INTERFACE DEVICE FOR CONNECTING PROCESS CONTROLLERS TO OPC UA PEER DEVICES

FIELD OF THE INVENTION

The present invention relates to managing communications between process controllers and peer devices for the purpose of industrial process automation.

BACKGROUND

Automated industrial plans typically comprise many field devices that are in a direct relationship with an industrial production process. These field devices are controlled by one or more process controllers. While several different kinds and architectures of fieldbuses have been used in the past to connect the field devices to the process controllers, many “industry 4.0” concepts assume that communication between process controllers and field devise will converge on IP communication in combination with Open Platform Communications Unified Architecture, OPC UA. Client-server architectures for OPC UA communications are known from WO 2009/083 133 A1.

Much of the functionality of OPA UA field devices is provided by means of “methods” of the respective devices that are to be called by the process controller. For example, if a function of an actuator field device enables to control a valve, a call to a respective method “move” allows to position the piston of the valve. Field devices may implement many more actions (like “shutdown” or “reset”) that may be triggered by calling respective methods.

OBJECTIVE OF THE INVENTION

It is an objective of the present invention to facilitate the automation of a plant with multiple field devices, and/or increase the utilization of the functionality provided by each field device, particularly in a setting where the process controller is not aware of the complete set of OPC UA methods that is implemented on the side of the field device.

This objective is achieved by an interface device with a process controller interface, a peer device interface and a logic unit.

DISCLOSURE OF THE INVENTION

The invention provides an interface device for connecting at least one process controller to one or more peer devices.

In particular, such peer devices may be field devices. Such field devices may, in particular, be devices that are in direct interaction with an industrial process. For example, the field devices may comprise measurement instrument that measures a physical quantity in the industrial process, such as a temperature, a pressure, a voltage, an amperage or a concentration of at least one substance.

The peer devices may also be other process controllers, such as programmable logic controllers, PLC. For example, a process control system may be segmented into relatively autonomous PLC-based subsystems. A central process controller may then, for example, communicate with these subsystems to orchestrate the process as a whole.

Generally speaking, the role of the process controller may, for example, be to orchestrate the operation of one or more peer devices of whatever kind towards a particular goal. For example, a low-level controller may have the job of actuating one or more field devices such that a certain quantity, such as a temperature or a pressure measured by another field device, is maintained at a given set-point. A higher-level controller may in turn control those given set-points such that another goal for the industrial process as a whole is met.

The interface device comprises a process controller interface for bidirectional communication with the process controller. The interface device further comprises a peer device interface for bidirectional OPC UA communication with the peer devices. In particular, the peer device interface and the process controller interface may be physically distinct interfaces, and/or they may operate according to different communication standards. But this is not required. Rather, both interfaces may operate according to the same communication standard and even within the same network. For example, if both the process controller and the peer device speak OPC UA and are connected to the same Ethernet network, the process controller interface and the peer device interface may be distinct virtual network interfaces on a same underlying physical Ethernet card.

The interface device comprises a logic unit. This logic unit implements a kind of “if-this-then-that” (IFTTT) logic. In response to one or more inputs from the process controller, and/or from the one or more peer devices, meeting one or more predetermined conditions, the logic unit determines one or more to-be-performed actions from the one or more inputs according to a predetermined logic. For example, such action may comprise communicating data that has been received from the process controller, respectively from the peer device, to the peer device, respectively to the process controller. The action may also comprise to cause the peer device to alter the course of the industrial process somehow.

The logic unit determines, according to the predetermined logic, one or more outputs which, when supplied to the process controller, and/or to the peer devices, cause the to-be-performed actions to be performed. For example, if the communicating of data between the process controller and the peer device is to be mediated and/or brokered, some sort of translation of the data may be involved on a physical layer and/or on a logical layer. If the course of the physical process is to be altered, the logic unit may transform this desire into one or more concrete commands that need to be issued to the peer device, such as a field device.

The determined outputs are supplied to the process controller, and/or to the one or more peer devices. For example, if a method provided by an OPC UA peer device is to be invoked, the output may be a call instruction that causes invocation of this method

when supplied to an application programming interface, API, of the peer device. Even if the peer device is also a process controller, such as a PLC, there may be a situation where the two process controllers speak different languages and need the interface device for mutual communication.

The inventors have found that in this manner, the interface device introduces a layer of abstraction between the peer device and the process controller. For example, such abstraction facilitates communication in case either the process controller or the peer device is upgraded to a respective new device. For example, process controllers on the one hand and field devices as peer devices on the other hand typically have different lifecycles and physical lifetimes. Therefore, they are quite likely to fall due for replacement at different times. Because the devices frequently have a longer physical life than it corresponds to their innovation lifecycle, at a time where the device falls due for replacement, its manufacturer may already have switched to marketing a newer model, phasing out the old model. If the plant operator no longer has a spare of the old model in stock, he may be faced with a choice between finding a secondary source from where the old model may still be bought at a higher price, or buying a newer model and making adaptations to the device that he continues to use. For example, if a field device is replaced with a new model, the new model may have different capabilities, and a different syntax may be required to interact with it. If a process controller is replaced with a new model, the new process controller may expect certain modern features to be present in field devices, and may not be aware of legacy field devices that lack these modern features.

The interface device provides a place where the adaptations may be made, so that no changes need to be made to the process controller or to the field device that is still used even after the respective other device has been replaced. Depending on the age of the respective devices, it may be a lot easier to translate the output of one device for the use by the other device than it is to make the one device provide matching output for the other device in the first place.

In particular, one single interface device may translate between one or more process controllers and a plurality of peer devices. So there is only one place where something needs to be changed even if multiple process controllers and peer devices are involved.

In a particularly advantageous embodiment, the process controller interface comprises at least one I/O port that is mapped to a variable of the logic unit. Upon a write request to the I/O port by the process controller, the process controller interface reads an input written to the I/O port by the process controller, and updates the value of the variable according to this input. The logic unit may then use the value of the variable in any suitable manner when communicating with the peer device. For example, from the input written to the I/O port by the process controller, a call to at least one UPC UA method of a field device as a peer device may be determined as a to-be-performed action through the predetermined logic of the logic unit. For example, the logic unit may determine, from the I/O port to which the write request was made, a parameter of the field device that the process controller intends to set, and call the OPC UA method of the field device that is to be used to set this parameter, with the input written to the I/O port as an argument. In the same manner, parameters of a process controller as a peer device may be set. For example, a superordinate process controller may set a set-point in a lower-level process controller.

Likewise, upon a read request to the I/O port by the process controller, and/or upon an update to the variable by the logic unit, the logic unit supplies the value of the variable to the process controller by means of the I/O port. In particular, the logic unit may determine, by virtue of its predetermined logic, from the read request to the I/O port by the process controller, a call to at least one OPC UA method of a peer device that causes an update of the variable as a to-be-performed action. For example, the logic unit may determine, from the I/O port to which the read request was made, a parameter of the peer device that the process controller intends to read. The logic unit may then call the OPC UA method of the peer device that is to be used to read this parameter, store the returned parameter in the variable, and deliver this to the process controller via the I/O port.

In this manner, the interface devices enables even process controllers that have only I/O ports available for inputting and outputting of data and are thus purely data-driven, and not aware of OPC UA in any way, to communicate with OPC UA peer devices. There is no limitation as to which features of the peer device a process controller may use, as long as the interface provides some mapping between data exchanged through the I/O port and the respective to-be-performed actions on the OPC UA peer device.

For example, the method of the OPC UA device that is called may be encoded in the input written to the I/O port by the process controller, or in the value of the variable derived therefrom. Therefore, in a further particularly advantageous embodiment, in response to determining that said input or said variable meets or exceeds a predetermined threshold, the logic unit may determine a call to a first OPC UA method of a peer device as a to-be-performed action. In response to determining that said input, and/or said variable, does not meet said predetermined threshold, the logic unit may determine a call to a second OPC UA method of a peer device as a to-be-performed action.

From the perspective of the process controller, the interface device acts somewhat akin to a telephone computer that enables a customer who only has a touch-tone telephone to perform arbitrary actions on the available peer devices. Figuratively speaking, the process controller dials the interface device and is greeted with a menu like, "Welcome to the peer devices in this plant. To list available peer devices, press 1. To read data from a particular peer device, press 2. To change a parameter on a particular peer device, press 3…"

For example, this may be realized in the predetermined logic of the logic unit by means of an "if… then… else" control structure or a "case" control structure with respect to an input written to the I/O port.

In a further particularly advantageous embodiment, the logic unit is further configured to subscribe to at least one message topic published by the at least one peer device in a publish-subscribe model, and determine one or more to-be-performed actions in response to receiving a message of the subscribed message topic. For example, the logic unit may forward information from the received message to a process controller that has previously instructed the logic unit to subscribe to said message topic. The instruction to subscribe and the obtained published information may be conveyed back and forth between the process controller and the interface device in any suitable manner. Even pure data-driven communication via inputs written to I/O ports and outputs read back from I/O ports is sufficient.

In particular, one and the same published message topic of a peer device may be of interest to several process controllers. Published messages will then need to travel across the network to the interface device only once and can then be distributed onwards to the process controllers that are interested in them.

Likewise, if multiple process controllers are interested in the value of one and the same parameter of a peer device, the interface device needs to call a method of the peer device to obtain an updated value of this parameter only once. The updated value may then be stored in a variable of the logic unit and distributed further to any process controller that needs it. The interface device has control over the frequency with which an updated value of the parameter is obtained from the peer device, no matter how frequently the process controller requests this parameter. In this manner, the load on the OPC UA servers within the peer devices may be throttled. For example, field devices as peer devices are typically low-power devices with limited compute and networking resources. Therefore, their OPC UA servers may be equipped to handle a few simultaneous requests at most. The interface device may have much more powerful compute and networking resources, so it does not matter if this receives a lot more requests.

The to-be-performed actions that the logic unit determined in response to an input from the process controller, and/or from the peer devices, may, for example comprise:

- obtaining at least one measurement value or value of an operational parameter from at least one field device; and/or
- setting at least one operational parameter of at least one field device; and/or
- causing the field device to exert at least one physical influence on the performing of an industrial process in which the field device participates.

The actions are not limited to these or other primitives. Rather, in response to one single input or constellation of inputs, a whole sequence of actions may unfold by virtue of the predetermined logic of the logic unit. For example, in response to a process controller signaling that a vessel needs to be purged, the interface device may orchestrate a repeated filling and draining of the vessel by actuating various valves, and continue to do so until the concentration of a particular substance drops below a pre-set threshold value.

The functionality of the interface device is not limited to facilitating what was previously done in the cooperation between the process controller and the one or more peer devices. Rather, new functionality may be added. In a further advantageous embodiment, the predetermined conditions evaluated by the logic unit may comprise, as a prerequisite for at least one to be-performed action, at least one interlock condition for a physical state variable in an industrial plant that comprises the peer device. For example, if the process controller outputs the instruction to turn on an electron beam evaporator or a mass spectrometer on a vacuum chamber in which thin films are manufactured, the interface device may read out an OPC UA pressure gauge field device on the chamber in order to check whether the vacuum in the chamber is sufficient. If the electron beam evaporator or the mass spectrometer is operated without a sufficiently good vacuum, it may suffer damage.

Likewise, the one or more outputs that the predetermined logic determines may also be based on information obtained from multiple peer devices. For example, the predetermined logic may aggregate measurement values from multiple temperature sensors to form an average temperature that is then communicated to the process controller.

In a further advantageous embodiment, the logic unit is further configured to map OPC UA methods provided by multiple peer devices to handles within one common OPC UA namespace. In response to an input from the process controller that is indicative of an intention to call such a handle, it may then determine a call to the corresponding OPC UA method of a peer device as a to-be-performed action. In this manner, the process controller is relieved from having to know in which namespace the method resides. In particular, it is often not important which of several available peer devices perform a specific action; rather, what matters is that the action is performed. For example, if three parallel pumps are available for draining a vessel, but only the capacity of one pump is needed, the logic unit may choose in a “round-robin” manner which pump to use, so as to distribute the wear and tear more equally across the pumps.

In a further particularly advantageous embodiment, the logic unit is configured to accept interpretable, compilable or executable code, such as Python code, and use this as predetermined logic for determining to-be-performed actions, and/or for determining outputs for performing these actions. This provides a further flexibility for the user to

adapt the interface device to a particular plant. For example, the logic may be adapted to interface between a particular legacy process controller that only supports data-driven communications over a plurality of I/O ports and a constellation of modern OPC UA peer devices.

The interface device may be implemented on one or more computers with appropriate hardware interfaces for communicating with the process controller and with the peer devices. For example, most computers already come with an Ethernet interface that may be used to communicate with OPC UA peer devices. For communicating with process controllers that use data-driven I/O, an appropriate hardware interface may be connected to the computer as an interface card or plugged in to the computer via a Thunderbolt or USB connection. The main functionality of the interface device may then be embodied in software. Therefore, the invention also relates to a computer program with machine-readable instructions that, when executed by one or more computers, upgrade the one or more computers to the interface device described above.

The computer program may, for example, be marketed on a non-transitory computer-readable storage medium or in the form of a download product that, e.g., allows instant fulfillment after a purchase in an online store. Therefore, the invention also relates to a non-transitory computer-readable storage medium or a download product with the computer program. The invention also relates to one or more computers with the computer program, and/or with the non-transitory computer storage medium and/or download product.

DESCRIPTION OF THE FIGURES

In the following, the invention is illustrated using Figures without any intention to limit the scope of the invention. The Figures show:

Figure 1: Embodiment of the interface device 2 in communication with a process controller 1 and multiple peer devices 3a-3d;

Figure 2: Exemplary mapping of methods 3a1-3a3, 3b1-3b3 of multiple peer devices 3a-3d to handles 6a-6f in a common namespace 6.

Figure 1 shows an exemplary embodiment of the interface device 2 in a setting where it is in communication with a process controller 1 and multiple peer devices 3a-3d. For clarity, Figure 1 only shows one direction of communication: An input 4 is provided by the process controller 1, and this ultimately results in an output 5 being transmitted to one or more peer devices 3a-3d. But it is also possible that an input 4, like a message of a subscribed message topic, is provided by one or more peer devices 3a-3d and results in an output 5 being transmitted to the process controller 1.

The interface device 2 comprises a process controller interface 21 that is connectable to the process controller 1, a logic unit 22 and a peer device interface that is connectable to the peer devices 3a-3d, e.g., via an Ethernet network.

In the example shown in Figure 1, the process controller interface comprises an I/O port 21a that may be written to and read from by the process controller 1 in a data-driven manner. The I/O port 21a is mapped to a variable 22d of the logic unit 22. That is, an input 4 written to the I/O port 21a updates the variable 22d, and the current value of the variable 22d may be queried by reading from I/O port 21a.

The input 4 is checked against condition 22a in logic unit 22. If the condition is fulfilled (truth value 1), one or more to-be-performed actions 22c are determined from the input 4 based on predetermined logic 22b. The predetermined logic 22b also converts the to-be-performed actions 22c into an output 5 that is then sent to the appropriate peer device 3a-3d. In the example shown in Figure 1, the to-be-performed action 22c comprises invoking method 3a2 of peer device 3a1 that provides methods 3a1, 3a2 and 3a3. Consequently, the output 5 comprises a call instruction to the method 3a2.

Figure 2 illustrates the aggregation of multiple methods 3a1-3a3, 3b1-3b3 provided by several peer devices 3a-3d into one single OPC UA namespace 6 within logic unit 22. Each method 3a1-3a3, 3b1-3b3 is mapped to one handle 6a-6f in the unified namespace 6. One effect of this aggregation is that a process controller 1 does not need to know on which peer device 3a-3d the to-be-invoked method 3a1-3a3, 3b1-3b3 is implemented.

List of reference signs

1	process controller
2	interface device
21	process controller interface of interface device 2
21a	I/O port in process controller interface 21
22	logic unit of interface device 2
22a	predetermined condition for determining actions 22c
22b	predetermined logic for determining actions 22c, output 5
22c	to-be-performed actions in response to input 4
22d	variable in logic unit 22
23	peer device interface of interface device 2
3a-3d	peer devices
3a1-3a3	exemplary methods provided by peer device 3a
3b1-3b3	exemplary methods provided by peer device 3b
4	input to interface device 2
5	output from interface device 2
6	unified OPC UA namespace
6a-6f	handles in namespace 6a-6f for methods 3a1-3a3, 3b1-3b3

Claims:

1. An interface device (2) for connecting at least one process controller (1) to one or more peer devices (3a-3d), comprising:
   - a process controller interface (21) for bidirectional communication with the process controller (1);
   - a peer device interface (23) for bidirectional OPC UA communication with the peer devices (3a-3d); and
   - a logic unit (22) that is configured to: in response to one or more inputs (4) from the process controller (1), and/or from the one or more peer devices (3a-3d), meeting one or more predetermined conditions (22a),
     - determine one or more to-be-performed actions (22c) from the one or more inputs according to a predetermined logic (22b);
     - determine, according to the predetermined logic (22b), one or more outputs (5) which, when supplied to the process controller (1), and/or to the peer devices (3a-3d), cause the to-be-performed actions (22c) to be performed; and
     - supply these outputs (5) to the process controller (1), and/or to the one or more peer devices (3a-3d).
2. The interface device (2) of claim 1, wherein the process controller interface (21) comprises at least one I/O port (21a) that is mapped to a variable (22d) of the logic unit (22), and wherein the process controller (21) interface is configured to:
   - upon a write request to the I/O port (21a) by the process controller (1), read an input (4) written to the I/O port (21a) by the process controller (1), and update the value of the variable (22d) according to this input (4); and
   - upon a read request to the I/O port (21a) by the process controller (1), and/or upon an update to the variable (22d) by the logic unit (2), supply the value of the variable (22d) to the process controller (1) by means of the I/O port (21a).
3. The interface device (2) of claim 2, wherein the predetermined logic (22b) specifically comprises:
   - determining, from the input (4) written to the I/O port (21a) by the process controller (1), a call to at least one OPC UA method (3a1-3a3) of a peer device (3a-3d) as a to-be-performed action (22c); and/or

- determining, from the read request to the I/O port (21a) by the process controller (1), a call to at least one OPC UA method (3a1-3a3) of a peer device (3a-3d) that causes an update of the variable (22d) as a to-be-performed action (22c).

4. The interface device (2) of any one of claims 2 to 3, wherein the logic unit (22) is configured to:

- in response to determining that the variable (22d), and/or the input (4) written to the I/O port (21a) by the process controller (1), meets or exceeds a predetermined threshold, determine a call to a first OPC UA method (3a1-3a3) of a peer device (3a-3d) as a to-be-performed action (22c); and
- in response to determining that the variable (22d), and/or the input (4) written to the I/O port (21a) by the process controller (1), does not meet said predetermined threshold, determine a call to a second OPC UA method (3a1-3a3) of a peer device (3a-3d) as a to-be-performed action (22c).

5. The interface device (2) of any one of claims 1 to 4, wherein the logic unit (22) is further configured to: subscribe to at least one message topic published by the at least one peer device (3a-3d) in a publish-subscribe model, and determine one or more to-be-performed actions (22c) in response to receiving a message of the subscribed message topic.

6. The interface device (2) of claim 5, wherein the one or more to-be-performed actions (22c) comprise updating the value of a variable (22d) of the logic unit (22) based on the received message of the subscribed message topic.

7. The interface device (2) of any one of claims 1 to 6, wherein the to-be-performed actions (22c) comprise:

- obtaining at least one measurement value or value of an operational parameter from at least one field device (3a-3d); and/or
- setting at least one operational parameter of at least one field device (3a-3d); and/or
- causing the field device (3a-3d) to exert at least one physical influence on the performing of an industrial process in which the field device (3a-3d) participates.

8. The interface device (2) of any one of claims 1 to 7, wherein the one or more predetermined conditions (22a) comprise, as a prerequisite for at least one to-be-performed action (22c), at least one interlock

condition for a physical state variable in an industrial plant that comprises the peer device (3a-3d).

9. The interface device (2) of any one of claims 1 to 8, wherein the predetermined logic (22b) comprises determining the one or more outputs (5) based on information obtained from multiple peer devices (3a-3d).

10. The interface device (2) of any one of claims 1 to 9, wherein the logic unit (22) is further configured to: map OPC UA methods (3a1-3a3, 3b1-3b3) provided by multiple peer devices (3a-3d) to handles (6a-6f) within one common OPC UA namespace (6), and determine, in response to an input (4) from the process controller (1) that is indicative of an intention to call such a handle (6a-6f), a call to the corresponding OPC UA method (3a1-3a3, 3b1-3b3) of a peer device (3a-3d) as a to-be-performed action (22c).

11. The interface device (2) of any one of claims 1 to 10, wherein the logic unit (22) is configured to accept interpretable, compilable or executable code and use this as predetermined logic for determining to-be-performed actions (22c), and/or for determining outputs for performing these actions (22c).

12. A computer program, comprising machine-readable instructions that, when executed by one or more computers, upgrade the one or more computers to the interface device of any one of claims 1 to 11.

13. A non-transitory computer-readable storage medium or a download product with the computer program of claim 12.

14. One or more computers with the computer program of claim 12, and/or with the non-transitory computer-readable storage medium and/or download product of claim 13.