

Part VIII

ECHONET Service Middleware Specification

Release information (as of August 7th 2001)

| | | | |
|----------------|------------------------|---------------|-------------------------------|
| a) Version1.0 | March 18 th | 2000 released | Open to the consortium member |
| | July | 2000 | Open to the public |
| b) Version1.01 | May 23 rd | 2001 | Open to the public |
| c) Version2.00 | August 7 th | 2001 | Open to the consortium member |

Notes: On and after Version2.00, Powerline communication protocol has drawn together as Powerline communication A.

| |
|---|
| <p>Specifications published by the ECHONET CONSORTIUM are established without regard to industrial property rights(patents, utility models and so on). ECHONET CONSORTIUM has no responsibility for industrial property rights over contents of specifications.</p> |
|---|

Contents

| | | |
|------------------|--|------------|
| Chapter 1 | Overview | 1-1 |
| 1.1 | BASIC CONCEPT | 1-1 |
| 1.2 | POSITIONING ON COMMUNICATION LAYERS..... | 1-2 |
| 1.3 | ECHONET SERVICE MIDDLEWARE AND SERVICE OBJECT DEFINING METHOD | 1-3 |
| 1.4 | SERVICE API | 1-4 |
| Chapter 2 | ECHONET Basic Service Middleware..... | 2-1 |
| 2.1 | SYSTEM MODEL..... | 2-1 |
| 2.2 | DEFINITION OF BASIC SERVICE MIDDLEWARE FUNCTIONS..... | 2-1 |
| 2.3 | DEFINITION OF BASIC SERVICE OBJECT | 2-5 |
| 2.3.1 | Linked action service object | 2-5 |
| 2.4 | API OF BASIC SERVICE MIDDLEWARE | 2-7 |
| 2.4.1 | API of linked action service middleware | 2-7 |
| Chapter 3 | Housing-dedicated EMS Service Middleware (Case Proposition) | 3-1 |
| 3.1 | SYSTEM MODEL..... | 3-1 |
| 3.2 | HOUSING-DEDICATED EMS FUNCTIONS..... | 3-3 |
| 3.2.1 | Housing-dedicated feedback-type peak cut EMS | 3-3 |
| 3.2.2 | Housing-dedicated feed forward type peak-cut EMS..... | 3-5 |
| 3.2.3 | Housing-dedicated hybrid type peak-cut EMS..... | 3-7 |
| 3.3 | HOUSING-DEDICATED EMS SERVICE MIDDLEWARE FUNCTIONS | 3-9 |
| 3.3.1 | Basic concept..... | 3-9 |
| 3.3.2 | Detailed functions of housing-dedicated EMS service middleware..... | 3-10 |
| 3.4 | HOUSING-DEDICATED EMS SERVICE OBJECT | 3-11 |
| 3.4.1 | Basic concept..... | 3-11 |
| 3.4.2 | Detailed definitions of housing-dedicated EMS service classes | 3-11 |
| 3.5 | HOUSING-DEDICATED EMS SERVICE API | 3-12 |
| 3.5.1 | Basic concept..... | 3-12 |
| 3.5.2 | List of function items..... | 3-12 |
| Chapter 4 | Small/Medium-sized Building/Store-dedicated EMS Service Middleware (Case Proposition) 4-1 | |
| 4.1 | SYSTEM MODEL..... | 4-1 |
| 4.2 | SMALL/MEDIUM-SIZED BUILDING/STORE-DEDICATED EMS FUNCTIONS | 4-2 |
| 4.3 | SMALL/MEDIUM-SIZED BUILDING/STORE-DEDICATED EMS SERVICE OBJECT. 4-4 | |
| 4.3.1 | Medium/small building/store-dedicated EMS service class..... | 4-4 |
| 4.3.2 | Details of small/medium-sized building/store-dedicated service classes | 4-5 |



- 4.4 SEQUENCE 4-9**
- 4.5 SMALL/MEDIUM-SIZED BUILDING/STORE-DEDICATED EMS SERVICE API..... 4-10**
 - 4.5.1 Basic Concept..... 4-10**
 - 4.5.2 List of Function Items 4-10**

Chapter 1 Overview

1.1 Basic Concept

When a system grows complex and applications perform advanced processing, the development load of application software can be relieved by using software that can provide such processing, for example, in the form of shared libraries. Furthermore, where specific functions or applications are desired, there are many more expert but common types of processing. The ECHONET service middleware is software that provides API so that defined common processing functions can be accessed from application software. The service middleware also opens part of its functions to the public to allow various vendors to construct a system efficiently. This is called the ECHONET service object.

The service middleware may include software handling common functions regardless of applications, namely, simple device linked action services, scheduled operation services, gateway services for connections outside ECHONET, etc. In addition, software handling specific applications may also be included, such as energy management service (EMS) applications for efficient home energy use in houses, small and medium buildings, and stores; applications for automatic metering of power meters and gas meters; and applications for device maintenance. In this standard, software that handles common functions regardless of the application is defined as basic service middleware, while software dedicated to specific applications is defined as individual service middleware. Thus, common functions and application types will be gradually expanded and standardized. In particular, the gateway service is a special-use service that handles connections between ECHONET and external systems, and is therefore separately specified in Part 9.

This definition of service middleware is intended not only to relieve the load of developing application software through the use of service middleware but also to allow vendors to concentrate on system/device development based on essential functions and performance, thereby creating an environment that encourages vendors to develop useful products for users.

To make it possible to construct a system efficiently by accessing service middleware from the network, an object model is defined. Called the ECHONET service object, this model permits access from the network using an ECHONET protocol.

Also defined is an API to access service middleware from application software. This is called a service API.

Application software developers use the service middleware and service API (for accessing it) to facilitate the development of home system applications. They can also access service objects from the network to construct a system easily and efficiently.

1.2 Positioning on Communication Layers

The service middleware is positioned in the high-order layer of the ECHONET communications processing block in the communications layer configuration and provides common functions for implementing a certain application service for application software. Fig. 1.1 shows positioning of the service middleware and service object in the communications layer diagram. The service middleware is positioned within specific application software. Accordingly, service middleware uses the basic API to access ECHONET communication middleware functions in its internal processing. ECHONET communication middleware treats the service middleware as application software.

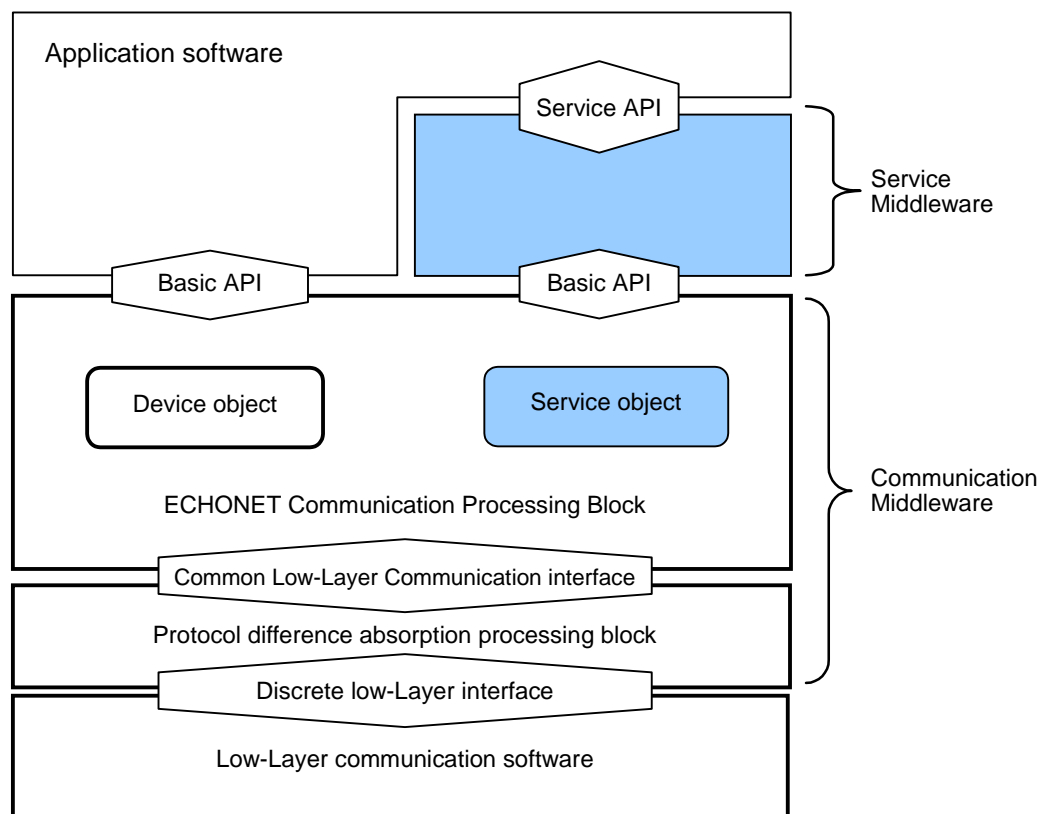


Fig. 1.1 Positioning of Service Middleware on Communication Layer Configuration

1.3 ECHONET Service Middleware and Service Object Defining Method

The concept of the service middleware and service object definitions is described below.

(1) Discussion regarding the system model

The scope of specific service application is defined, and the assumed system configuration is classified.

(2) Definition of service middleware functions

Standard and common functions are extracted and defined.

(3) Definition of service object

The network functions to be open to the public are provided as models on the basis of service middleware functions and defined as class specifications (function, service, property).

(4) Definition of sequence

The information exchange sequence between objects is defined.

(5) Definition of service API

API to access the service middleware are designed in accordance with the API design level.

The class group code of service objects is specified as 0x0D in Part 2, Chapter 4, Table 4.1. Table 1.1 shows a list of specified class codes in the service class group code 0x0D concerning the class codes of the service object that will be specified sequentially in detail in and after Chapter 2 of Part 8. 0x00 to 0xDF are assigned as basic and common service objects.

Table 1.1 List of Class Codes in Class Group Code (X1 = 0x0D service class)

| Class code | Object name | Remark |
|--------------|-------------------------|--------|
| 0x00 | Reserved for future use | |
| 0x01 | Linked action service | |
| 0x02 to 0xDE | Reserved for future use | |
| 0xDF | Gateway service | |
| 0xE0 to 0xFF | Reserved for future use | |

1.4 Service API

The service API is specified as an API for application software to access the service middleware described in Chapter 1. API function items are shown as functional outlines, and the two levels mentioned in the specified levels of the basic API are assumed for the service API as levels of the detailed specifications.

- (1) Level 1: Specification of function items and input/output data taking mounting into consideration
- (2) Level 2: Detailed interface specification intended for a specific language

Chapter 2 ECHONET Basic Service Middleware

2.1 System Model

The basic and common functions assumed for housing (including both single- and multiple-family dwellings) monitoring control functions, being the main focus of ECHONET application, are designed as service middleware. These basic functions may be used in common in specific applications and are defined as the basic service middleware. Accordingly, they are also applicable to small and medium-sized buildings and stores.

The target system model may connect devices mutually that are autonomously operated, or may have a controller to control centralized devices that are autonomously operated. That is, there are no special limitations on its form.

2.2 Definition of Basic Service Middleware Functions

The following two types of basic service middleware are defined:

- System construction service middleware
- Linked action service middleware

Their respective functions are described below.

(1) System construction service middleware

This service middleware provides a service to establish a system configuration. It has the following functions:

Instance management:

This function is such that when notice of a new EA or instance creation, or EA or instance deletion is received, the necessary table in the self is corrected according to this notice.

Installation place linked action setting:

This function creates an interconnected relation according to the information in each instance installation place property in the domain.

Domain management:

This function collects node profile information on every node in the domain and constantly updates and holds it as up-to-date property values.

(2) Linked action service middleware

Linked action means that when the operation status or measured value of one ECHONET device changes, the operation status of the other ECHONET device changes with it. That is, when the property value of a specific instance changes, the property value of the other specific instance also changes.

This information, associated between properties, is defined as linked action information. The linked action information is defined by the event triggering an interconnecting operation and the action corresponding to it.

The trigger event is a specific property behavior of a specific instance, a property value change of a device object, or a property value change of a service object. The action is a specific property behavior of a specific instance, a property value change of a device object, or a property value change of a service object.

The above interconnecting function takes one of the following three forms, depending on the information required for association:

Form

An interconnecting function is provided (with classes and properties mounted) and can perform linked action by specifying a concrete interconnecting destination instance. This is a tightly interconnected relationship that depends greatly on the application. The properties that can be connected as events are fixed. In this form, the device operation scenario is clear, and automatic setting is possible. In the case shown in Fig. 2.1, when an air conditioner is turned on, the fan is turned on in an interconnected form.

Form

An action to be entered in an interconnecting operation is specified, and if a trigger event is specified, an interconnecting operation can be performed. Alternatively, if an event is specified and an action is specified, an interconnecting operation can be performed. The former depends on an application and is fixed, and the action-executing side detects the occurrence of an optional event, thereby starting the operation. That is, an event is registered as linked action information on the action executing side. Fig. 2.2 shows a case in which an air conditioner has a function to change the operation status for a registered event as an external interconnecting function. In the latter, the event is fixed, and the event-managing side gives an instruction to execute an action to an optional action executing side set by the event managing side. That is, an action is registered as linked action information on the event managing side.

Form

When a trigger event is specified and an action is specified, an interconnecting operation can be performed. The generation of events is managed on the linked action information on events and actions that is optionally set, and an instruction to execute an action is given. The linked action information is registered in either an event generating source, an action executing source, or a third party.

Form is highly dependent on the application and can generate an interconnected relation by recognizing and controlling the opposite party of communication. Accordingly, a node profile object can be treated as Other-device control instance. Form , depending on the application, can handle information in fixed form according to each property; it is designed as middleware in Part 2 as a communication definition object of the communication middleware. The basic service middleware is not application-dependent and handles form (which also is not application-dependent) and performs function definition as linked action service middleware. Figs. 2.1 to 2.3 show examples of linked action forms to .

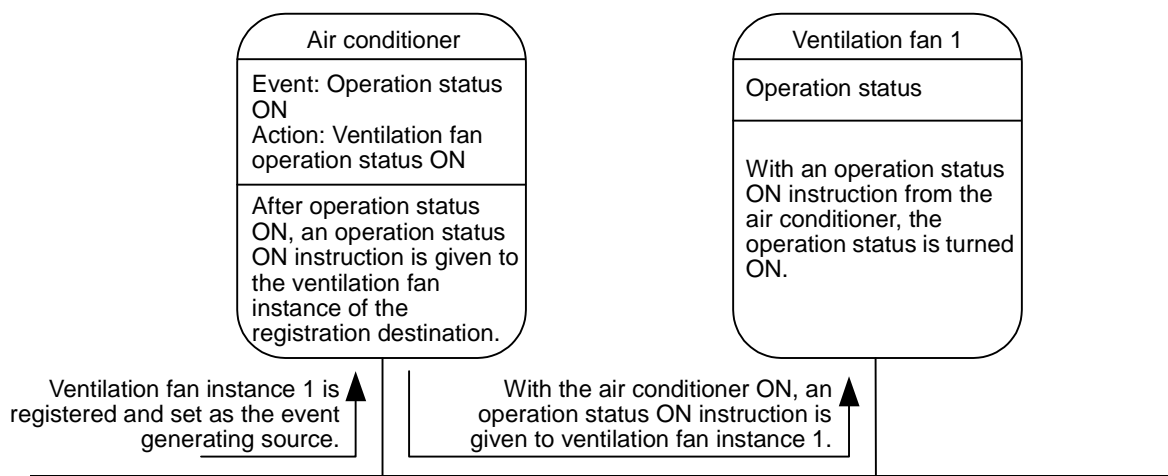


Fig. 2.1 Example of Linked action Form

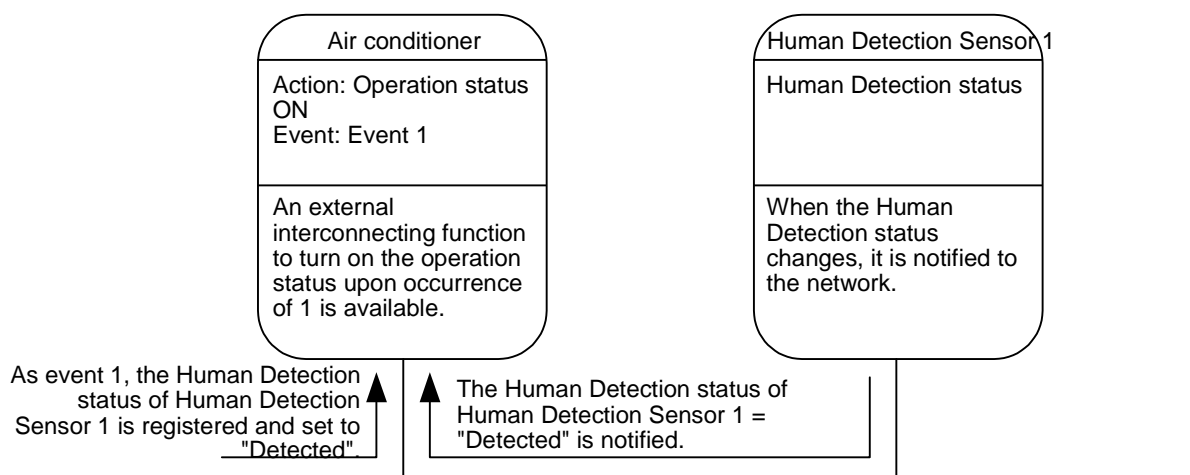


Fig. 2.2 Example of Linked action Form

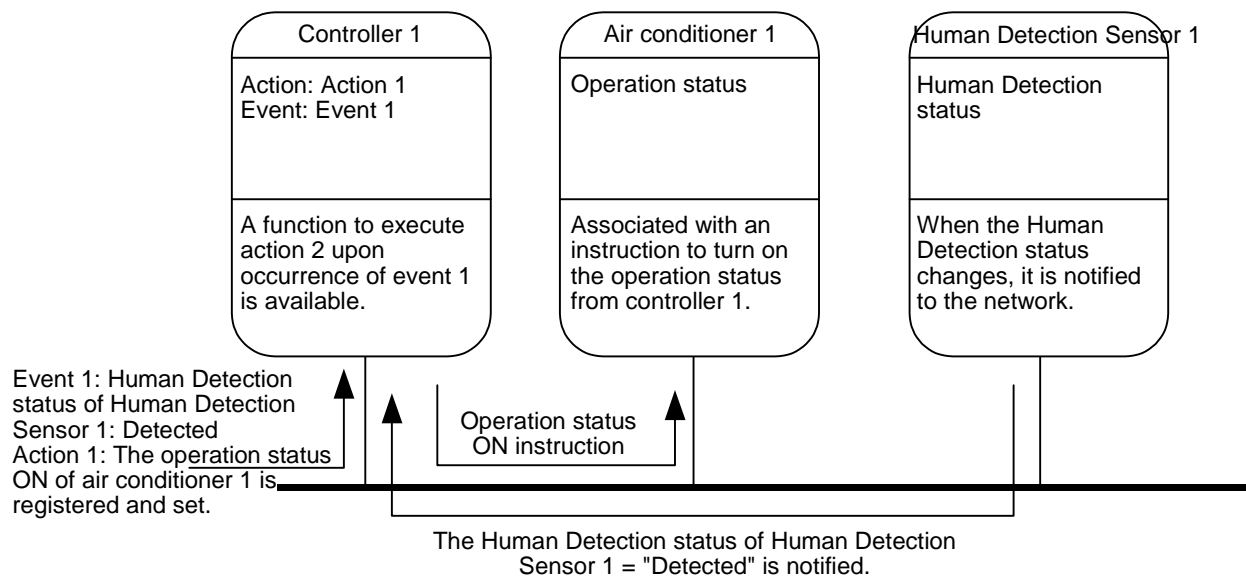


Fig. 2.3 Example of Linked action Form 3

2.3 Definition of Basic Service Object

The object corresponding to the linked action service middleware is defined as part of the basic service middleware.

2.3.1 Linked action service object

(1) Class No.

The linked action service middleware shall be provided with a class No. so that the linked action service middleware may accept a service of the ECHONET protocol from another ECHONET Node. The class No. assignment shall be as follows:

Class Name : Linked action service class
Class group code : 0x0D
Class code : 0x01

(2) Object property

The linked action service object supports the object properties shown in Table 2.2. However, these properties are used to access this linked action service object by the ECHONET protocol.

Table 2.2 Object Properties of Linked Action Service Class

| Property name | EPC | Contents of property | Data type | Data size | Unit | Access rule | Mandatory | Announcement at state change | Remark |
|---|------|--|--------------------|-----------|------|-------------|-----------|------------------------------|--------|
| | | Value range (decimal notation) | | | | | | | |
| Linked action information | 0xC0 | Indicates linked action information. | Array of Structure | — | — | Set | ○ | — | |
| | | {{ trigger EA } + { trigger EOJ } + { trigger EPC } + { trigger EDT } + { trigger EDT judgement condition }} + {{ action EA } + { action EOJ } + { action EPC } + { action ESV } + { action EDT }} | | | | Get | ○ | | |
| Maximum number of pieces of linked action information | 0xC1 | Maximum number of groups of linked action information | Unsigned Short | 2Byte | — | Get | ○ | — | |

- Explanation of linked action information

Arrangement of structure to store registered values related to linked action. Details are shown in Table 2.3.

- * Explanation of terms

Trigger : Event (as trigger) generating side

Action : Action (reacting to event) executing side

Table 2.3 Members Comprising a Linked Action Information Property

| Structure member name | Member name | Data type | Contents | Value |
|--------------------------------|------------------------|---------------|--|--|
| Trigger EA | Interlock_TrigEA | Array of BYTE | EA as trigger condition | |
| | NetID | BYTE | | |
| | NodeID | BYTE | | |
| Trigger EOJ | Interlock_TrigEOJ | Array of BYTE | EOJ as trigger condition | |
| | ClassGroup | BYTE | | |
| | Class | BYTE | | |
| | Instance | BYTE | | |
| Trigger EPC | Interlock_TrigEPC | BYTE | EPC as trigger condition | |
| Trigger EDT type | Interlock_TrigEDT_Type | BYTE | EDT type | 0x00:char 0x01:unsigned char 0x02:short 0x03:unsigned short 0x04:long 0x05:unsigend long 0x06:float(4byte) 0x07-0x7F:Reserved 0x80-0xFF:UserDefind |
| Trigger EDT | Interlock_TrigEDT | | EPC value as trigger condition | |
| Trigger EDT judgment condition | Interlock_Logic | BYTE | EDT judgment condition as trigger condition | 0x00:no check 0x01:= 0x02:">" 0x03:"<" 0x04:">=" 0x05:"<=" |
| | | | | 0x06-0x7F:Reserved 0x80-0xFF:UserDefind |
| Action EA | Interlock_ActEA | Array of BYTE | EA as action target | |
| | NetID | BYTE | | |
| | NodeID | BYTE | | |
| Action EOJ | Interlock_ActEOJ | Array of BYTE | EOJ as action target | |
| | ClassGroup | BYTE | | |
| | Class | BYTE | | |
| | Instance | BYTE | | |
| Action EPC | Interlock_ActEPC | BYTE | EPC as action target | |
| Action ESV | Interlock_ActESV | BYTE | Contents of operation for EPC as action target | |
| Action EDT type | Interlock_ActEDT_Type | BYTE | EDT type | The same as trigger condition |
| Action EDT | Interlock_ActEDT | | EPC value as action target | |

2.4 API of Basic Service Middleware

2.4.1 API of linked action service middleware

This section describes the API specification for accessing the linked action middleware.

(1) Outline of API functions

The API for the linked action service middleware are specified at the function item level.

(a) Linked action service middleware start/stop API

Accepts a request for start or stop of linked action service middleware.

(b) Linked action information registration/deletion API

Registers or deletes linked action information.

(c) Linked action information reference API

Accepts reference to linked action information.

(d) Maximum linked action information reference API

References maximum number of pieces of settable linked action information

(2) Level 1 API definition

For function level API, input/output data items and their specifications are described.

(a) Linked action service middleware start/stop API

| Direction | Data name | Contents and condition | Mounting specification |
|-----------|--------------|-------------------------------|------------------------|
| Input | – | | |
| Output | Return Value | TRUE: Success, FALSE: Failure | Optional |

(b) Linked action information registration/deletion API

| Direction | Data name | Contents and condition | Mounting specification |
|-----------|---------------------------|---|------------------------|
| Input | Linked action information | Contents shown in linked action information (Table 2.3) | Required |
| Output | Return Value | TRUE: Success, FALSE: Failure | Optional |

(c) Linked action information reference API

| Direction | Data name | Contents and condition | Mounting specification |
|-----------|---------------------------|---|------------------------|
| Input | Member specification | Specifies member of linked action information. | Optional |
| Output | Linked action information | Contents shown in Linked action Information (Table 2.3) | Required |

(d) Maximum linked action information reference API

| Direction | Data name | Contents and condition | Mounting specification |
|-----------|---|------------------------|------------------------|
| Input | — | | |
| Output | Maximum number of pieces of linked action information | Maximum value | Required |

Chapter 3 Housing-dedicated EMS Service Middleware (Case Proposition)

This Chapter provides examples of EMS Service Middleware, designed exclusively for home use, as a basis for future discussion concerning service objects.

3.1 System Model

As a system model for housing-dedicated EMS (Energy Management Service), a home-dedicated peak-cut EMS is designed as a control system to prevent total current consumption in the target range (usually within a single home) from exceeding a set value. The following methods might be adopted to implement such a system:

(A) Housing-dedicated feedback-type peak-cut EMS

When the total current consumption exceeds the set value, the capacity of the device in operation is reduced according to the rule specified by the controller.

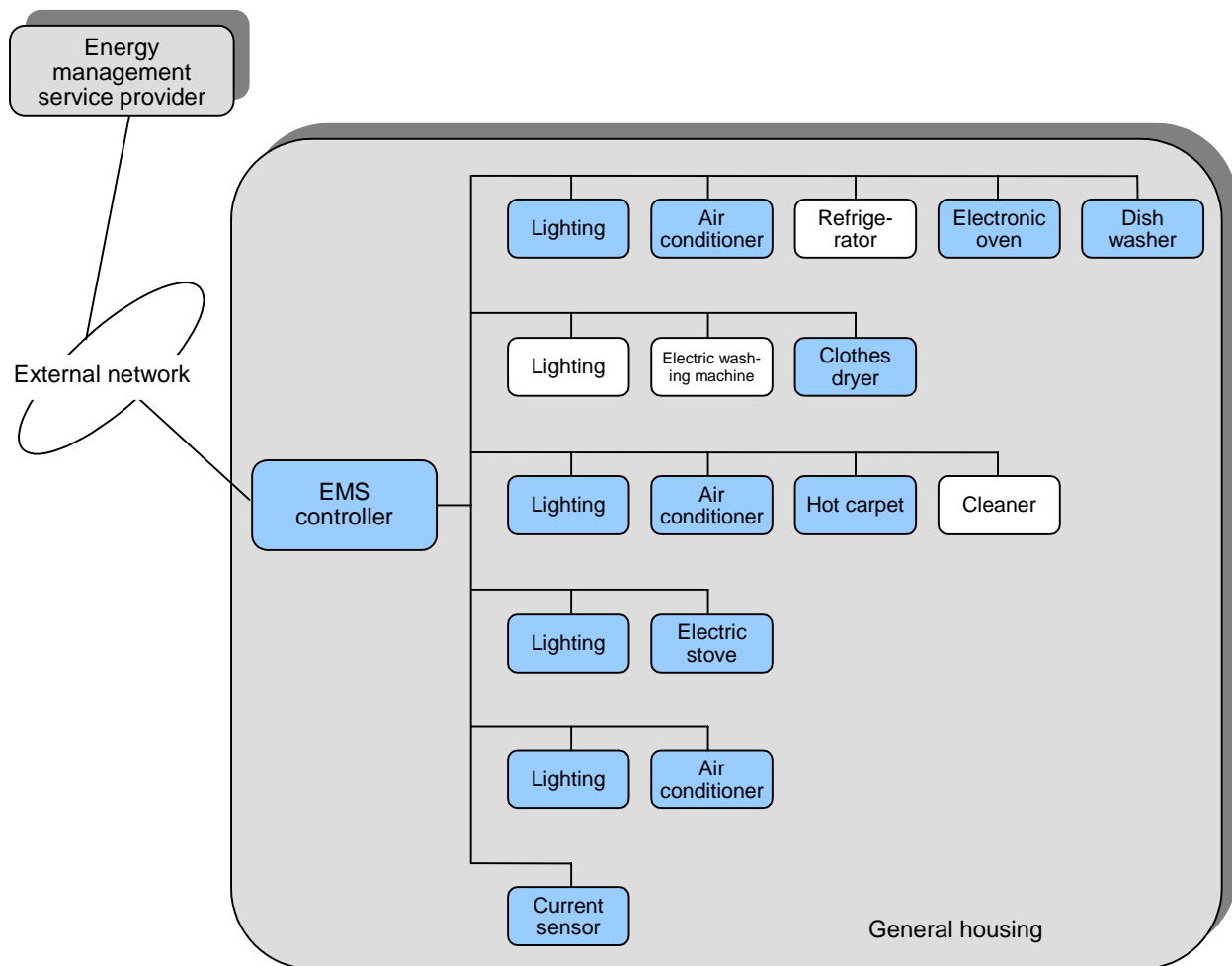
(B) Housing-dedicated feed forward type peak-cut EMS

When the device changes the capacity, this intention is applied to the controller, and the controller compares the current total consumption with the set value and assigns a consumable current value to the device.

(C) Housing-dedicated hybrid type peak-cut EMS

When the device changes the capacity, this intention is applied to the controller, and the controller compares the current total consumption with the set value and assigns the consumable current to the device. As a result, if the total current consumption exceeds the set value, the controller reduces the capacity of the device in operation according to a certain rule. (Hybrid of method A and method B)

Fig. 3.1 shows the network configuration on which the system is based. In the figure, the shaded devices are target devices for housing-dedicated EMS control.



3.2 Housing-dedicated EMS Functions

Regarding the three types of housing-dedicated EMS described in the previous section, this section outlines the control to be exerted by the EMS controller and describes its assumed functions.

3.2.1 Housing-dedicated feedback-type peak cut EMS

(1) Outline of controller control

Fig. 3.2 shows the outline of control to be exerted by the controller supervising housing-dedicated feedback-type peak-cut EMS control.

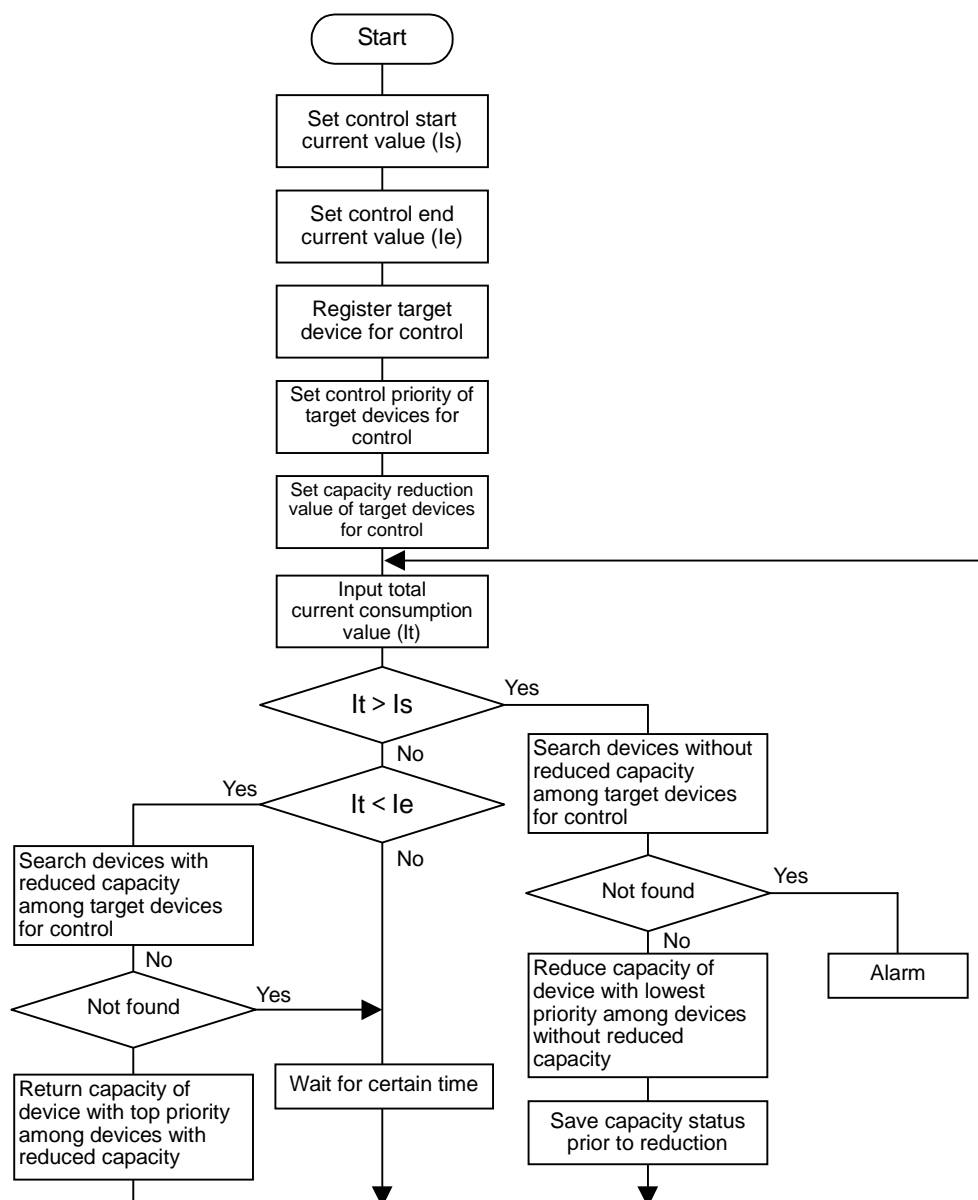


Fig. 3.2 Outline of Housing-dedicated Feedback Type Peak-cut EMS Control

(2) Controller functions

The controller functions required to implement housing-dedicated feedback-type peak-cut EMS control are described below.

(a) Control-target device managing function

A function to register the target ECHONET devices for EMS control and set and hold the parameters for individual devices. The parameters to be set are shown below.

- Control priority
- Capacity reduction value
- Recovery capacity value (capacity value immediately before capacity reduction)

The target capacity for control differs with each device type. Table 3.1 shows device types and examples of control-target capacity for these devices.

Table 3.1 Device Types and Control-target Capacity

| Device type | Control-target capacity |
|-----------------|-------------------------------|
| Lighting | Lighting, power ON/OFF |
| Air conditioner | Set temperature, power ON/OFF |
| Dish washer | Water temperature, suspension |
| Clothes dryer | Heater power, suspension |
| Hot carpet | Set temperature, power ON/OFF |
| Electric stove | Heater power, power ON/OFF |

(b) Current value monitoring function

A function to monitor the current value, which is a control start condition, and to set and hold the control start current value and control end current value as a reference for device control.

(c) Device control function

A function to exert peak-cut control according to the data of the control-target device managing function and the current value monitoring function.

3.2.2 Housing-dedicated feed forward type peak-cut EMS

(1) Outline of controller control

Fig. 3.3 shows the outline of control to be exerted by the controller supervising housing-dedicated feed forward type peak-cut EMS control.

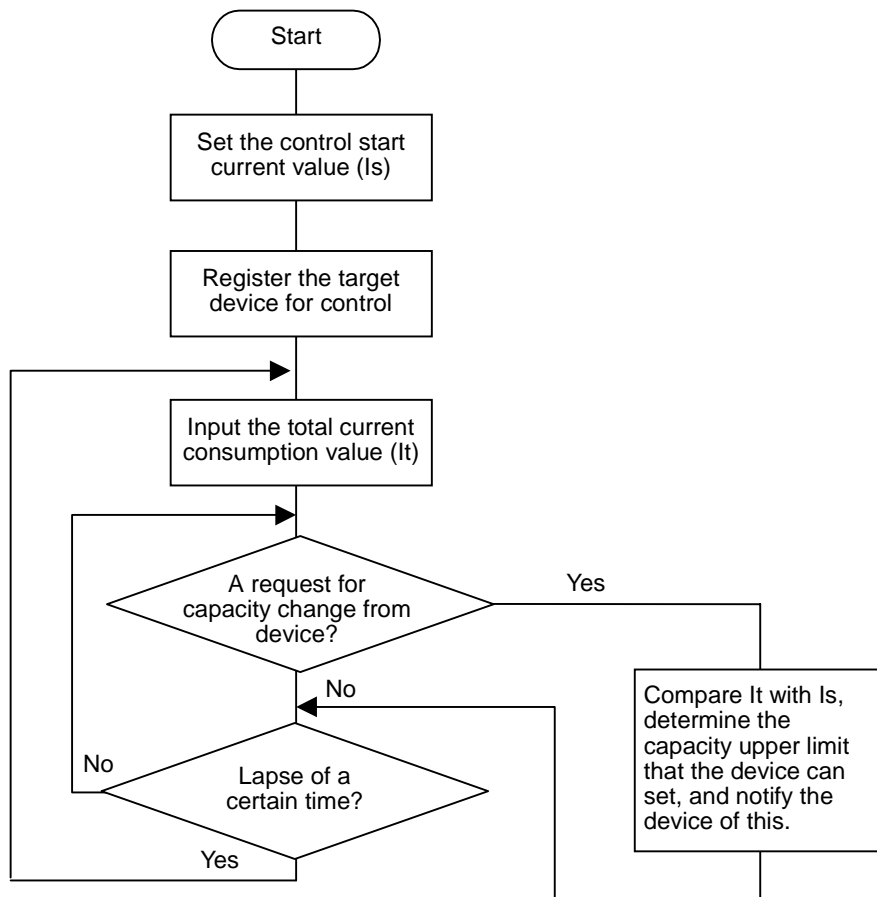


Fig. 3.3 Outline of Housing-dedicated Feed Forward Type Peak-cut EMS Control

(2) Controller functions

The functions of the controller required to implement housing-dedicated feed forward type peak-cut EMS control are described below.

(a) Control-target device managing function

A function to register the target ECHONET devices for EMS control

(b) Current value monitoring function

A function to monitor the current value, which is a control start condition, and to set and hold the control start current value and control end current value as a reference for device control.

(c) Device control function

A function to exert peak-cut control according to the data of the control-target device managing function and the current value monitoring function.

3.2.3 Housing-dedicated hybrid type peak-cut EMS

(1) Outline of controller control

Fig. 3.4 shows the outline of control to be exerted by the controller supervising housing-dedicated feedback-type peak-cut EMS control.

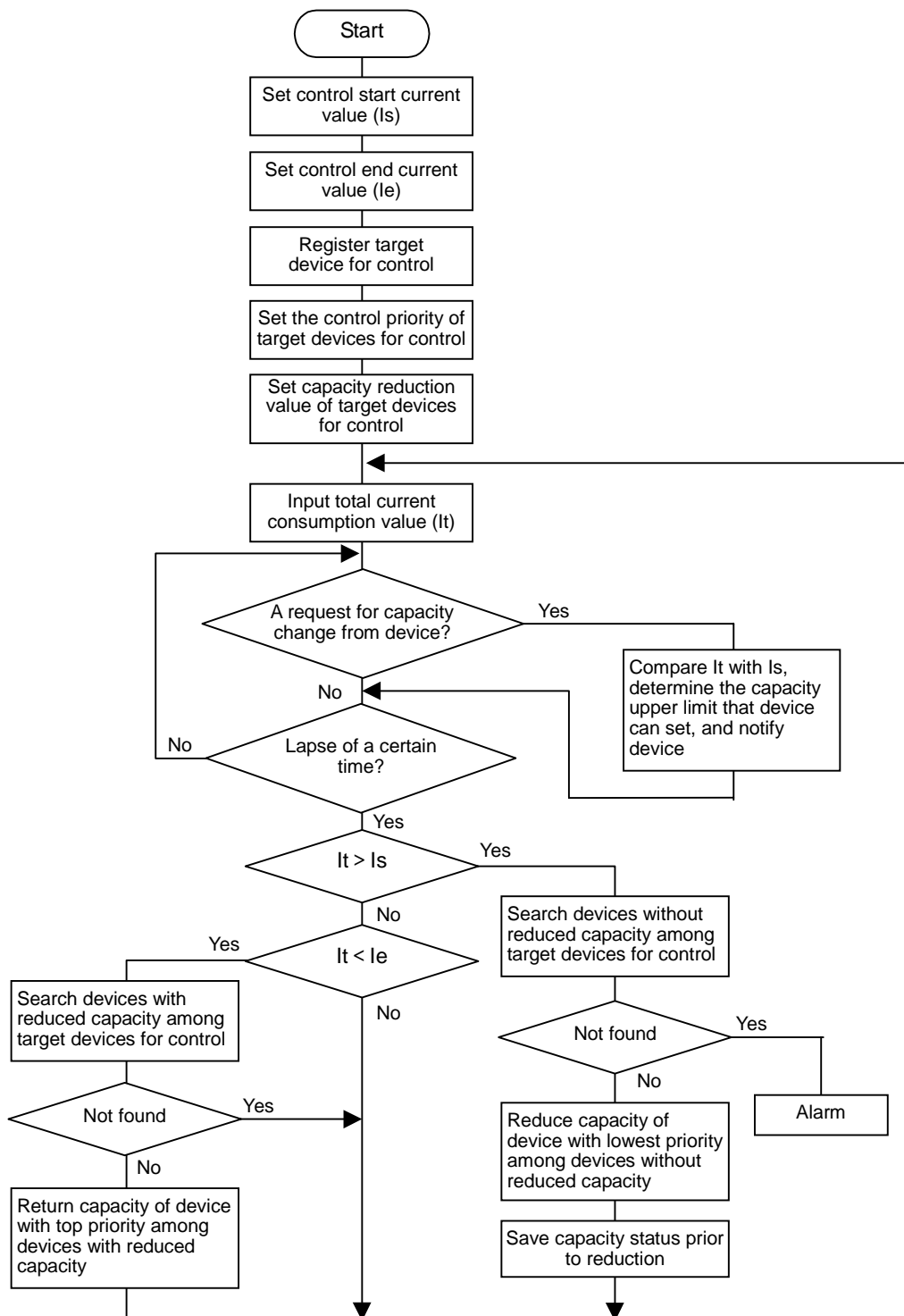


Fig. 3.4 Outline of Housing-dedicated Feed Forward Type Peak-cut EMS Control

(2) Controller functions

The controller functions required to implement the housing-dedicated hybrid type peak-cut EMS control are described below.

(a) Control-target device managing function

A function to register the target ECHONET devices for EMS control and to set and hold the parameters for individual devices. The parameters to be set are shown below.

- Control priority
- Capacity reduction value
- Recovery capacity value (capacity value immediately before capacity reduction)

The target capacity for control differs with each device type. For examples of device types and control-target capacity, see Table 3.1.

(b) Current value monitoring function

A function to monitor the current value, which is a control start condition, and to set and hold the control start current value and control end current value as a reference for device control.

(c) Device control function

A function to select devices to start or to terminate peak-cut control and give the corresponding instructions according to the data of the control-target device managing function and the current value monitoring function.

3.3 Housing-dedicated EMS Service Middleware Functions

3.3.1 Basic concept

The functions for housing-dedicated EMS shown as an example of housing-dedicated EMS in the previous section are divided into one part for general-purpose use and one special part for control. The former part is modeled and defined as single housing-dedicated EMS service middleware.

General-purpose part
Control-target device managing function
Current value monitoring function

Special part for control
Device control function

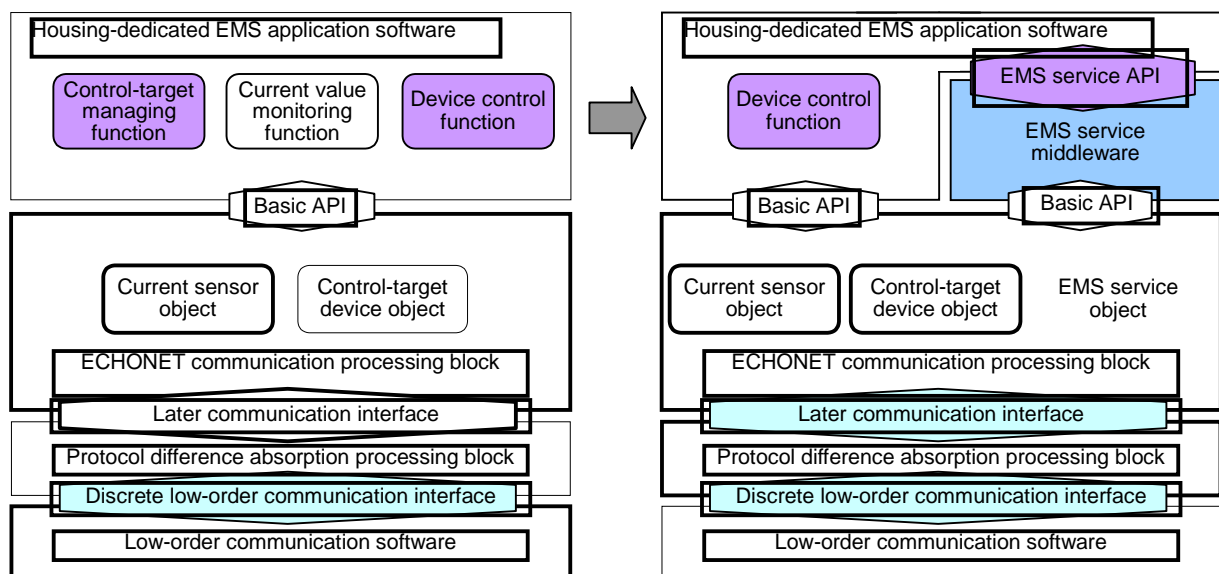


Fig. 3.5 Housing-dedicated EMS Service Middleware and Service Object

3.3.2 Detailed functions of housing-dedicated EMS service middleware

Table 3.2 shows the functions of the housing-dedicated EMS service middleware with a control-target device managing function and a current value monitoring function for housing-dedicated EMS.

Table 3.2 Housing-dedicated EMS Service Middleware Functions

| Function name | Contents of function |
|--|--|
| Addition of control-target device | Adds device specified by ECHONET address (EA) and ECHONET object code (EOJ) to EMS control-target device list as EMS control-target device, and creates unique EMS device ID (Apparatus ID) associated with set of EA and EOJ at 1:1. |
| Control parameter setting | Sets control priority, capacity reduction value, recovery capacity value, and upper limit capacity value for device specified by EMS device ID (ApparatusID). |
| Deletion of control-target device | Deletes device specified by EMS device ID (ApparatusID) from EMS control-target device list. |
| Current sensor setting | Holds both EA and EOJ of current sensor to measure current value as control reference. |
| Measurement time interval setting | Holds current measuring time interval of current sensor. |
| EMS condition setting | Sets and holds control start current value and control end current value as control reference. |
| Current value measurement | Gets and holds measured value of current sensor at each measurement time interval. |
| Acquisition of current value | Gets the up-to-date current value measured by the specified current sensor. |
| Search for EMS control-status device group | Searches EMS device IDs (ApparatusIDs) of all devices under EMS control in EMS control-target device list. |
| Search for EMS non-control-status device group | Searches EMS device IDs (ApparatusIDs) of all devices not under EMS control in EMS control-target device list. |
| Search for EMS control-status device | Searches EMS device ID (ApparatusID) of device with top priority among all devices under EMS control in EMS control-target device list. |
| Search for EMS non-control-status device | Searches EMS device ID (ApparatusID) of device with lowest priority among all devices not under EMS control in EMS control-target device list. |
| Event generation | Compares measured value of specified current sensor with control start current value and control end current value and generates an event. Measured value > Control start current value Generation of EV_ST Measured value < Control end current value Generation of EV_SP |
| Capacity reduction | Sets capacity of device specified by EMS device ID (ApparatusID) to capacity reduction value. The capacity prior to this change is stored as a capacity recovery value. |
| Capacity recovery | Sets capacity of device specified by EMS device ID (ApparatusID) to capacity recovery value. |

3.4 Housing-dedicated EMS Service Object

3.4.1 Basic concept

Housing-dedicated EMS service classes are defined to open the functions defined in the EMS service middleware (which are shown as an example in the previous section) to another device or the controller.

3.4.2 Detailed definitions of housing-dedicated EMS service classes

Table 3.3 provides definitions of the housing-dedicated EMS service class properties.

Table 3.3 Housing-dedicated EMS Service Middleware (1/2)

| Property name | EPC | Contents of property | Data type | Data size | Unit | Access rule | Mandatory | Announcement at state change | Remark |
|----------------------------------|------|---|---------------|---------------------|------|----------------------------|-----------|------------------------------|--------|
| | | Value range (decimal notation) | | | | | | | |
| Number of control-target devices | 0xC0 | Total number of target devices for EMS control | Unsigned char | 1 byte | – | Get | ○ | | |
| | | 0x00 ~ 0xFF (0 ~ 255) | | | | | | | |
| Control-target device list | 0xC1 | Arrangement of sets of (EA, EOJ). The element number becomes the EMS device ID (ApparatusID). | Array | 10 bytes × Max. 255 | – | Get GetM | ○ | | |
| Control priority list | 0xC2 | Control priority of EMS-target devices. The element number becomes the EMS device ID (ApparatusID). | Array | 1 byte × Max. 255 | – | Get GetM | ○ | | |
| | | 0x00 ~ 0xFF (0 ~ 255) | | | | | | | |
| Capacity reduction value list | 0xC3 | Capacity reduction value of EMS-target device. The element number becomes the EMS device ID (ApparatusID). | array | 2 bytes × Max. 255 | – | Get GetM | ○ | | |
| Recovery capacity value | 0xC4 | Recovery capacity value of EMS-target device. The element number becomes the EMS device ID (ApparatusID). | array | 2 bytes × Max. 255 | – | Get GetM | ○ | | |
| Upper limit capacity value list | 0xC5 | Upper limit capacity value of EMS-target device. The element number becomes the EMS device ID (ApparatusID). | array | 2 bytes × Max. 255 | – | Get GetM | ○ | | |
| EMS control status list | 0xC6 | List of flags to indicate whether the EMS control-target device is under control. The element number becomes the EMS device ID (ApparatusID). | array of char | 1 byte × Max. 255 | – | Get GetM Set SetM | ○ | | |
| | | 0x30: Under control 0x31: Not under control | | | | | | | |
| Current sensor | 0xC7 | EA and EOJ of current sensor as control reference | unsigned char | 6 bytes | – | Get/Set | ○ | | |
| Measuring time interval | 0xC8 | Time interval in getting measured value of current sensor | short | 2 bytes | Sec. | Get | ○ | | |
| | | 0x0000 ~ 0xFFFF | | | | | | | |

Table 3.3 Housing-dedicated EMS Service Middleware (2/2)

| Property name | EPC | Contents of property | Data type | Data size | Unit | Access rule | Mandatory | Announcement at state change | Remark |
|---------------------------------|------|---|---------------|-----------|------|-------------|-----------|------------------------------|--------|
| | | Value range (decimal notation) | | | | | | | |
| Control start/end current value | 0xC9 | Current value at start/end of EMS control | Short × 2 | 4 bytes | A | Get | ○ | | |
| | | 2 high-order bytes: Control start current value 2 low-order bytes: Control end current value | | | | | | | |
| Current value | 0xCA | Up-to-date current value input from current sensor | unsigned char | 1 byte | A | Get/Set | ○ | | |
| | | 0x00 ~ 0xFD (0 ~ 253) | | | | | | | |
| Event | 0xCB | Holds an event | unsigned char | 1 byte | — | Get | | | |
| | | Not generated: 0x30 EV_ST: 0x31 EV_SP: 0x32 | | | | | | | |

3.5 Housing-dedicated EMS Service API

This specification is under review in accordance with the housing-dedicated EMS service middleware specification of Sections 1 to 3 shown at the case proposition level. Regarding the API specification, only function items are presented as a case proposition.

3.5.1 Basic concept

This service API is designed around a basic concept specified by the service object API. API are defined as those that are executed by calling the previously specified service middleware function.

3.5.2 List of function items

Table 3.5 shows a list of function items of the service API to access the housing-dedicated EMS service middleware.

Table 3.4 List of Housing-dedicated EMS Service API Function Items

| API name | Contents of function |
|-------------------------------------|--|
| Control start/end | Requests service middleware to start or end. |
| Addition of control-target device | Inputs set of ECHONET address (EA) and ECHONET object code (EOJ) and adds it to EMS service object control-target device list. Returns EMS device ID (ApparatusID) as a return value. |
| Control parameter setting | Inputs EMS device ID (ApparatusID), control priority, capacity reduction value, recovery capacity value, and upper limit capacity value. If any device corresponds to EMS device ID (ApparatusID), input value is written to element specified by Apparatus ID of control priority list, capacity reduction value list, recovery capacity value list, and upper limit capacity list of EMS service object. |
| Deletion of control-target device | Inputs EMS device ID (ApparatusID). If target device is found in EMS service object control-target device list, it is deleted from list. |
| Search for device | Returns EMA device ID (ApparatusID) of control-target device satisfying specified conditions. Specification conditions: Device under EMS control, device not under ESM control, top control priority, lowest control priority (combinable) |
| Current sensor setting | Inputs both EA and EOJ of current sensor for measuring current value as control reference and writes them to current sensor property of EMS service object. |
| Measurement time interval setting | Inputs current measuring time interval of current sensor and holds it in current sensor object of EMS service object. |
| EMS condition setting | Inputs control start current value and control end current value as control reference and writes them to control start/end current value of EMS service object. |
| Current value measurement start/end | Asks service middleware to start/end current measurement. |
| Acquisition of current value | Reads out current property value of EMS service object. |
| Acquisition of event | Returns event property value of EMS service object. |
| Capacity reduction | Inputs EMS device ID (ApparatusID), sets capacity of target device in capacity reduction value, and holds capacity prior to change as capacity recovery value. |
| Capacity recovery | Inputs EMS device ID (ApparatusID) and sets capacity of device specified by EMS device ID (ApparatusID) in capacity recovery value. |

Chapter 4 Small/Medium-sized Building/Store-dedicated EMS Service Middleware (Case Proposition)

This Chapter describes examples of the small/medium-sized building/store-dedicated EMS service middleware and the service object as a basis for future discussion.

4.1 System Model

The functional requirements for the small/medium-sized building/store-dedicated power EMS system are analyzed. Based on the results, the small/medium-sized building/store-dedicated power EMS service middleware functions are defined and indicated as small/medium-sized building/store-dedicated EMS service objects. Fig. 4.1 shows an example of a store energy management system configuration.

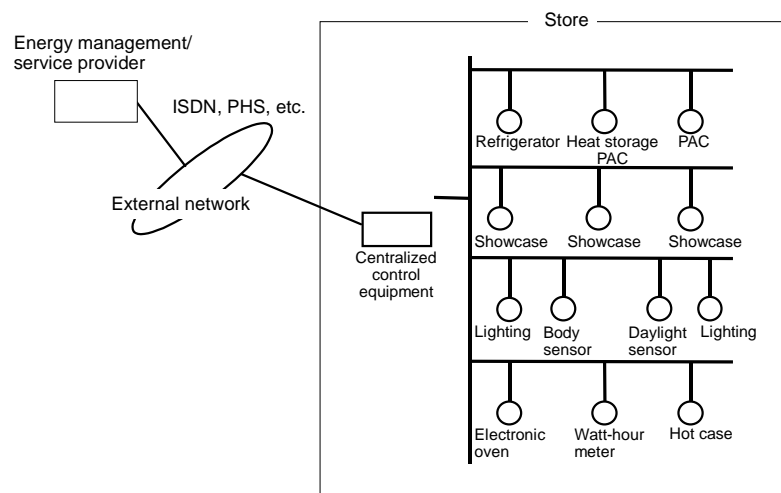


Fig. 4.1 Example of a Store EMS Configuration

Fig. 4.2 shows an example of a small/medium-sized building energy management system configuration.

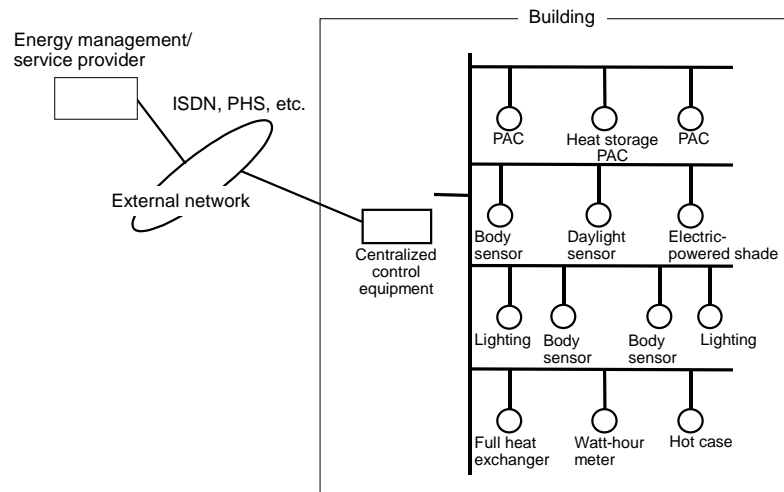


Fig. 4.2 Example of a Small/Medium-sized Building EMS Configuration

4.2 Small/Medium-sized Building/Store-dedicated EMS Functions

This section describes the functions supposed for small/medium-sized building/store-dedicated EMS and shows examples of each control function.

(1) Energy-saving control

- Linked action control

Outline: Works with existent/non-existent and lighting detecting outputs, changes operation status of each device, and executes optimum operation.

| Event | Control-target device and contents of control |
|---|---|
| Human Detection Sensor Existent/non-existent detection | Change of operation status of lighting, etc. ON/OFF, brightness adjustment |
| Daylight sensor detection brightness | Lighting brightness adjustment and shade adjustment by detecting brightness |

(2) Demand linked action function

- Inter-device linked action control

Outline: Prevents simultaneous operation between specific devices.

| Event | Control-target device and contents of control |
|------------------------|---|
| Electronic oven ON/OFF | Change of hot case operation status and air conditioner set temperature control/control release |

(3) Peak shift function

- Heat storage application device peak shift control

Outline: Heat storage device control to shift air conditioner load, such as daytime cooling to nighttime power.

| Event | Control-target device and contents of control |
|------------------------------|---|
| Cold/heat storage start time | Heat storage PAC: Cold/heat storage enabled |
| Cold/heat storage end time | Heat storage PCA: Cold/heat storage disabled |

(4) Demand time zone power reduction control

- Short time zone peak shift control

Outline: Avoids demand time zone and shifts energy use before and after an event.

| Event | Control-target device and contents of control |
|------------------------|--|
| Demand time zone start | Air conditioner setting temperature control (one degree up or down) |
| Demand time zone end | Air conditioner setting temperature control release (user-set temperature) |

- Power consumption control and power consumption monitoring priority control

Outline: Monitors power consumption in facilities and controls the device operation depending on increase/decrease from specified value according to previously determined priority.

| Event | Control-target device and contents of control |
|---|--|
| Based on monitoring demand time zone start or power consumption, it is predicted that specified value will be exceeded. | Changes operation contents of each device based on device priority and power monitoring result. Control example: Priority Low → high Capacity saving to air conditioner → Hot case thermo ON disabled → Reduction of lighting brightness |

4.3 Small/Medium-sized Building/Store-dedicated EMS Service Object

4.3.1 Medium/small building/store-dedicated EMS service class

The small/medium-sized building/store-dedicated EMS service classes resulting from modeling the small/medium-sized building/store EMS functions described in Section 3.2 are defined. Table 4.1 shows their relationship with small/medium-sized building/store-dedicated EMS service middleware functions.

Table 4.1 List of Small/Medium-sized Building/Store-dedicated EMS Service Classes

| Functional classification | Class name | Outline of function working with the service middleware |
|---------------------------|--|--|
| (1) Demand control | (A) Demand time zone power consumption monitoring priority control class | Monitors power consumption only in demand time zone and controls it based on priority upon occurrence of each event of specified power consumption conditions. |
| | (B) Demand linked action control class | Demand control linked action control between devices. |
| | (C) Demand time zone linked action control class | Executes linked action control of demand control only in demand time zone. |
| | (D) Demand schedule control class | Performs specification operation at specified time. |
| (2) Peak shift control | Peak shift control class | Starts or ends heat storage in connection with peak shift by heat storage equipment. |
| (3) Energy-saving control | Energy-saving control class | Linked action control between devices |

Fig. 4.3 shows a small/medium-sized building/store-dedicated service object model diagram. The service middleware function corresponding to each object is to monitor the occurrence of an event corresponding to the purpose and to operate the device on the specified algorithm as basic operations. Properties and services such as those shown in the small/medium-sized building/store EMS service object are provided as super-classes of these classes. The algorithm (method) for determining execution of the monitoring target, operation target, and operation contents differs with the service middleware corresponding to each object. Demand control objects have the four sub-classes shown in Table 3.1.

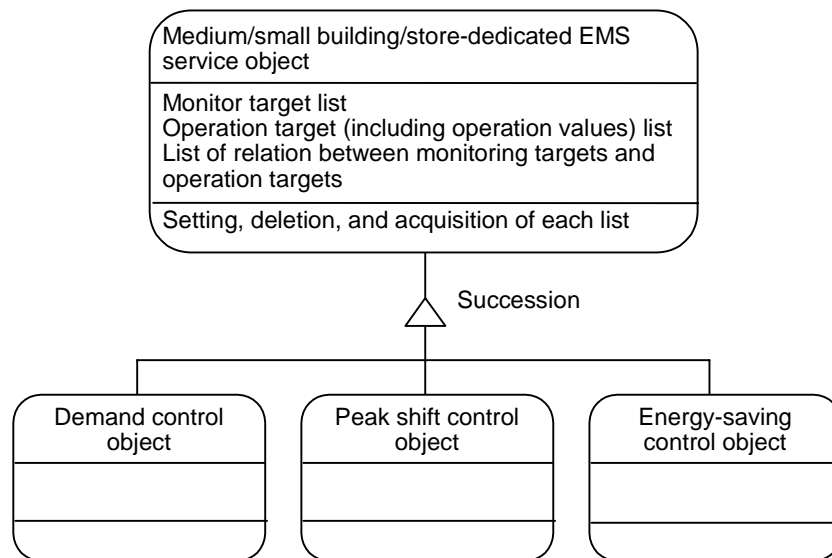


Fig. 4.3 Small/Medium-sized Building/Store-dedicated EMS Service Object Model Diagram

4.3.2 Details of small/medium-sized building/store-dedicated service classes

The details of each previously described service object are provided using data definitions that may be properties to be handled.

The target device objects for operation, monitoring, etc. (groups of properties and values of individual instances) are described below.

{Echonet Device Object} = {Echonet Address} + {Class} + {Instance} + {Property} + {Property Value}

This notation is based on the data dictionary notation method in the general function modeling technique.

{ } denotes data name.

{A} = {B} + {C} means that abstracted data A consists of data B and data C.

{B} = {a | b | c} means that data B may take the value of a, b, or c.

{A}n means that set n (n is at least one or more) of the information defined by data A is defined.

(1) Demand control

(A) Demand time zone power consumption monitoring priority control class

If the power consumption (Wh) exceeds the specified power value (Wh) during monitoring, the devices are operated in order of the preset order of priority.

- Monitoring target and conditions

$\{\text{Monitoring-target object}\} = \{\text{Echonet Device Object}\} + \{\text{Event}\}$

$\{\text{Event}\} = \{\text{Over} \mid \text{Under}\}$

- Operation priority and contents

$\{\text{Contents of operation priority}\} = \{ \{ \text{Priority} \} + \{ \text{Echonet Device Object} \} \} n$

(B) Demand linked action control class

The contents of the operation of one device are changed based on the event of another device. Events and operation targets are associated with each other. Used for sensor linked action and device linked action control, this class has the following functions:

(a) Individual 1:1 linked action control

One device is operated for one input event.

- Linked action event conditions

$\{\text{Linked action condition}\} = \{\text{Echonet Device Object}\} + \{\text{Event}\}$

$\{\text{Event}\} = \{\text{Over} \mid \text{Under}\}$

- Operation target

$\{\text{Operation target}\} = \{\text{Echonet Device Object}\}$

(b) Group linked action control 1:n n>1

Multiple devices are operated for one input event.

- Linked action event condition

$\{\text{Linked action condition}\} = \{\text{Echonet Device Object}\} + \{\text{Event}\}$

- Operation target

$\{\text{Operation target}\} = \{ \{ \text{Echonet Device Object} \} \} n$

(c) Multiple event/group linked action control {n:1} n n > 1

Multiple devices are operated using OR/AND of multiple event conditions as an event.

- Linked action event condition
 $\{\text{Event condition}\} = \{ \{\text{Echonet Device Object}\} + \{\text{Event}\} \} n$
- Relationship between event conditions
 $\{\text{Event relation}\} = \{\text{Or} \mid \text{And}\}$
- Operation target
 $\{\text{Operation target}\} = \{ \{\text{Echonet Device Object}\} \} n$

(C) Demand time zone linked action control class

Demand control linked action control is executed only in the demand time zone. This has the following functions:

a) Demand time zone setting

- $$\{\text{EMS schedule}\} = \{ \{\text{Start time}\} + \{\text{End time}\} \} n$$
- $$\{\text{Start time} \mid \text{End time}\} = \{\text{Day of the week}\} + \{\text{Hour}\} + \{\text{Minute}\}$$
- $$\{\text{Day of the week}\} = \{\text{Every day} \mid \text{Holiday} \mid \text{Weekday} \mid \text{Day} \mid \text{to} \mid \text{Sat.}\}$$

Other functions are the same as (B) Demand linked action control.

(D) Demand schedule control class

The specified device is operated at the specified time.

- Specified time
 $\{\text{Schedule}\} = \{ \{\text{Schedule No.}\} + \{\text{Schedule type}\} + \{*\text{Schedule value}\} \} n$
 $\{\text{Schedule type}\} = \{\text{Weekly} \mid \text{Daily}\}$
 $* \{\text{Weekly schedule value}\} = \{\text{Day of the week}\} + \{\text{Hour}\} + \{\text{Minute}\}$
 $* \{\text{Daily schedule value}\} = \{\text{Hour}\} + \{\text{Minute}\}$
- Specification operation
 $\{\text{Operation target}\} = \{\text{Schedule No.}\} + \{\text{Echonet Device Object}\} \} n$

(2) Peak shift control class

Peak shift control is executed by heat storage equipment. This has the following functions:

Heat storage enable/disable control

Cold/heat storage operation enable/disable is operated for the target heat storage equipment in the heat storage contract time.

Heat storage start time → Cold/heat storage operation enable

Heat storage end time → Cold/heat storage operation disable

- Heat storage time zone setting
 {Heat storage time zone}
 = { {Start time} + {End time} } n
 {Time} = {Day of the week} + {Hour} + {Minute}
 {Day of the week} = {Every day | Sun. | to |Sat.}
 - Operation target
 {Operation target} = { {Echonet Device Object} } n
 (Cold/heat storage enable/disable property of heat storage equipment)
- (3) Energy-saving control class
- The function is the same as (B) Demand linked action control. Conditions and operation priority differ.

4.4 Sequence

Fig. 4.4 shows the relationship between the small/medium-sized building/store-dedicated EMS service object and an application using an event trace diagram as an object model sequence.

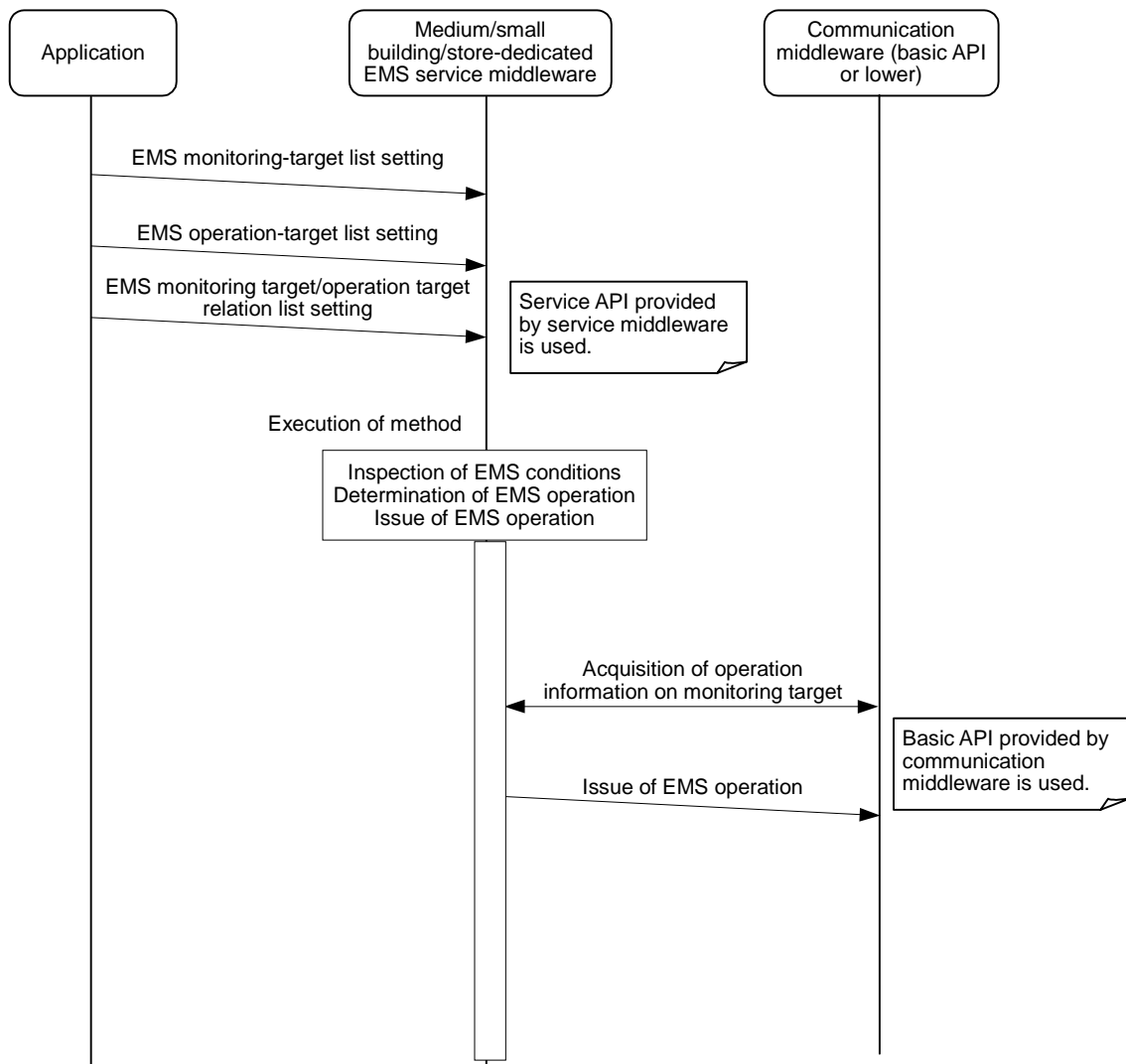


Fig. 4.4 Basic Operation Sequence of Small/Medium-sized Building/Store-dedicated EMS Service Middleware

4.5 Small/Medium-sized Building/Store-dedicated EMS Service API

This specification is under review according to the small/medium-sized building/store-dedicated EMS service middleware described in Sections 1 to 4 in this Chapter . Accordingly, regarding the API specification level, only functions are presented as a case proposition.

4.5.1 Basic Concept

This service API is designed around the basic concept specified for the service object API. It has been simplified as much as possible by preparing the object access interface at the super-class level resulting from abstracting all small/medium-sized building/store-dedicated EMS service middleware functions described in Section 4.3. In other words, when the monitoring target satisfies the specified conditions by monitoring the operation information (including measured values of sensors, etc.) of various devices installed in the facilities, the small/medium-sized building/store-dedicated EMS service executes the operations previously specified by the algorithm for each service object. Therefore, this API provides condition setting services for the monitoring/operation target as API.

4.5.2 List of Function Items

Table 4.2 shows a list of service API function items to access the small/medium-sized building/store-dedicated EMA service object.

Table 4.2 List of Function Items of the Small/Medium-sized Building/Store-dedicated EMS Service API

| API name | Definition of function |
|-----------------------------------|--|
| EMS monitoring-target setting API | An API to set a monitoring target as a condition for executing EMS control. This API sets the monitoring target condition to execute EMS control for operation information value (event) changes (occurrence of event, excess or lowering of value from the specified value, etc.) on the operation information of a specific device in the EMS time zone for each EMS type (demand control, peak shift, energy-saving) in the service middleware. |
| EMS operation-target setting API | An API to set the contents of operation to the EMS operation target as a condition for executing EMS control. The contents are a request for updating the operation information of a specific device. |
| EMS control contents setting API | This API sets a combination of the above monitoring target and an operation target as a condition for executing EMS control. The service middleware issues operation to the operation target on occurrence of a monitoring condition according to this combination condition. |