettuce, the temperature during the day must be maintained at 20–22 °C, and the temperature at night must be maintained at 10–15 °C to achieve the best growth state of lettuce.

![Figure 9. EPA 6.](image)

EPA7 (Figure 10)—Take the goal of correcting temperature regulation only through temperature changes, when the greenhouse temperature is rising sharply or the temperature is too high, the temperature will be lowered, and when entering the state will be issued the need to lower the temperature of the event, and vice versa, increase the temperature to maintain the greenhouse temperature stability.

![Figure 10. EPA 7.](image)

EPA8 (Figure 11)—Taking the connection status of the exhaust fan provided to the worker as an example: the exhaust fan has two connection states: connected and disconnected, and the worker is notified of the current connection state of the exhaust fan when the status changes.

![Figure 11. EPA 8.](image)
EPA9 (Figure 12)—Take the control method with shading, down ventilation and water valve control, and the need to cool down through the down ventilation as an example: there are three states: non-operational, equipment connection, and executable. The condition for normal operation is that the exhaust fan, down ventilation and water valve are controlled at the same time. It is in the connected state, and the lower ventilation is in the state of opening and cooling.

![Figure 12. EPA 9.](image)

EPA10 (Figure 13)—Taking the humidification method as an example: there are two states: humidification (N) and dehumidification (F). When the environment needs to be humidified, the method does not need to be executed. When the environment needs to be humidified, it determines whether to enter the predetermined threshold and generates the environment list event in each clock cycle.

![Figure 13. EPA 10.](image)

EPA11 (Figure 14) In the process of greenhouse regulation, there are two types of situations. One is the general situation where the greenhouse can be adjusted back to the target state through a series of controls without causing other factors to deviate from the established range. The other is the special situation where a certain environmental factor deviates too far from the target, cannot find a suitable method to adjust it back to the target state, and needs to restore the factor to the target state first. The evaluation functions of the above two cases are, respectively, denoted as $f_{\text{search}}$ and $f_{\text{extreme}}$. When the two functions are searched in the method set, they can return the number of the optimal control method and whether it is found.

$$f_{\text{search}} = x, \min \varphi, \max |l|$$

$s.t. \quad \forall \text{Cl}_t(x, \text{now}, \text{RUN}), \exists (x, \text{now}, \varphi, l)$

$$f_{\text{extreme}} = x$$

$s.t. \quad x[\text{PI}.\text{param}] = [\text{Any}]$

$x[\text{PI}.\text{value}] = \pm \inf$

Est([\text{Any}], \text{EL}|\text{EH}) \quad \text{from Jump}_{\text{yn}}$
EPA12 (Figure 15) In the case of the boundary method, the length of the queue for receiving events is 1. For general methods, you do not have to generate an environment alert event at the end.

EPA13 (Figure 16) In the case of fan exhausts, the length of the receive event queue is 1. The control information generated by receiving controller state information and method generates formal control information, whether the control in the feedback method is effective, and whether the feedback to the worker controller is wrong so that the greenhouse can cope with complex situations.
5. Experiments

Based on the modeling method designed by Peng et al. [76] and collected 10 kinds of real control data, a model of temperature and humidity changes in Zhuozhou was established to simulate the changes after greenhouse regulation. Inputting the collected real data and the control instructions output by the CEP system into the above model, the entire process of temperature and humidity changes and operations under the control of the CEP system can be simulated.

After considering the above situation, the CEP system built in this experiment has realized six functions. The first is to study the changing trend of environmental factors from environmental information. Second, mark the difference between the state of the environment and setting goals. The third is to determine the actual adjustment range of each factor in the greenhouse. Fourth, collect the required environmental information and estimate whether the method is suitable for the current environmental conditions. Fifthly, score each control method and select the one with the highest current score. The sixth is to execute concrete control methods and issue instructions. Based on the above procedures, we built a CEP engine for simulation experiments of short-term greenhouse control.

In order to verify the performance of our CEP system in the real greenhouse scenario, we collected the temperature and humidity as supporting data of the solar greenhouse of Zhuozhou Farm of China Agricultural University from January 15 to 17, 2021, with the collection frequency of approximately 200 s/time. We use it as the bias value in response to changes in temperature and in the experiment to simulate fluctuation of temperature in real scenarios and compare the early warning capabilities of the three types of methods. A total of 1069 pieces of valid data were collected in this experiment, with a duration of about two and a half days. The curve of temperature and humidity are represented in Figure 17. They are used as the temperature and humidity fluctuations in the real greenhouse environment to be superimposed on the experimental model for subsequent control experiments.

![Figure 17. Greenhouse temperature and humidity data simulating the general greenhouse scene.](image)

Substituting the above data, Figure 18 can be obtained. Figure 18 is divided into three parts from top to bottom: temperature, humidity curve, and corresponding controller status. The dark color part in the background of the curve denotes the optimal growth interval of the crop, while the light color denotes the normal growth interval of the crop.
The abscissa is the number of cycles, 200 s per cycle, which is the same as the frequency of the sensor during acquisition, while the ordinate is the temperature and humidity value. In the lower controller state diagram, the states of the upper vent, lower vent, water pump, and sun visor are drawn from top to bottom. The color corresponding to each period on the right signifies the current state of the controller, including the opening and closing of the upper and lower vents, the opening and closing of the water pump, and the shading area of the sun visor. The first three controllers can be switched on and off. The sun visor can be switched in three states: 0%, 50%, and 100%.

Figure 18. Simulate the results of the experimental CEP system operation.

For example, the humidity was low at first, the temperature was normal, and the greenhouse was being ventilated and watered. After a period, the CEP system found that the humidity had returned to normal through the collected information, and immediately issued an instruction to stop watering. However, at that time, affected by the external humidity, the humidity in the greenhouse was still rising, the CEP system immediately shut down the lower ventilation, and the humidity and temperature dropped so that the temperature and humidity could be controlled within a reasonable range.

Similarly, around the 400th time, the temperature was high, and the humidity was normal. In such conditions, the CEP system issued a command to adjust the shading area from 0% to 50% and then opened the lower ventilation, which greatly reduced the temperature. However, the opening of the lower ventilation also caused a significant drop in humidity, which was lower than the predetermined target value. At this time, the system immediately adjusted the shading area from 50% to 0%, which increased the temperature, accelerated the transpiration of plants, and brought the humidity to normal.

Table 7 is the proportion of data points where the temperature and humidity are in the most normal and optimal growth range of the crop during regulation. It can be clearly found that the proportion of temperature and humidity data points in the most suitable and normal range after being regulated by the CEP system has increased significantly compared with that before being regulated. This illustrates regulation effect of our system is remarkable.
Table 7. The proportion of each interval in the experimental data.

| Classify          | Factor       | Proportion of Optimal Growth Data Points | Proportion of Normal Growth Data Points |
|-------------------|--------------|-----------------------------------------|----------------------------------------|
| Raw data          | Temperature  | 0.10383                                 | 0.18521                                |
|                   | Humidity     | 0.50888                                 | 0.63985                                |
| Results of experiment | Temperature | 0.87558                                 | 0.98409                                |
|                   | Humidity     | 0.65575                                 | 0.91300                                |

According to the values in the experiment, when our greenhouse CEP system is applied to the absolute real environment, the temperature and humidity can be controlled within the predetermined range by controlling ventilation, watering, and shading. The experimental results demonstrate that the architecture of the greenhouse complex event processing system designed in this paper is accessible and effective.

6. Conclusions

This paper designs a greenhouse automatic control method and system based on complex event processing and provides a general greenhouse-oriented CEP system implementation plan for agricultural experts and related workers. This structure of our system has the advantages of high efficiency, convenience, and low coupling, which can solve the problem of identification and integration of complex patterns in the greenhouse. This paper categorizes the information appearing in the greenhouse, defines 21 types of event types and structures, 13 types of EPA, and specific construction methods of EPA, and obtains a CEP method for greenhouse control. Specifically, the information obtained is used to determine whether an atomic event has occurred. Then, the atomic events are input into the CEP engine to form complex events and control information. Finally, the greenhouse controller in the greenhouse is regulated by this information. The CEP engine designed according to the theory of this article effectively solves the problems of a large amount of greenhouse data, strong coupling, and low cohesion. It is suitable for most greenhouse control scenarios, and also brings convenience to agricultural experts and related workers when using and expressing complex scenarios.

7. Patents

The results of this article have been sorted and published as a patent and authorized by the State Intellectual Property Office in China. The patent number is ZL202010506335.6.

Author Contributions: Conceptualization, Y.J. and X.L.; methodology, Y.J. and X.L.; software, Y.J.; validation, Y.J. and X.L.; formal analysis, Y.J. and X.L.; investigation, Y.J. and X.L.; resources, X.L.; data curation, Y.J.; writing—original draft preparation, Y.J.; writing—review and editing, Y.J. and X.L.; visualization, Y.J.; supervision, X.L.; project administration, Y.J. Both authors have read and agreed to the published version of the manuscript.

Funding: The study was partially supported by Natural Science Foundation of China through Grants (61601471).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulated experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.
Abbreviations

The following abbreviations are used in this manuscript:

- CEP: Complex event processing
- EPA: Event processing agent
- EPN: Event processing network
- BNF: Backus Naur Form
- BES: Basic event structure

References

1. Marcu, I.; Suciu, G.; Balaceanu, C.M.; Vulpe, A.; Drăgulinescu, A.-M. Arrowhead technology for digitalization and automation solution: Smart cities and smart agriculture. *Sensors* **2020**, *20*, 1464. [CrossRef]

2. De Oliveira, R.L.M.; Santos, I.V.; Graciano, G.F.; Libânio, A.A.C.; de Oliveira, L.K.; Bracarense, L.D. A sustainable approach for urban farming based on city logistics concepts for local production and consumption of vegetables. *Res. Transp. Econ.* **2021**, *87*, 101038. [CrossRef]

3. Lujuan, D.; Qiaoling, F.; Shujun, L.; Yuelan, N.; Baozhen, L. Research status and development direction of intelligent greenhouse environment control. *J. Zhengzhou Univ. Light Ind.* **2003**, *4*, 20–23.

4. Wang, Z.; Gai, S.; Cui, Y.; Dong, J. Research on intelligent agriculture system based on digital twins and blockchain. *J. Hebei Acad. Sci.* **2021**, *38*, 66–73.

5. Juan, N.; Ruizhi, S.; Xuefeng, D.; Hao, Y. Processing of uncertain and complex events in precision agriculture based on data lineage management. *Trans. Chin. Soc. Agric. Mach.* **2016**, *47*, 245–253.

6. Li, X.; Gao, H.; Qiao, Y.; Wang, H. Real time complex event processing engine for Internet of things. *Small Microcomput. Syst.* **2015**, *36*, 2047–2053.

7. Huang, Y.; Zhu, X. Design of multi-dimensional agricultural ecological intelligent control system based on green manufacturing. *Mod. Bus. Trade Ind.* **2019**, *40*, 193–194.

8. Ji, P.; Fu, Y.; Yang, B. Research on multivariable decoupling control of greenhouse system based on single neuron PID. *Chin. J. Agric. Mach.* **2020**, *41*, 143–147.

9. Alipour, M.; Loghavi, M. Development and evaluation of a comprehensive greenhouse climate control system using artificial neural network. *Univers. J. Control. Autom.* **2013**, *1*, 10–14. [CrossRef]

10. Huansen, F.; Yuangui, L.; Xuelian, Z.; Hu, C. Application of intelligent expert system in vegetable greenhouse planting. *Chin. J. Agric. Mach. Chem.* **2014**, *35*, 240–244.

11. Wu, X. Research on Key Technologies of Greenhouse Cultivation Expert System. Master’s Thesis, Tianjin University of Technology, Tianjin, China, 2013.

12. Xiao, D.; Xiu, J. *Software Engineering Models and Methods*, 2nd ed.; Beijing University of Posts and Telecommunications Press: Beijing, China, 2014.

13. Hangdong, J.; Zhonghong, L. Analysis of a winter hail weather process. *Guangxi Meteorol.* **2005**, *51*, 134–136.

14. Li, X.; Wang, J.; Gao, H. Spatiotemporal event model for complex event processing of farmland Internet of Things. *Trans. Chin. Soc. Agric. Mach.* **2015**, *51*, 153–161.

15. Li, X.; Wei, X.; Chen, X.; Tang, X.; Xie, T.; Jia, L. Construction and specification of complex events from sensor network data in farmland. *Int. Agric. Eng.* **2017**, *26*, 269–282.

16. Luckham, D.; Wantao, L.; Songlin, H. *Introduction to Complex Event Handling*; Science Press: Beijing, China, 2015.

17. Wen, B.; Yang, J. Research and application of complex event processing technology. *Mech. Eng. Autom.* **2018**, *6*, 27–29.

18. Dayarathna, M.; Perera, S. Recent advancements in event processing. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef]

19. Wang, Y.; Zheng, L.; Fan, W. Manufacturing workshop data acquisition and fusion based on standard semantic model and complex event processing under Cloud Architecture. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 3103–3115.

20. Yin, D.; Gong, H. Smart home middleware system based on event matching. *Fujian Comput.* **2019**, *35*, 28–30.

21. Jing, X.; Zhang, J.; Li, J. Complex event processing technology and its application in logistics internet of things. *Comput. Appl.* **2013**, *33*, 2026–2030.

22. Hooper, A.W.; Davis, P.F. Control of greenhouse air temperature with an adaptive control algorithm. *Symp. Greenh. Clim. Control* **1985**, *174*, 407–412. [CrossRef]

23. Ali, I.A.; Abdalla, A.M. A microcomputer-based system for all-year-round temperature control in greenhouses in dry arid lands. *Comput. Electron. Agric.* **1993**, *8*, 195–210. [CrossRef]

24. Xu, D. Research on Optimal Control Algorithm for Double Time Scale Greenhouse Production; China Agricultural University: Beijing, China, 2019.

25. Kodali, R.; Jain, V.; Karagwal, S. IoT Based smart greenhouse. In *Proceedings of the 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Agra, India, 21–23 December 2016.

26. Linker, R.; Gutman, P.O.; Seginer, I. Robust controllers for simultaneous control of temperature and CO₂ concentration in greenhouses. *Control Eng. Pract.* **1997**, *7*, 851–862. [CrossRef]
27. Bennis, N.; Duplaix, J.; Enéa, G.; Haloua, M.; Youlal, H. Greenhouse climate modelling and robust control. Comput. Electron. Agric. 2008, 61, 96–107. [CrossRef]
28. Pasgianos, G.D.; Arvanitis, K.G.; Polycarpou, P.; Sigrimis, N. A nonlinear feedback technique for greenhouse environmental control. Comput. Electron. Agric. 2003, 40, 153–177. [CrossRef]
29. Wu, X.; Zhao, L. Design and MATLAB simulation of non-linear controller for temperature, room temperature and humidity. J. Shenyang Agric. Univ. 2011, 42, 121–124.
30. Cunha, J.B. Real-time adaptive control for greenhouse heating. In Proceedings of the Cooling and CO2 Enrichment Computers in Agriculture and Natural Resources, Orlando, FL, USA, 23–25 July 2006.
31. Rodríguez, F.; Guzmán, J.L.; Berenguel, M.; Arahal, M.R. Adaptive hierarchical control of greenhouse crop production. Int. J. Adapt. Control Signal Process. 2010, 22, 180–197. [CrossRef]
32. Ouammi, A.; Achour, Y.; Dagdougui, H.; Zejli, D. Optimal operation scheduling for a smart greenhouse integrated microgrid. Energy Sustain. Dev. 2020, 58, 129–137. [CrossRef]
33. Piñón, S.; Camacho, E.F.; Kuchen, B.; Peña, M. Constrained predictive control of a greenhouse. Comput. Electron. Agric. 2005, 49, 317–329. [CrossRef]
34. Ramírez, A.A. Improving efficiency of greenhouse heating systems using model predictive control. In Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, 3–5 July 2005.
35. Berenguel, M.; Rodríguez, F.; Guzman, J.L.; Lacasa, D.; Pérez-Parra, J. Greenhouse diurnal temperature control with natural ventilation based on empirical models. Acta Hortic. 2006, 79, 57–64. [CrossRef]
36. Deng, L.; Zhang, K.; Gong, Y.; Chen, C. Preliminary study on multi-level control system and optimized target value setting of greenhouse environment. Trans. Chin. Soc. Agric. Eng. 2005, 5, 119–122.
37. Xu, L.; Su, Y.; Liang, Y. Requirements and status quo of control-oriented greenhouse system microclimate environment model. Trans. Chin. Soc. Agric. Eng. 2013, 29, 1–15.
38. Zhou, R.; Zhang, W. Application and review of greenhouse microclimate environmental system model. Res. Agric. Mech. 2011, 33, 189–192.
39. Zadeh, L.A. Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected papers by Lotfi. A. Zadeh; Archive for Mathematical Logic: Berlin/Heidelberg, Germany, 1996.
40. Peng, X.; Liu, G. Intelligent water-saving irrigation system based on fuzzy control and wireless sensor network digital home (ICDH). In Proceedings of the 2012 Fourth International Conference, Washington, DC, USA, 17–19 August 2012; pp. 252–256.
41. Mei, Z.; Chunhong, L.; Yaoguang, W.; Yingyi, C. Wireless intelligent monitoring system for temperature, room temperature and humidity based on fuzzy control. Agric. Eng. 2013, 3, 47–50.
42. Wang, Y.; Chai, T. Research summary of adaptive fuzzy control theory. Control Eng. 2006, 3, 193–198.
43. Ayodele, T.O. Introduction to Machine Learning; MIT Press: Cambridge, MA, USA, 2004.
44. Bishop, C.M. Pattern Recognition and Machine Learning; Springer: New York, NY, USA, 2006.
45. Qu, Y.; Ning, D.; Lai, Z.; Cheng, Q.; Mu, L. Neural network PID control of greenhouse temperature control system. Trans. Chin. Soc. Agric. Eng. 2011, 27, 307–311.
46. Sun, Y.; Cai, Y. Research on greenhouse multi-feature data fusion method based on WDNN. Trans. Chin. Soc. Agric. Mach. 2019, 50, 273–280.
47. Zadoks, J.C. EPIPRE: A disease and pest management system for winter wheat developed in the Netherlands. Eppo Bull. 2010, 11, 365–369. [CrossRef]
48. Carey, M.J.; Livny, M.; Jauhari, R. The HiPAC project: Combining active databases and timing constraints. ACM SIGMOD Rec. 1988, 17, 51–70. [CrossRef]
49. Li, X. Research on real-time reasoning technology based on active rules. Beijing. Grad. Sch. Chin. Acad. Sci. 2011, 11, 25.
50. Gatziu, S.; Dittrich, K.R. SAMOS: An active object-oriented database system. IEEE Q. Bull. Data Eng. Spec. Issue Act. Database 1992, 15, 23–26.
51. Gatziu, S.; Dittrich, K.R. Events in an active object-oriented database system 1 rule definition language. Rules Database Syst. 1993, 8, 23–39.
52. Gyllstrom, D.; Wu, E.; Chae, H. SASE: Complex Event Processing Over Streams. In Proceedings of the Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 7–10 January 2007; pp. 408–411.
53. Magid, Y.; Adi, A.; Barnea, M.; Botzer, D.; Rabinovich, E. Application Generation Framework for Real-Time Complex Event Processing. In Proceedings of the 32nd Annual IEEE Int Computer Software and Applications Conference, Turku, Finland, 28 July–1 August 2008; Volume 1, pp. 1162–1167.
54. Xiao, F.; Zhan, C.; Lai, H.; Tao, L.; Qu, Z. New parallel processing strategies in complex event processing systems with data streams. Int. J. Distrib. Sens. Netw. 2017, 13. [CrossRef]
55. Paschke, A. A Semantic Design Pattern Language for Complex Event Processing. In Proceedings of the AAAI Spring Symposium—Intelligent Event Processing, Stanford, CA, USA, 23–25 March 2009; pp. 54–60.
56. Bülow, S.; Backmann, M.; Herzberg, N.; Hille, T.; Meyer, A.; Ulm, B.; Wong, T.Y.; Weske, M. Monitoring of business processes with complex event processing. In Proceedings of the International Conference on Business Process Management, Beijing, China, 26 August 2013; pp. 277–290.
57. Barquero, G.; Burgueño, L.; Troya, J.; Vallecillo, A. Extending Complex Event Processing to Graph-structured Information. In Proceedings of the 21th ACM/IEEE International Conference, Copenhagen, Denmark, 14–19 October 2018.

58. Peng, S.; Liu, H.; Guo, X.; He, J. State-based event detection optimization for complex event processing. Sens. Transducers 2014, 164, 242.

59. Yan, L.; Liao, H.; Su, H. Parallel complex event detection based on regular tree pattern matching. In Proceedings of the 2018 International Conference on Big Data Technologies, Hangzhou, China, 18–20 May 2018; pp. 34–39.

60. Macia, H.; Valero, V.; Díaz, G.; Ortiz, G. Complex event processing modeling by prioritized colored petri nets. IEEE Access 2016, 4, 7425–7439. [CrossRef]

61. Ding, K.; Zhang, X.; Deng, B. C-NMSP: A high performance network management service platform for complex event processing international conference on future generation. Commun. Netw. 2008, 1, 7–10.

62. Mazon-Olivo, B.; Hernández-Rojas, D.; Maza-Salinas, J.; Pen, A. Rules engine and complex event processor in the context of internet of things for precision agriculture. Comput. Electron. Agric. 2018, 154, 347–360. [CrossRef]

63. Li, P.; Wang, J. Research progress on intelligent management of greenhouse environmental information. J. Agric. Mach. 2014, 45, 236–243.

64. Deng, X.; Sun, R.; Nie, J.; Wang, W. Modeling of a greenhouse environment monitoring IoT system based on time automata. Trans. Chin. Soc. Agric. Mach. 2016, 7, 301–308.

65. Apache Kafka: A Distributed Streaming Platform. Available online: https://kafka.apache.org/documentation/ (accessed on 15 June 2021).

66. Sun, K. Research on Acquisition and Classification Methods of Time Series Flow Events; Liaoning University: Liaoning, China, 2020.

67. Li, Y.; Xu, L. Design of a description language for LALR(1) parser based on BNF paradigm. New Technol. New Process 2015, 6, 70–72.

68. Zhang, L.; Duan, Q.; Li, D. Research on the construction technology of the diagnosis and treatment of corn diseases and insect pests. Agric. Mech. Res. 2012, 34, 41–45.

69. Xiuli, H.; Zhi, C. Design and implementation of heterogeneous web platform based on JSON. Comput. Technol. Dev. 2021, 31, 120–125.

70. Liu, H.; Xiong, W.; Li, Y.; Yang, D. Research progress on the impact of climate change on China’s crop rotation system. Chin. J. Agric. Meteorol. 2017, 38, 613–631.

71. Shi, Z. Italian Lettuce. Yangtze River Veg. 1996, 2, 23.

72. Sun, H. Lettuce, Oleracea, Lettuce; Guangdong Science Technology Press: Guangzhou, China, 2002.

73. Yu, Y.; Liu, W. The effect of light quality and photoperiod on the growth and quality of hydroponic lettuce under low light conditions. Chin. J. Agric. Meteorol. 2015, 36, 739–745.

74. Zhou, Q. The Influence of Facility Weak Light on Lettuce Growth and Nitrate Accumulation; Jiangsu University: Jiangsu, China, 2008.

75. Feng, Y.; Chen, H.; He, L.; Zhao, C.; Yu, T. The effect of light intensity factors on the growth and yield of sweet waxy corn. Anhui Agric. Sci. 2007, 17, 5109–5111.

76. Peng, X. Research and Development of Decoupling Control. System for Greenhouse Environment Variables Based on Neural Network; Xinjiang University: Xinjiang, China, 2018.