

**PUCC Device Discovery and Service Invocation Protocol  
(Version 3.0 – March 22, 2012)**

**Peer-to-Peer Universal Computing Consortium (PUCC)**

**Intellectual Property Notice**

©Copyright PUCC 2012. Confidential – Disclosure to PUCC members only. The information contained in this work is confidential and must not be reproduced, disclosed to non-PUCC-members without the prior written permission of PUCC, or used except as expressly authorized in writing by PUCC

**Table of Content**

1. Introduction .....	6
2. Terminology .....	6
2.1. Definitions .....	6
3. References .....	7
4. Goals and Requirements .....	7
4.1. Goals .....	7
4.2. Requirements .....	7
4.2.1. Application Independence .....	7
4.2.2. Interoperability .....	7
5. Protocol Overview .....	8
5.1. Protocol Stack .....	8
5.2. Protocol Design .....	8
6. PUCC Device Discovery and Service Invocation Protocol .....	9
6.1. Discover Method .....	9
6.1.1. Discover Message .....	10
6.1.1.1. Discover .....	10
6.1.1.2. DiscoverDestination .....	10
6.1.1.3. ID .....	10
6.1.1.4. Keyword .....	10
6.1.2. DiscoverResponse Message .....	11
6.1.2.1. DiscoverResponse .....	11
6.1.2.2. ResultList .....	11
6.2. Invoke Method .....	12
6.2.1. Invoke Message .....	12
6.2.1.1. Invoke .....	12
6.2.1.2. TargetDevice .....	12
6.2.1.3. TargetService .....	12
6.2.1.4. InputParameterList .....	12
6.2.1.5. Parameter .....	13
6.2.2. InvokeResponse Message .....	13
6.2.2.1. InvokeResponse .....	13
6.2.2.2. Status .....	13
6.2.2.3. Reason .....	13

***PUCC Device Discovery and Service Invocation Protocol***

- 6.2.2.4. OutputParameterList..... 14
- 6.2.2.5. Parameter ..... 14
- 6.3. Notify Method ..... 15
- 6.3.1. Notify Message..... 15
  - 6.3.1.1. Notify..... 17
  - 6.3.1.2. TargetDevice ..... 17
  - 6.3.1.3. SubscriptionID ..... 17
  - 6.3.1.4. EventDateTime..... 18
  - 6.3.1.5. EventList..... 18
  - 6.3.1.6. Event ..... 18
  - 6.3.1.7. StateVariableList ..... 18
  - 6.3.1.8. StateVariable ..... 18
  - 6.3.1.9. EncryptedStateVariables ..... 18
- 6.3.2. NotifyResponse Message ..... 20
  - 6.3.2.1. NotifyResponse ..... 20
  - 6.3.2.2. Status ..... 20
  - 6.3.2.3. Reason ..... 20
- 6.4. Subscribe Method ..... 21
- 6.4.1. Subscribe Message..... 22
  - 6.4.1.1. Subscribe..... 25
  - 6.4.1.2. TargetDevice ..... 25
  - 6.4.1.3. TargetDeviceList ..... 25
  - 6.4.1.4. RequestedDuration..... 25
  - 6.4.1.5. SubscriptionID ..... 25
  - 6.4.1.6. NotifyType..... 26
  - 6.4.1.7. EventConditionList ..... 26
  - 6.4.1.8. EventCondition ..... 26
  - 6.4.1.9. ConditionExpression ..... 26
  - 6.4.1.10. NotificationVariableList..... 27
  - 6.4.1.11. StateVariable ..... 27
  - 6.4.1.12. EventConditionIDList ..... 27
  - 6.4.1.13. EventConditionID..... 27
- 6.4.2. SubscribeResponse Message ..... 28
  - 6.4.2.1. SubscribeResponse ..... 29

***PUCC Device Discovery and Service Invocation Protocol***

6.4.2.2.	Status .....	29
6.4.2.3.	Reason .....	29
6.4.2.4.	SubscriptionID .....	30
6.4.2.5.	NegotiatedDuration .....	30
6.4.2.6.	MaxCount .....	30
6.5.	Unsubscribe Method .....	30
6.5.1.	Unsubscribe Message .....	30
6.5.1.1.	Unsubscribe .....	30
6.5.1.2.	SubscriptionID .....	30
6.5.2.	UnsubscribeResponse Message .....	31
6.5.2.1.	UnsubscribeResponse .....	31
6.5.2.2.	Status .....	31
6.5.2.3.	Reason .....	31
6.6.	Update Method .....	32
6.6.1.	Update Message .....	32
6.6.1.1.	Update .....	35
6.6.1.2.	TargetDevice .....	35
6.6.1.3.	EventConditionList .....	35
6.6.1.4.	EventCondition .....	35
6.6.1.5.	ConditionExpression .....	35
6.6.1.6.	NotificationVariableList .....	36
6.6.1.7.	StateVariable .....	36
6.6.2.	UpdateResponse Message .....	38
6.6.2.1.	UpdateResponse .....	39
6.6.2.2.	Status .....	39
6.6.2.3.	Reason .....	39
6.6.2.4.	Device .....	39
6.6.2.5.	PrmitiveDevice .....	39
6.7.	EventSolicited Method .....	40
6.7.1.	EventSolicited Message .....	40
6.7.1.1.	EventSolicited .....	41
6.7.1.2.	TargetDevice .....	41
6.7.1.3.	SubscriptionID .....	41
6.7.1.4.	Period .....	41

***PUCC Device Discovery and Service Invocation Protocol***

6.7.1.5. Start.....	41
6.7.1.6. End .....	41
6.7.1.7. LeaveEvent .....	41
Appendix A. Version History .....	42
Appendix B: Namespace Definitions .....	43
Appendix C: Notification of Event Difference by Notify Message.....	44
Appendix D: Exclusive Control of the Device.....	48
Appendix E: Notification of Events Accumulated in the Device .....	51
Appendix F: PUCC Event Monitoring after PUCC connection terminated .....	52
Appendix G: Access Control Operation Response by ACL (Access Control List).....	54

## **1. Introduction**

The widespread deployment of inexpensive communications technology, computational resources in the peer-to-peer networking infrastructure, and network-enabled end devices poses an interesting problem for end users: how to locate a particular network service or device out of hundreds of thousands of accessible services and devices. Another issue for services and devices location and discovery is how to describe the services and devices to encode such factors as cost, performance, location, and sub-device or service-specific capabilities.

This document intends to provide a highly-available, fault-tolerant, incrementally scalable service for discover and locate services or devices in the peer-to-peer network environment.

## **2. Terminology**

### **2.1. Definitions**

**[Device]** A Device is a physical system which has a Service. A Device may be configured with multiple logical Primitive Devices. A Device is able to generate an Event when the condition for its event generation are met.

**[Service]** A Service is a functional interface offered by a Device. A Service is able to define one or more input/output parameter(s). A Service is able to have one return value.

**[Event]** An Event is generated when its condition expression using state variables is satisfied.

**[Metadata]** Metadata represents an XML document which defines the Device's static data, state variables, Service, and condition expressions of Events.

The following terms are defined in Peer-to-Peer Architecture Specification.

- PUCC Core Protocol;
- Node;
- Community;
- Community ID;

### **3. References**

- PUCC Architecture specification Version 1.0
- PUCC Basic Protocol specification Version 1.0
- PUCC Device and Service Metadata template Version 1.0
- “Extensible Markup Language (XML) 1.0 (Second Edition)”, W3C Recommendation 6 October 2000, T. Bray et al. URL: <http://www.w3.org/TR/2000/REC-xml-20001006>
- “Hypertext Transfer Protocol -- HTTP/1.1”, RFC2616, R. Fielding et al., June 1999. URL:<http://www.ietf.org/rfc/rfc2616.txt>
- “UPnP Device Architecture v1.0.1 draft”  
URL: <http://www.upnp.org>
- “UUIDs and GUIDs”, Internet Draft, Paul J. Leach et al, August 1998.
- “Universal Resource Identifier”  
URL: <http://www.w3.org/Addressing/>
- “Document Type Definition”  
URL: <http://www.w3.org/TR/REC-xml#dt-doctype>

### **4. Goals and Requirements**

#### **4.1. Goals**

The goals of this specification are:

- ◆ To define device discovery and service invocation protocol over the Peer-to-Peer Core Protocol
- ◆ To leverage existing standards where possible, especially existing and evolving Internet standards.

#### **4.2. Requirements**

##### **4.2.1. Application Independence**

The protocol must be independent of particular applications.

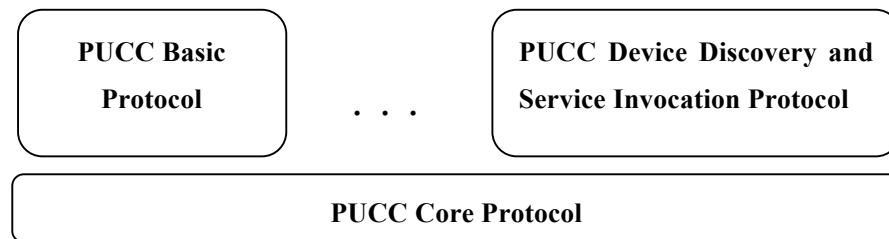
##### **4.2.2. Interoperability**

The protocol provides support for interoperability with other existing protocols such as UPnP.

## 5. Protocol Overview

### 5.1. Protocol Stack

PUCC Device Discovery and Service Invocation Protocol is defined over the core protocol, shown as following figure.



**Figure 5-1: Pucc Protocol Stack**

### 5.2. Protocol Design

PUCC Device Discovery and Service Invocation Protocol consists of following methods.

- Discover Method  
This method is defined to search devices or services on P2P networks.
- Invoke Method  
This method is defined to execute the service defined by the metadata.
- Notify Method  
This method is defined to notify of the status of devices or services.
- Subscribe Method  
This method is defined to subscribe to receive notify messages.
- Unsubscribe Method  
This method is defined to cancel a subscription to receive notify messages.
- Update Method  
This method is defined to add, update or delete event generation conditions in the metadata.
- EventSolicited Method  
This method is defined to solicit the events accumulated in the device.◦



## **6. PUCC Device Discovery and Service Invocation Protocol**

### **6.1. Discover Method**

This method is a Request/Response type message, defined for devices/services discovery. The search condition could be service/device URI or keyword. Discover message is a broadcast or a multicast message, and DiscoverResponse is a unicast message. In case of no search result, DiscoverResponse is not replied. In case a device is to be searched, “Device” shall be set to the type attribute of the Discover message. In case a service is to be searched, “Service” shall be set to the type attribute of the Discover message. In case search is performed by setting a URI of a device or service as a search condition, corresponding URI shall be set to the ID element of the Discover message. The device that received the Discover message shall confirm whether it matches with the URI of its device or its primitive device, and when they match, the DiscoverResponse message with metadata shall be returned. In case a device or service is to be searched using an arbitrary readable character string as a search condition, a corresponding character string shall be set to the Keyword element of the Discover message. The device that received the Discover message shall confirm whether there is a partial match with all the readable parts in the related metadata such as its device service, its primitive device, primitive device service. In case there is a matched metadata, the DiscoverResponse shall be returned with the metadata being set.

### 6.1.1. Discover Message

The following is definition of fields (Parameter) in Discover message.

**Table6-1-1: Fields of Discover Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
Discover	XML Fragment (DiscoveryDestination element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
DiscoverDestina tion	XML Fragment ( ID element or Keyword element)	-	-	1	Required
		type	“Service”, “Device” <string>	1	Required
ID	URI of Objective Metadata <anyURI>	-	-	1	Optional
Keyword	Keyword of Objective Metadata <string>	-	-	1	Optional

#### 6.1.1.1. Discover

This fragment is defined for device/service discovery.

#### 6.1.1.2. DiscoverDestination

This fragment is defined for condition declaration of device/service discovery. The search objective is declared by type attribute: “Device” or “Service.”

#### 6.1.1.3. ID

This element is defined for the declaration of URI metadata search condition.

#### 6.1.1.4. Keyword

This element is defined for the declaration of keyword search condition.

### 6.1.2. DiscoverResponse Message

The following is definition of fields (Parameter) in DiscoverResponse message.

**Table6-1-2: Fields of DiscoverResponse Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
DiscoverResponse	XML fragment (ResultList element)	-	-	1	Required
		xmlns	Name space identifier (URI) of Pucc Device Discovery and Service Invocation Protocol <anyURI>	1	Required
ResultList	XML fragment (Device element, Primitive Device element, Service element)	-	-	1	Required

#### 6.1.2.1. DiscoverResponse

This fragment is defined for the response message for device discovery.

#### 6.1.2.2. ResultList

This fragment is defined for the result of discovery. Instance of result is embedded in Device element, Primitive Device element or Service element.

## 6.2. Invoke Method

This method is a Request/Response type message, defined for service execution.

### 6.2.1. Invoke Message

The following is definition of fields (Parameter) in Invoke message.

**Table6-2-1: Fields of Invoke Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
Invoke	XM Fragment (TargetDevice element, TargetService element, InputParameterList element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUC C Device Discovery and Service Invocation Protocol <anyURI>	1	Required
TargetDevice	Device ID for the target device or primitive device <string>	-	-	1	Required
TargetService	URI for the target service <anyURI>	-	-	1	Required
InputParameterList	XML Fragment (Parameter element)	-	-	1	Required
Parameter	Parameter value.			1 or more	Optional
	This data type follows datatype attribute.	name	<string>	1	Required
		datatype	<string>	1	Required

#### 6.2.1.1. Invoke

This fragment is defined for service execution

#### 6.2.1.2. TargetDevice

This fragment declares the device ID for target device or target primitive device. Device ID is assigned in the target device or primitive device metadata.

#### 6.2.1.3. TargetService

This fragment declares the URI for target service.

#### 6.2.1.4. InputParameterList

This element enumerates the parameter.

**6.2.1.5. Parameter**

This element is defined for parameter description in the target service metadata.

**6.2.2. InvokeResponse Message**

The following is definition of fields (Parameter) in InvokeResponse message.

**Table6-2-2 : Fields of InvokeResponse Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
InvokeResponse	XM Fragment (Status element, Reason element, OutputParameterList element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
Status	“Success” or “Failure” <string>	-	-	1	Required
Reason	“ResourceDepletion”, “PermissionDenied”, Reason of the failure <string>	-	-	1	Depend on Status element
OutputParameterList	XML Fragment (Parameter element)	-	-	1	Depend on Status element
Parameter	Parameter value. This data type follows datatype attribute.	-	-	1 or more	Optional
		name	<string>	1	Required
		datatype	<string>	1	Required

**6.2.2.1. InvokeResponse**

This fragment is defined for service execution report to the service requester. The returned content depends on the devices. Each device has its own unique response message.

**6.2.2.2. Status**

This element returns the status of the response message. The value is “Success” or “Failure”.

**6.2.2.3. Reason**

This element returns the reason of the failure. If the Status element value is “Failure”, this element is required. Don’t

***Pucc Device Discovery and Service Invocation Protocol***

set this element, if it is not so.

**6.2.2.4. OutputParameterList**

This element enumerates the parameter. If the Status element value is “Success”, this element is required. Don’t set this element, if it is not so.

**6.2.2.5. Parameter**

This element is defined for parameter description in the target service metadata.

### 6.3. Notify Method

This method is an advertise type message or a Request/Response type message,, defined for event notification.

Advertise type message demands no confirmation from the node receiving Notify message.

Request/Response type message demands confirmation from the node receiving Notify message.

#### 6.3.1. Notify Message

The following is a definition of the fields (Parameter) in the notify message.

**Table6-3-1: Fields of Notify Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
Notify	XML Fragment (TargetDevice element, SubscriptionID element, EventDateTime element, EventList element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
		confirmation	“yes” or “no” <string>	1	Optional Default =“no”
		closure	“yes” or “no” <string>	1	Optional Default =“no”
TargetDevice	Device ID for target device or target primitive device <string>	-	-	1	Optional
SubscriptionID	<string>	-	-	1	Required
EventDateTime	<dateTime>	-	-	1	Optional
EventList	XML Fragment (Event element)	-	-	1	Required
Event	XML Flagment (TargetDevice element,StateVariable List element, EncryptedStateVariabl	-	-	1 or more	Required
		notificationID	Event Notification ID <integer>	1	Optional
		previousNotific	Previous Event	1	Optional

***PUCC Device Discovery and Service Invocation Protocol***

	es element)	ationID	Notification ID. To be set to notify the difference. <integer>		
		eventConditionID	Event Condition ID <string>	1	Optional
StateVariable List	XML fragment (StateVariable element)	-	-	1	Optional
StateVariable	Value of the state variable. This data type follows datatype attribute. The encrypted data are set in a format that can be described in XML documents such as base64Binary and hexBinary.	-	-	1 or more	Optional
		name	<string>	1	Required
		datatype	<string>	1	Required
		encrypted	"true" "false" Indicates whether encryption is used or not. Default value is "false." <string>	1	Optional
		encryptionMethod	Encryption method. This is not set when encryption is not used. <string>	1	Optional



**PUCC Device Discovery and Service Invocation Protocol**

		differenceType	Indicates whether the value is the increase type or the decrease type when the value of the State Variable element is set to the difference of State Variables. "+", "or", "-" <string>		Optional
EncryptedStateVariables	The values of StateVariables encrypted all together. <string>	-	-	1	Optional
		encryptionMethod	Encryption method <string>	1	Required

**6.3.1.1. Notify**

This fragment is defined for sending messages to notify of the change of device state.

This element has a confirmation attribute. This attribute value is “yes” or “no”. If the confirmation attribute is “yes”, this Notify message demands confirmation from the node receiving this Notify message, so the MessageType element in the PUCC Core Protocol parameter of this Notify message is mapped to “Request”. If the confirmation attribute is “no”, this Notify message demands no confirmation from the node receiving this Notify message, so the MessageType element in the PUCC Core Protocol parameter of this Notify message is mapped to “Advertise”.

This attribute is optional. If this attribute is omitted, this attribute value is “no”.

This element has a closure attribute. This attribute value is “yes” or “no”. If this attribute is “yes”, the event notification is completed with the same SubscriptionID. If this attribute is “no”, the event notification is continued with the same SubscriptionID.

The TargetDevice element is specified, when the event source device of all Event elements which are set in a Notify message includes one device.

**6.3.1.2. TargetDevice**

This fragment declares the device ID for the target device or the target primitive device. Device ID is assigned in the target device or primitive device metadata.

**6.3.1.3. SubscriptionID**

This fragment declares the ID for subscription.

SubscriptionID is unique among the PUCC nodes. When the node which has conducted Subscribe releases the

session, the SubscriptionID is also released in the both PUCC nodes which Subscribed and was Subscribed to.

#### **6.3.1.4. EventDateTime**

This fragment declares the date and time when the change of the state for the target device or the target primitive device.

#### **6.3.1.5. EventList**

This element enumerates the Event.

#### **6.3.1.6. Event**

This element sets StateVariableList element or EncryptedStateVariables element. The Event elements has the notificationID attribute, the previousNotificationID attribute and the eventConditionID attribute. The notificationID attribute is set to the Event notification ID. The notificationID shall be unique from the time of the Subscribe message to the Unsubscribe message between the event subscriber and the device. This specification document does not specify any initial value of the notification ID. The increase/decrease value increments by 1 as an event occurs. When the difference between the state variables is notified, the previousNotificationID attribute is set to the previous Event Notification ID in order to indicate the Event Notification ID from which the difference is demonstrated. Refer to Appendix C for samples of State Variable difference notification. The eventCondition ID attribute is set to the Event Condition ID in order to indicate which event generation condition is satisfied.

The TargetDevice element is specified, when the event source devices of all Event elements which are set in a Notify message are more than one device.

#### **6.3.1.7. StateVariableList**

This element enumerates the StateVariable.

When the values of state variables are encrypted all together, the EncryptedStateVariables element is used instead of the StateVariableList element.

#### **6.3.1.8. StateVariable**

This element designates the state variable. This element must be set a state variable after state transition. A state variable other than that after state transition can also be set. This element has a name attribute and a datatype attribute, differenceType attribute. The name attribute designates the name of the state variable defined in the target device metadata. The datatype attribute designates the data type of the state variable defined in the target device metadata. In setting the StateVariable element to the difference of the state variables, the differenceType attribute is set to the plus "+" or the minus "-" in order to indicate whether the state variable difference is the "increase" type or the "decrease" type. When the values of state variables are encrypted all together, EncryptedStateVariables is used instead of StateVariableList.

#### **6.3.1.9. EncryptedStateVariables**

This element sets the values of state variables encrypted all together.

When the values of state variables are encrypted individually, StateVariableList is used in stead of StateVariableList.

***PUCC Device Discovery and Service Invocation Protocol***

The following is a sample of the Notify message sent when a normal event notification is conducted.

```
<Core xmlns=" Namespace of PUCC Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
    <Notify xmlns=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
      <SubscriptionID>SubscriptionID</SubscriptionID>
      <EventDateTime>2008-02-04T01:12:00.123+09:00</EventDateTime>
      <EventList>
        <Event notificationID=" 1" eventConditionID=" event1" >
          <StateVariableList>
            <StateVariable name=" Temperature" datatype=" integer" >40</StateVariable>
          </StateVariableList>
        </Event>
      </EventList>
    </Notify>
  </MsgBody>
</Core>
```

**Figure 6.3.2-1. Notify message for normal event notification**

The following is a sample of the Notify message sent when a difference of state variable is notified.

```
<Core xmlns=" Namespace of PUCC Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
    <Notify xmlns=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
      <SubscriptionID>SubscriptionID</SubscriptionID>
      <EventDateTime>2008-02-04T01:12:00.123+09:00</EventDateTime>
      <EventList>
        <Event notificationID=" 2" previousNotificationID=" 1" eventConditionID=" event1" >
          <StateVariableList>
            <StateVariable name=" Temperature" datatype=" integer" differenceType=" +" >5</StateVariable>
          </StateVariableList>
        </Event>
      </EventList>
    </Notify>
  </MsgBody>
</Core>
```

**Figure 6.3.2-2. Notify message for state variable difference notification**

### 6.3.2. NotifyResponse Message

The following is a definition of the fields (Parameter) in the NotifyResponse message. This message is used when the Request type Notify message is received. This message is not used when the Advertise type Notify message is received.

**Table6-3-2 : Fields of NotifyResponse Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
NotifyResponse	XM Fragment (Status element, Reason element)	-	-	1	Required
		xmlns	Name space identifier (URI) of Pucc Device Discovery and Service Invocation Protocol <anyURI>	1	Required
Status	“Success” or “Failure” <string>	-	-	1	Required
Reason	Reason of the failure. <string>	-	-	1	Depend on Status element

#### 6.3.2.1. NotifyResponse

This fragment is defined for event notification report to the service requester. This message is used when the Request type Notify message is received. This message is not used when the Advertise type Notify message is received.

#### 6.3.2.2. Status

This element returns the status of the response message. The value is “Success” or “Failure”.

#### 6.3.2.3. Reason

This element returns the reason of the failure. If the Status element value is “Failure”, this element is required. Don’t set this element, if it is not so.

#### **6.4. Subscribe Method**

This method is a Request/Response type message, defined to subscribe to receive notify messages. At initial event registration (not at event registration update, the Subscribe message shall be transmitted to the device without the SubscriptionID being set. In case the event registration is successful, the device allocates a unique SubscriptionID, sets it to the SubscribeResponse message and returns it. When an event occurs in a device, the device sets the SubscriptionID allocated at device registration and transmits a Notify message. If the event registration is conducted without setting the RequestedDuration, the registration becomes infinitely valid until the event registration release is conducted with the Unsubscribe method. In case the event registration is conducted by setting the RequestedDuration, the registration shall be released automatically when the set time has been passed. When one does not want the registration to be released, the Subscribe message shall be transmitted by setting the SubscriptionID allocated to the event being registered before the RequestedDuration time has been passed. In other words, at event registration update, the SubscriptionID shall be set to the Subscribe message and transmitted to the device.

The Subscribe message has the exclusive attribute. Transmission of a Subscribe message which sets the exclusive attribute to “yes” allows exclusive control of event registration by another event subscriber. When the event subscriber to which exclusive control has been designated terminates its event registration, another event subscriber is allowed to conduct event registration. Event control accompanying exclusive control will fail, however, if event registration is already made for the target device by another event subscriber. In such a case, the device sets the Result value of the SubscribeResponse message to Failure and sets a comment in the Reason value stating that event registration has already been conducted by another event subscriber, both of which are returned to the sender of the Subscribe message.

The Subscribe message has the NotifyType element. When the event subscriber sends to a device a Subscribe message with its NotifyType elements being set to "solicited," the device accumulates the events that have happened until it receives an EventSolicited message from the event subscriber. Upon reception of an EventSolicited message from the event subscriber, the device notifies the event subscriber of the events it has accumulated so far all together using the Notify message.

### 6.4.1. Subscribe Message

The following is a definition of the fields (Parameters) in the Subscribe message.

Event registration using the Subscribe message is conducted by means of one of the following methods.

(1) When no event generation condition is specified:

When a Subscribe message with neither EventConditionList element nor the eventConditionIDList element being set is sent to the device, the device starts monitoring the event for all event generation conditions described in the device metadata.

(2) When a specific event condition described in the metadata is selected:

When a Subscribe message with the eventConditionID element of the EventConditionIDList element being set to the ID of a specific event generation condition described in the metadata is sent to the device, the device starts monitoring the event only for the event generation condition of which ID is designated in the EventConditionIDList element.

(3) When a new event generation condition is registered:

When a Subscribe message with the EventCondition element of the EventConditionList element being set to the expression of the event generation condition to be newly registered (ConditionExpression element) and the state variables to be notified of when the event generation condition is fulfilled (NotificationVariableList element), the device starts monitoring the event solely based on the event generation condition designated in the EventConditionIDList element in the Subscribe message. An event generation condition that is newly registered by the Subscribe message is deleted from the device when the event registration is terminated by the Unsubscribe method, and is not to be added to the list of event generation conditions in the device metadata. The Update method needs to be used in order to update the event generation conditions in the device metadata.

Event registration will fail when the id attribute value of the EventCondition element set in the Subscribe message is the same as any of the event generation conditions described in the device metadata.

With regard to how to describe expressions for Pucc-specific event generation conditions, refer to the guideline for describing generation condition expressions in the Pucc Metadata Template Specification.

**Table6-4-1: Fields of Subscribe Message**

<b>Element name</b>	<b>Element Value</b> <datatype> = XML Schema datatype	<b>Attribute name</b> <b>(if present)</b>	<b>Attribute Value</b> <datatype> = XML Schema datatype	<b>Occurrence</b>	<b>Status</b>
Subscribe	XML Fragment	-	-	1	Required

**PUCC Device Discovery and Service Invocation Protocol**

	(TargetDevice element, TargetDeviceList element, RequestedDuration element, SubscriptionID element, EventConditionList element, EventConditionIDList element)	xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
		exclusive	With or without exclusive control of the device. "yes" or "no." The default value is "no." <string>	1	Optional
TargetDevice	Device ID for target device or target primitive device <string>	-	-	1 or more	Optional
TargetDevice List	XML Fragment (TargetDevice element)	-	-	1	Optional
RequestedDuration	<integer>	-	-	1	Optional
SubscriptionID	<string>	-	-	1	Optional
NotifyType	The method of event notification from the device. "solicited" or "unsolicited". The default value is "unsolicited."	-	-	1	Optional
EventConditionList	XML Fragment (EventCondition element)	-	-	1	Optional
EventCondition	XML Fragment	-	-	1 or more	Required

***PUCC Device Discovery and Service Invocation Protocol***

on	(ConditionExpression element,NotificationVariableList element)	id	Event condition ID	1	Required
ConditionExpression	Event condition expression. When the condition expression and either the time or period attribute are set, the event will be generated if the described condition is satisfied at the moment the designated period or time comes. <string>	-	-	1	Required
		format	The description method of the condition expression such as "PUCC","SQL"and"SPARQL." The default value is "PUCC." <string>	1	Optional
		period	Event generation period <string>	1	Optional
		time	Event generation time <time>	1	Optional
NotificationVariableList	XML Fragment (StateVariable element)	-	-	1	Required
StateVariable	State variables to be sent in event notification. Nothing is set for the element value.	-	-	1 or more	Optional
		name	The name of the state variable to be notified in the event notification. . <string>	1	Required
		notificationType	Indicates whether the difference between state variables designated in the name attribute is notified of or just the values of the state	1	Optional



***PUCC Device Discovery and Service Invocation Protocol***

			variables are reported as they are. "normal" or "difference". The default value is "normal." <string>		
EventConditionIDList	XML Fragment (EventConditionID element)	-	-	1	Optional
EventConditionID	Event Condition ID	-	-	1 or more	Required

**6.4.1.1. Subscribe**

This fragment is defined for subscription to receive notify messages.

The Subscribe element has the exclusive attribute. When exclusive control over the device it is subscribing to is needed to exclude event registration from another event subscriber, the exclusive attribute is set to "yes," Otherwise, it is set to "no." The default value is "no."

**6.4.1.2. TargetDevice**

This fragment declares the device ID for target device or target primitive device. Device ID is assigned in the target device or primitive device metadata. It is necessary to set the TargetDevice element to either of the Subscribe element under or the TargetDeviceList element under. When the event is registered in two or more devices, it is set under the TargetDeviceList element .

The event observation is done for the both of device and primitive device, when the target device is a device which has primitive device and doesn't have an event condition which is specified by the EventConditionList.

The event observation is done only for the specified device, when the target device is a device which has an EventConditionList.

**6.4.1.3. TargetDeviceList**

This element is a list of device ID when the collective event registration is done for plural devices.

**6.4.1.4. RequestedDuration**

Requested duration until subscription expires, either number of seconds or 0. If this element value is set to 0, or this element is empty or not specified, it is regarded as infinite.

**6.4.1.5. SubscriptionID**

This fragment declares the ID for subscription. This element is required when a subscription is renewed. This element

is unused when subscription is initialized..

The Subscription ID is unique among the PUC nodes. When the PUC node which conducted Subscribe releases the session, the SubscriptionID is released in the both PUC nodes which Subscribed and was Subscribed.

#### **6.4.1.6. NotifyType**

This element specifies the way events are notified of from the device. This element is set to “solicited” or “unsolicited.” When it is set to “solicited,” the device accumulates the generated events.

As for the way to obtain the events accumulated in the device, refer to Appendix E.

When this element is set to “unsolicited.” the device performs event notification using a Notify message every time an event occurs.

##### **6.4.1.5.1 Timing for Deletion of Events the Device Accumulates**

Event are managed by the device for every event subscriber. When event registration is released using Unsubscribe, all the events accumulated in the device are deleted.

When the maximum number of events the device can store is determined, the device returns a SubscribeResponse message to the event subscriber with the MaxCount element in the message being set to the maximum number. When the number of events accumulated in the device reaches the maximum number, the device deletes events starting from the oldest one.

The device deletes all events it has accumulated after obtaining the event if the LeaveEvent element in the EventSolicited message is set to “no” at the time when the event is obtained.

#### **6.4.1.7. EventConditionList**

This element enumerates the EventCondition.

#### **6.4.1.8. EventCondition**

This element designates an ConditionExpression and NotificationVariableList.

This element specifies the condition for new event generation in a Subscribe message. The EventCondition element has the id attribute. The id attribute is set to the Condition ID of the event to be Subscribed.

The EventConditionID is unique in the PUC device. The ID shall not duplicate the event generation ID described in the device metadata. For a message sample, [refer to Figure 6.4.1-3.](#)

#### **6.4.1.9. ConditionExpression**

This element defines an event condition expression. The ConditionExpression element has the format attribute, the period attribute and the time attribute.

The format attribute is set to the description type of the condition expression such as "PUC," "SQL," and "SPARQL." The default value is “PUC.” The period attribute is set to the event generation period while the time attribute is set to the event generation time.

When the condition expression element and either the time attribute or the period attribute are specified, the event will be generated at the moment the period or time specified comes if the condition described in the condition expression is satisfied.

**6.4.1.10. NotificationVariableList**

This element enumerates the StateVariable.

**6.4.1.11. StateVariable**

No value is set for this element. The StateVariable element has the name attribute and the notificationType attribute. The name attribute is set to the name of the state variable to be notified of in event notification. The notificationType attribute is set to either the value of the state variable or the value (“normal” or “difference”) indicating that the difference of the state variables is to be notified of. As for the way to notify of the difference of state variables, refer to Appendix C.

**6.4.1.12. EventConditionIDList**

This element enumerates the EventConditionID.

**6.4.1.13. EventConditionID**

This element designates an EventConditionID.

This element is used to specify the event generation condition described in the device metadata in a Subscribe message. The EventConditionID is unique in the PUC device. The EventCondition ID shall be set to the ID of the event generation condition described in the device metadata. As for a sample of message, refer to [Figure 6.4.1-2](#).

The following is a sample of the Subscribe message sent when the event generation condition is not specified.

```
<Core xmlns=" Namespace of PUC Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345. 2002-12-20T16: 15: 32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of PUC Device Discovery and Service Invocation Protocol" >
    <Subscribe xmlns=" Namespace of PUC Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
    </Subscribe>
  </MsgBody>
</Core>
```

**Figure 6.4.1-1. Subscribe message with the event generation condition unspecified**

The following is a sample of the Subscriber message sent when the event generation condition described in the metadata is selected.

```
<Core xmlns=" Namespace of PUC Core Protocol" >
  <MsgType>Request</MsgType>
```

**Pucc Device Discovery and Service Invocation Protocol**

```

<MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
<Destination>
  <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
</Destination>
<Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
<ComType>Unicast</ComType>
<MsgBody protocol=" Namespace of Pucc Device Discovery and Service Invocation Protocol" >
  <Subscribe xmlns=" Namespace of Pucc Device Discovery and Service Invocation Protocol" >
    <TargetDevice>TargetDeviceID</TargetDevice>
    <EventConditionIDList>
      <EventConditionID>event1</EventConditionID>
      <EventConditionID>event2</EventConditionID>
    </EventConditionIDList>
  </Subscribe>
</MsgBody>
</Core>

```

**Figure 6.4.1-2. Subscribe message when the event generation condition described in the metadata is selected**

The following is a sample of the Subscribe message sent when the event generation condition described in the metadata is selected.

```

<Core xmlns=" Namespace of Pucc Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of Pucc Device Discovery and Service Invocation Protocol" >
    <Subscribe xmlns=" Namespace of Pucc Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
      <EventConditionList>
        <EventCondition id=" event3" >
          <ConditionExpression>Temperature<lt;20</ConditionExpression>
          <NotificationVariableList>
            <StateVariable name=" Temperature" />
            <StateVariable name=" WaterVolume" />
          </NotificationVariableList>
        </EventCondition>
      </EventConditionList>
    </Subscribe>
  </MsgBody>
</Core>

```

**Figure 6.4.1-3. Subscribe message sent when a new event generation condition is registered**

## 6.4.2. SubscribeResponse Message

The following is a definition of the fields (Parameter) in the SubscribeResponse message.

**Table6-4-2 : Fields of SubscribeResponse Message**

***PUCC Device Discovery and Service Invocation Protocol***

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
SubscribeResponse	XML Fragment (Status element, Reason element, SubscriptionID element, NegotiatedDuration element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
Status	“Success” or “Failure” <string>	-	-	1	Required
Reason	“ResourceDepletion”, “PermissionDenied”, Reason of the failure <string>	-	-	1	Depend on Status element
SubscriptionID	<string>	-	-	1	Depend on Status element
NegotiatedDuration	<integer>	-	-	1	Optional
MaxCount	<integer>	-	-	1	Optional

**6.4.2.1. SubscribeResponse**

This fragment is defined for subscription report to the subscription requestor.

**6.4.2.2. Status**

This element returns the status of the response message. The value is “Success” or “Failure”.

**6.4.2.3. Reason**

This element returns the reason of the failure. If the Status element value is “Failure”, this element is required. Don’t set this element, if it is not so.

Value	Description
ExclusiveControlOfEventRegistration	Exclusive Control is conducted over the node to exclude event subscription to it from another node.
EventConditionIDAlreadyDefined	The event generation condition ID is already defined in the device metadata.
IllegalConditionExpression	The expression for the event generation condition is illegal.

**Table 6.4.2.3-1: Values of Reason element**

#### 6.4.2.4. SubscriptionID

This fragment declares the ID for subscription. If the Status element value is “Success”, this element is required. Don’t set this element, if it is not so.

The SubscriptionID is unique among the Pucc nodes. When the Pucc node which Subscribed releases the session, the SubscriptionID will be released in the both Pucc nodes which Subscribed and was Subscribed to.

#### 6.4.2.5. NegotiatedDuration

Actual duration until subscription expires, either number of seconds or 0. If this element value is set to 0, or this element is empty or not specified, it is regarded as infinite. If the Status element value is “Success”, this element can be set. Don’t set this element, if it is not so.

#### 6.4.2.6. MaxCount

The maximum number of event that a device is able to accumulate. This element is set when the NotifyType of the Subscribe message is set to “solicited,” and the number of events that can be accumulated in the device is limited.

### 6.5. Unsubscribe Method

This method is a Request/Response type message, defined for canceling subscription.

#### 6.5.1. Unsubscribe Message

The following is a definition of the fields (Parameter) in the unsubscribe message.

**Table6-5-1: Fields of Unsubscribe Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
Unsubscribe	XML Fragment (SubscriptionID)	-	-	1	Required
		xmlns	Name space identifier (URI) of Pucc Device Discovery and Service Invocation Protocol <anyURI>	1	Required
SubscriptionID	<string>	-	-	1	Required

##### 6.5.1.1. Unsubscribe

This fragment is defined for canceling the subscription.

##### 6.5.1.2. SubscriptionID

The SubscriptionID is unique among the Pucc nodes. When the Pucc node which conducted Subscribe releases the session, the SubscriptionID will be released in the both Pucc nodes which Subscribed and was Subscribed to.

### 6.5.2. UnsubscribeResponse Message

The following is a definition of the fields (Parameter) in the UnsubscribeResponse message.

**Table6-5-2 : Fields of UnsubscribeResponse Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
Unsubscrib eResponse	XML Fragment (Status element, Reason element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
Status	“success” or “Failure” <string>	-	-	1	Required
Reason	“ResourceDepletion”, Reason of the failure <string>	-	-	1	Depend on Status element

#### 6.5.2.1. UnsubscribeResponse

This fragment is defined for cancel report to the subscription requestor.

#### 6.5.2.2. Status

This element returns the status of the response message. The value is “Success” or “Failure”.

#### 6.5.2.3. Reason

This element returns the reason of the failure. If the Status element value is “Failure”, this element is required. Don’t set this element, if it is not so.

## 6.6. Update Method

This method is a Request/Response type message, defined for updating metadata.

When a node wants to update the metadata, it sends an Update message which contains the parameter necessary for device metadata update to the device node.

The part of the metadata to be updated shall only be the list of the event generation conditions(EventConditionList element) for the device and primitive device metadata.

No update is made to the service metadata or any other area than the event generation condition list.

The following are the list of updates that can be made as for the event generation list for the device metadata.

- (1) To add a new event generation condition

To add a new event generation condition to the list of the existing event generation conditions in the metadata.

- (2) To delete the event generation condition of the designated ID.

To delete the event generation condition of the designated ID among the existing event generation conditions in the metadata.

- (3) To update the event generation condition of the designated ID.

To update the event generation condition of the designated ID among the existing event generation conditions in the metadata.

The device node updates the device metadata in accordance with the parameter set in the Update message.

The device node returns an UpdateResponse message which contains the result of the successful or failed metadata update and the updated metadata (the device data or the primitive device data) if the result is successful to the event subscriber.

The metadata of the device cannot be updated if the device is already registered by another event subscriber for events.

If the device receives an Update message while it is registered for events, it returns an UpdateResponse(Failure) to the event subscriber.

### 6.6.1. Update Message

The following is a definition of the fields (Parameter) in the update message.

**Table6-6-1: Fields of Update Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
Update	XML Fragment	-	-	1	Required



***PUCC Device Discovery and Service Invocation Protocol***

	(TargetDevice element,EventCondition element)	xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
TargetDevice	Device ID for target device or target primitive device <string>	-	-	1	Required
EventConditionList	XML Fragment (EventCondition element)	-	-	1	Required
EventCondition	XML Fragment (ConditionExpression element,NotificationVariableList element)	-	-	1 or more	Required
		type	Types of updates of the event generation condition "add," "delete," and "update." <string>	1	Required
		id	id is set to the event condition ID to be newly added when the type attribute value is "add." id is set to the event condition ID to be deleted when the type element value is "delete." id is set to the event condition ID to be updated when the type element value is	1	Required

**PUCC Device Discovery and Service Invocation Protocol**

			"update." <string>		
ConditionExpression	Event condition expression. When the condition expression and either the time or period attribute are set, the event will be generated if the described condition is satisfied at the moment the designated period or time comes.	-	-	1	Required
		format	The description method of the condition expression such as "PUCC," "SQL," and "SPARQL." The default value is "PUCC." <string>	1	Optional
		period	Event generation period. <string>	1	Optional
		time	Event generation time. <time>	1	Optional
NotificationVariableList	XML Fragment (StateVariable element)	-	-	1	Required
StateVariable	Nothing is set for the element value.	-	-	1 or more	Optional
		name	The name of the state variable notified of in the event notification. <string>	1	Required
		notificationType	Indicates whether notify of the value of the state variable or the difference of the state variables designated in the name attribute. "normal" or	1	Optional

***PUCC Device Discovery and Service Invocation Protocol***

			"difference" The default value is "normal." <string>		
--	--	--	---	--	--

**6.6.1.1. Update**

This fragment is defined for updating metadata.

**6.6.1.2. TargetDevice**

This fragment declares the device ID for target device or target primitive device. Device ID is assigned in the target device or primitive device metadata.

**6.6.1.3. EventConditionList**

This element enumerates the EventCondition.

**6.6.1.4. EventCondition**

This element includes the event condition expression and the list of the state variables to be notified of when the event condition is met.

The EventCondition element has the type attribute and the id attribute. The type attribute is set to either type of event generation conditions ("add," "delete" or "update"). The id attribute is set to the ID of the event condition to be newly added when the type attribute is "add." The EventConditionID is unique in the PUCC device. The EventConditionID shall not be the same as any of the event condition IDs described in the metadata. When the type element value is "delete," the EventConditionID is set to the ID of the event condition to be deleted. When the type element value is "update," the EventConditionID is set to the ID of the event condition to be updated. When the type attribute value is "delete," no value is set as the EventCondition element value.

**6.6.1.5. ConditionExpression**

This element defines an event condition expression. The ConditionExpression element has the format attribute, the period attribute and the time attribute.

The format attribute is set to the description type of the condition expression such as "PUCC," "SQL," and "SPARQL." The default value is "PUCC." The period attribute is set to the event generation period while the time attribute is set to the event generation time.

When the condition expression element and either the time attribute or the period attribute are specified, the event will be generated at the moment the period or time specified comes if the condition described in the condition expression is satisfied.

#### 6.6.1.6. NotificationVariableList

This element lists the state variables to be notified of when the event condition is satisfied.

#### 6.6.1.7. StateVariable

No value is set for this element. The StateVariable element has the name attribute and the notificationType attribute. The name attribute is set to the name of the state variable to be notified of in the event notification. The notificationType attribute is set to either the value of the state variable or the value (“normal” or “difference”) indicating that the difference of the state variables is to be notified of. As for the way to notify the difference of state variables, refer to Appendix C.

The following is a sample of the Update message when a new event condition is added to the metadata..

```

<Core xmlns=" Namespace of PUCC Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
    <Update xmlns=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
      <EventConditionList>
        <EventCondition type="add" id="event3">
          <ConditionExpression>WaterVolume<180</ConditionExpression>
          <NotificationVariableList>
            <StateVariable name="Temperature"/>
            <StateVariable name="WaterVolume"/>
          </NotificationVariableList>
        </EventCondition>
      </EventConditionList>
    </Update>
  </MsgBody>
</Core>

```

**Figure 6.6.1-1. Update message sent when a new event condition is added to the metadata.**

***PUCC Device Discovery and Service Invocation Protocol***

The following is a sample of the Update message sent when an event condition in the metadata is deleted..

```
<Core xmlns=" Namespace of PUCC Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345. 2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
    <Update xmlns=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
      <EventConditionList>
        <EventCondition type="delete" id="event2"/>
      </EventConditionList>
    </Update>
  </MsgBody>
</Core>
```

**Figure 6.6.1-2.** Update message sent when an event condition in the metadata is deleted.

The following is a sample of Update message sent when an event condition in the metadata is updated.

```
<Core xmlns=" Namespace of PUCC Core Protocol" >
  <MsgType>Request</MsgType>
  <MsgID>12345. 2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
    <Update xmlns=" Namespace of PUCC Device Discovery and Service Invocation Protocol" >
      <TargetDevice>TargetDeviceID</TargetDevice>
      <EventConditionList>
        <EventCondition type="update" id="event2">
          <ConditionExpression>Temperature<math><lt;=5</lt></math></ConditionExpression>
          <NotificationVariableList>
            <StateVariable name="Temperature"/>
            <StateVariable name="WaterVolume"/>
          </NotificationVariableList>
        </EventCondition>
      </EventConditionList>
    </Update>
  </MsgBody>
</Core>
```

**Figure 6.6.1-3.** Update message sent when an event condition in the metadata is updated.

### 6.6.2. UpdateResponse Message

The following is definition of fields (Parameter) in UpdateResponse message.

**Table6-6-2 : Fields of UpdateResponse Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
UpdateResponse	XML Fragment (Status element, Reason element, Device element, PrimitiveDevice element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
Status	“Success” or “Failure” <string>	-	-	1	Required
Reason	“PermissionDenied”, Reason of the failure <string>	-	-	1	Depend on Status element
Device	XML fragment (Specification element, StateVariableList element, ServiceList element, PrimitiveDeviceList element, EventConditionList element. For details, refer to the PUCC Device and Service Metadata Template Specification.) Updated device metadata.	-	-	1	Optional
		type	URI representing device type. <anyURI>	1	Required
		id	Global unique ID for this device. Should be < 32 characters. <string>	1	Required
		name	Short user-friendly name. Should be < 64 characters. <string>	1	Required
PrimitiveDevi	XML fragment	-	-	1	Optional

***PUCC Device Discovery and Service Invocation Protocol***

ce	(Specification element, StateVariableList element, ServiceList element, PrimitiveDeviceList element, EventConditionList element. For details, refer to the PUCC Device and Service Metadata Template Specification.) Updated primitive device metadata .	type	URI representing device type. <anyURI>	1	Required
		id	Global unique ID for this device. Should be < 32 characters. <string>	1	Required
		name	Short user-friendly name. Should be < 64 characters. <string>	1	Required

**6.6.2.1. UpdateResponse**

This fragment is defined for update report to the update requestor.

**6.6.2.2. Status**

This element returns the status of the response message. The value is “Success” or “Failure”.

**6.6.2.3. Reason**

This element returns the reason of the failure. If the Status element value is “Failure”, this element is required. Don’t set this element, if it is not so.

**6.6.2.4. Device**

The Device element is set to the updated metadata. Even when any of the event condition update requested in the Update message fails and the Status is set to "Failure," this element is set to the updated metadata if any of the event conditions is successfully updated.

**6.6.2.5. PrmitiveDevice**

The PrimitiveDevice element is set to the updated primitive device metadata. Even when any update of multiple event conditions requested in the Update message fails and the Status is set to "Failure," this element is set to the updated metadata if any of the event conditions is successfully updated.

## 6.7. EventSolicited Method

This method is an Advertise-type message used to solicit events accumulated in the device.

If the event subscriber sets the NotifyType attribute of the Subscriber message to “solicited” when it subscribes to events, the device accumulates the events that occur. The event subscriber sends an EventSolicited message to the device in order to obtain the events accumulated in the device. When receiving an EventSolicited message, the device returns a Notify message containing the accumulated events to the event subscriber. For a sequence used to obtain accumulated events, refer to Appendix E.

### 6.7.1. EventSolicited Message

The following is a definition of the fields (Parameter) in the eventsolicited message.

**Table6-7-1: Fields of EventSolicited Message**

Element name	Element Value <datatype> = XML Schema datatype	Attribute name (if present)	Attribute Value <datatype> = XML Schema datatype	Occurrence	Status
EventSolicited	XML Fragment (TargetDevice element, SubscriptionID element, Period element, LeaveEvent element)	-	-	1	Required
		xmlns	Name space identifier (URI) of PUCC Device Discovery and Service Invocation Protocol <anyURI>	1	Required
TargetDevice	Device ID for target device or target primitive device <string>	-	-	1	Required
SubscriptionID	<string>	-	-	1	Required
Period	XML Fragment (Start element , End element)	-	-	1	Optional
Start	<dateTime>	-	-	1	Required
End	<dateTime>	-	-	1	Required
LeaveEvent	XML fragment (StateVariable	-	-	1	Optional



	element)				
--	----------	--	--	--	--

**6.7.1.1. EventSolicited**

This fragment is defined for a message sent from the event subscriber to the device in order to solicit events.

**6.7.1.2. TargetDevice**

This fragment declares the device ID for target device or target primitive device. Device ID is assigned in the target device or primitive device metadata.

**6.7.1.3. SubscriptionID**

This fragment declares the ID for subscription.

**6.7.1.4. Period**

The period during which the events are obtained. When this element is not set, all events accumulated in the device are solicited.

**6.7.1.5. Start**

The time to start obtaining events.

**6.7.1.6. End**

The time to stop obtaining events.

**6.7.1.7. LeaveEvent**

This element sets whether the events are to be left or not in the device after they are obtained. This is set to “yes” or “no.” The default value is “no.”

Whichever value is set for this element, the device deletes all of its accumulated events when the event registration is released by an Unsubscribe message.

## Appendix A. Version History

(This appendix does not form an integral part of this specification.)

Document number	Date	Note
PUCC Device Discovery and Service Invocation Protocol	30 Sep. 2007	Version 1.0
PUCC Device Discovery and Service Invocation Protocol	30 Nov. 2009	Version 2.0
PUCC Device Discovery and Service Invocation Protocol	22 March. 2012	Version 3.0

## **Appendix B: Namespace Definitions**

Namespaces are defined with URI.

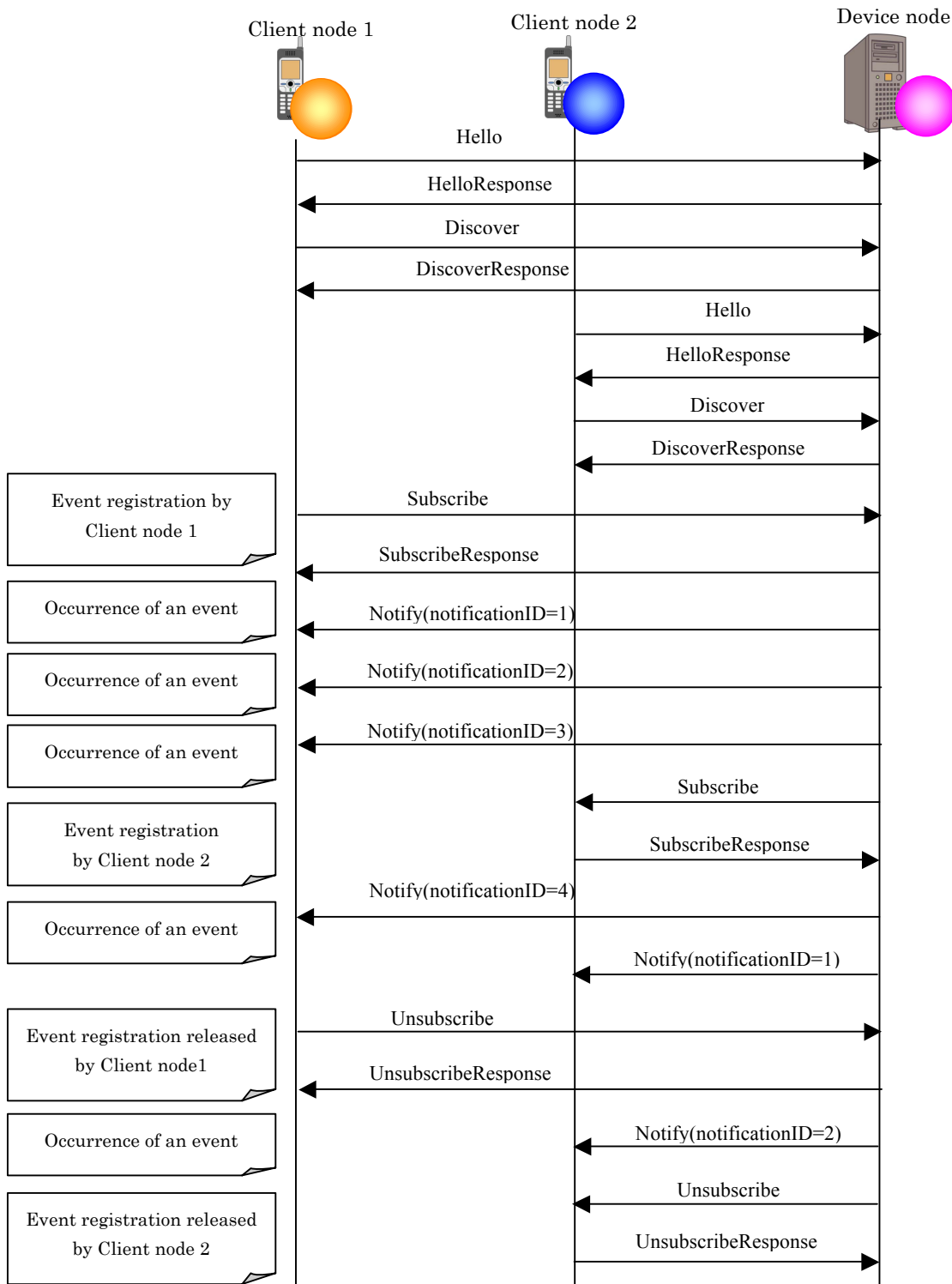
*<http://www.pucc.jp/2012/03/ddsi>*

xmlns attribute of MsgBody element uses it in Pucc Device Discovery and Service Invocation Protocol.

## **Appendix C: Notification of Event Difference by Notify Message**

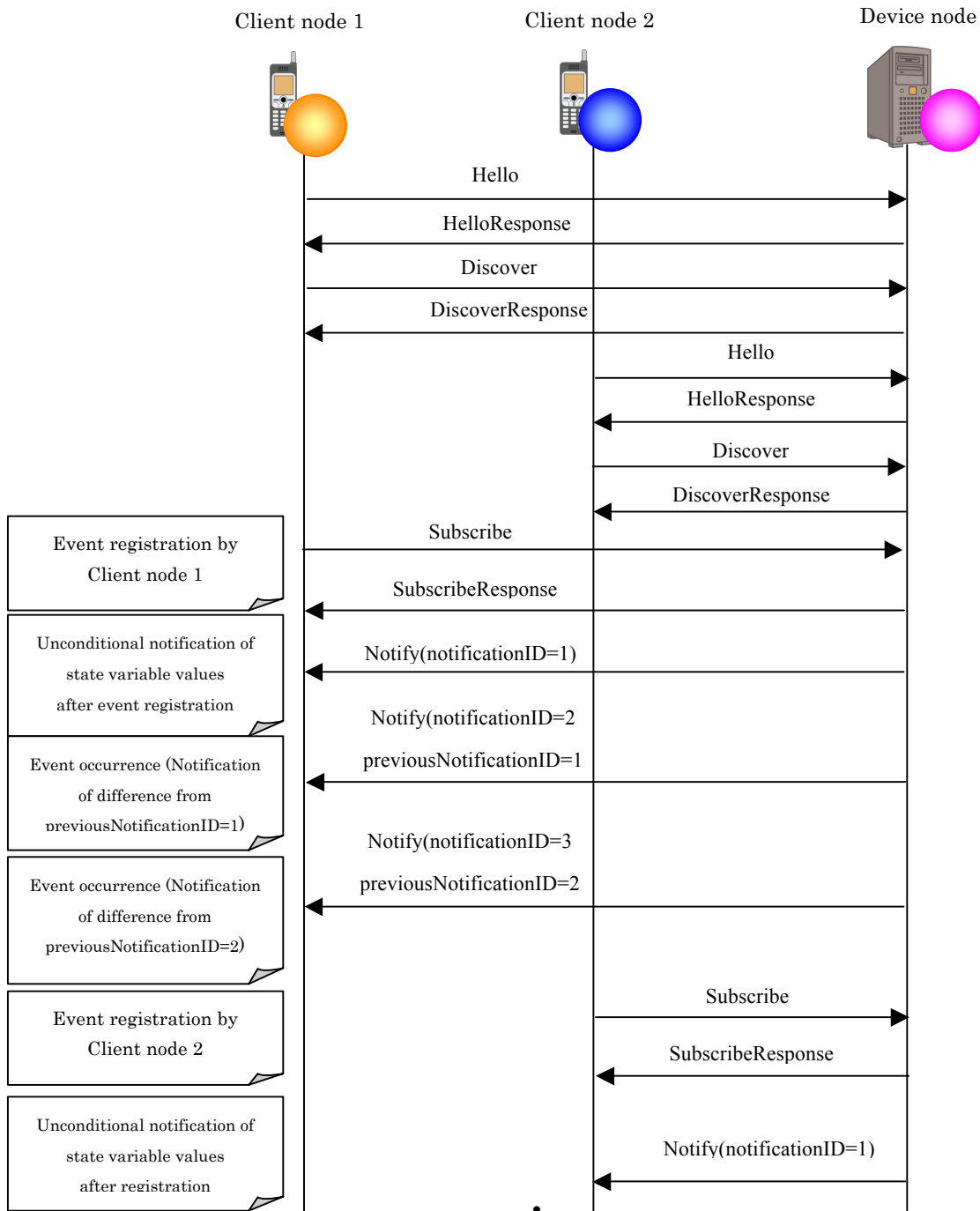
When the device receives a Subscribe message, it starts monitoring events in consideration of the condition of the target event. When an event that satisfies the event generation condition occurs, the device sends a Notify message containing the Event element set to the notificationID. When the difference of the state variables are notified of, the Event element is set to the notification ID and the previousNotificationID indicating from which notification ID the difference is demonstrated. Since more than one Event elements can be set in the EventList of the Notify message, the difference of the notification IDs of the Event elements and the difference between the notificationID and the previousNotificationID shown in a Notify message do not necessarily be one (1).

Figure C-1 below shows a sequence used in a normal event notification (when the notificationType attribute of the StateVariable element in the event condition is set to “normal.”)



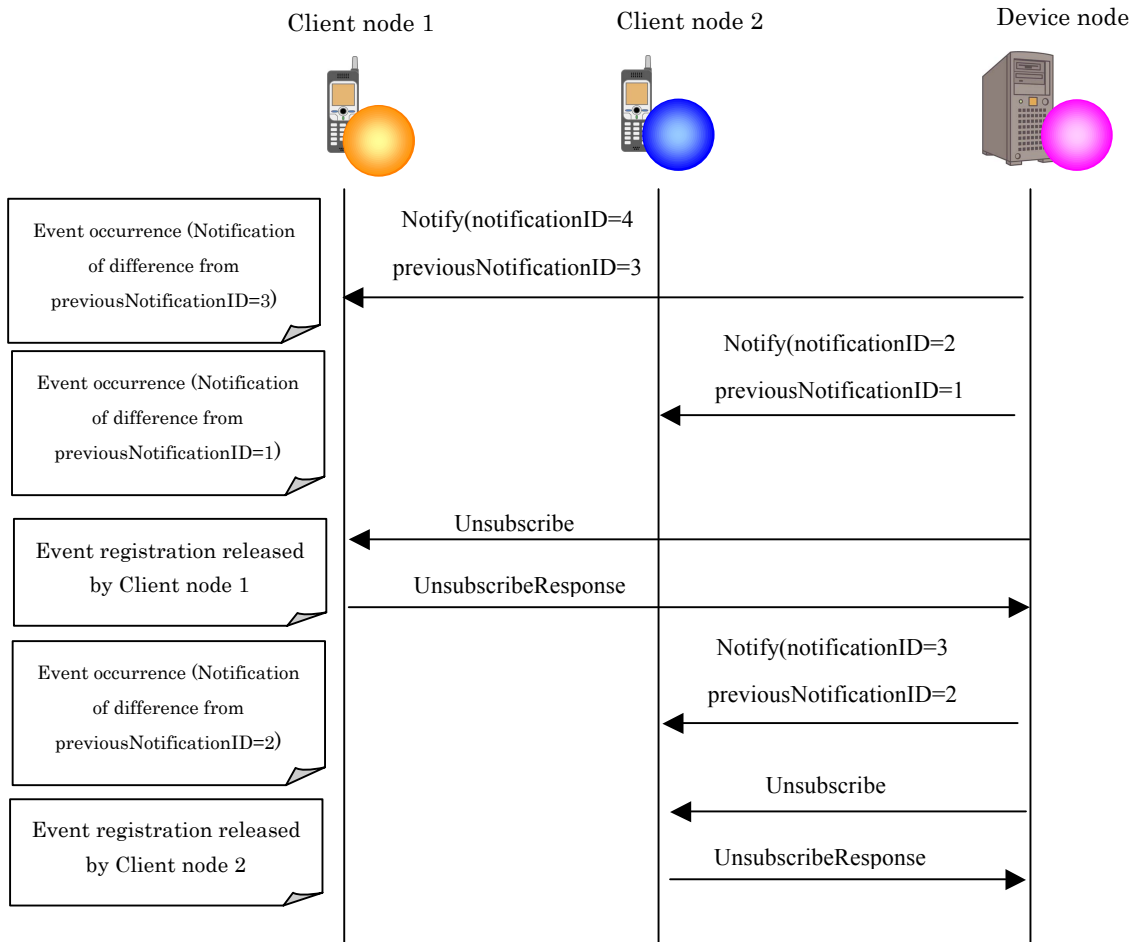
**Figure C-1. Normal Event Notification**

Figure C-2 and Figure C-3 below show a sequence used in an event notification that notifies of event differences (when the notificationType attribute of the StateVariable element in the event condition is set to “difference” ).



**Figure C-2. Event Notification of State Variable Difference(1)**

**PUCC Device Discovery and Service Invocation Protocol**

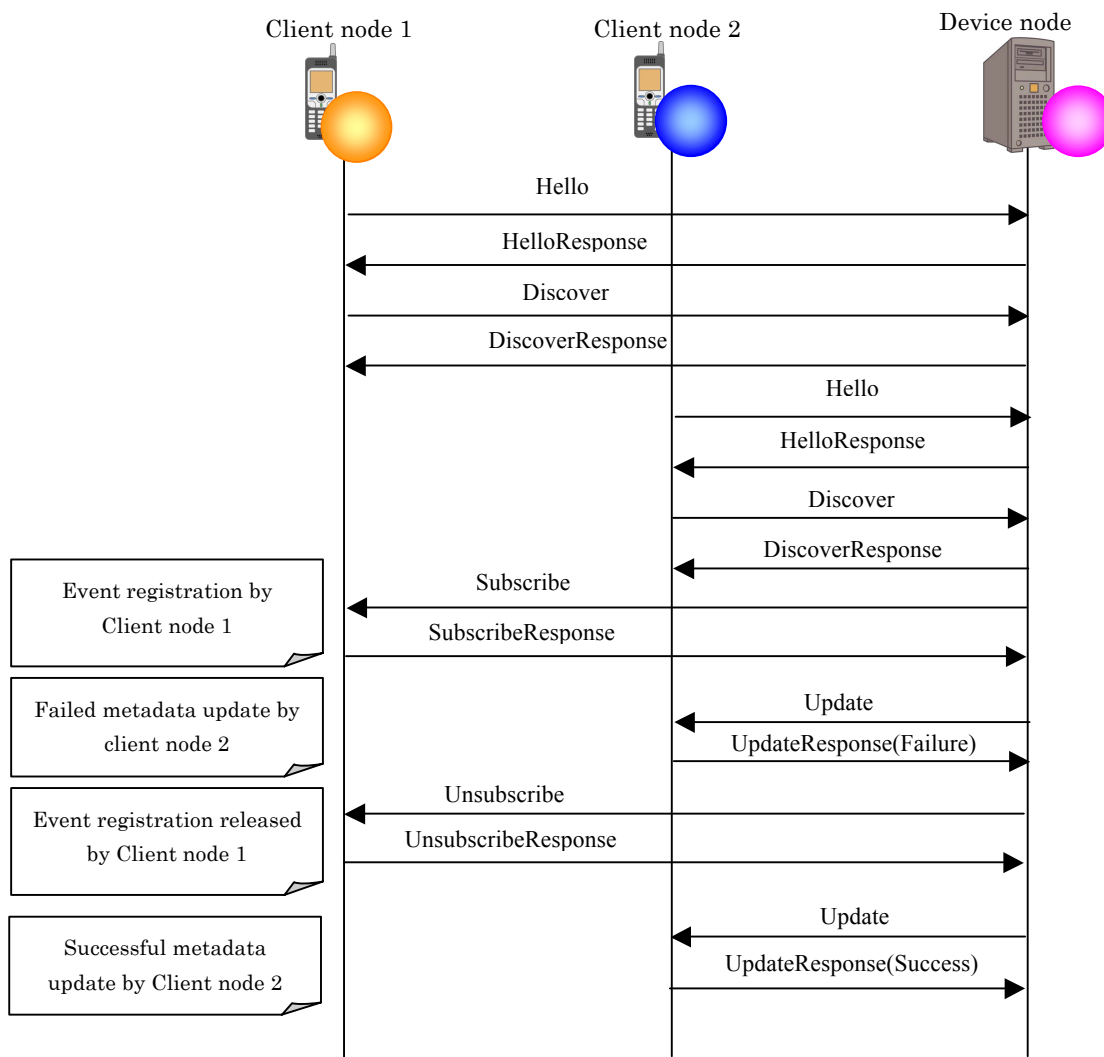


**Figure C-3. Event Notification of State Variable Difference(2)**

## Appendix D: Exclusive Control of the Device

### ① Metadata update of Device being Subscribed

Any change of an event generation condition in the device metadata being Subscribed may cause the node which has Subscribed to be notified of an unexpected event. In order to prevent such a situation, the device returns an UpdateResponse(Failure) message when it receives an Update message while it is subscribed.



**Figure D-1. Metadata Update of Device being Subscribed**



**PUCC Device Discovery and Service Invocation Protocol**

②Event registration to the device which cannot be Subscribed to by multiple event subscribers

If the device which cannot allow more than one event subscribers to Subscribe to it due to its characteristic features is already Subscribed, it does not accept any more Subscribe from another event subscriber.

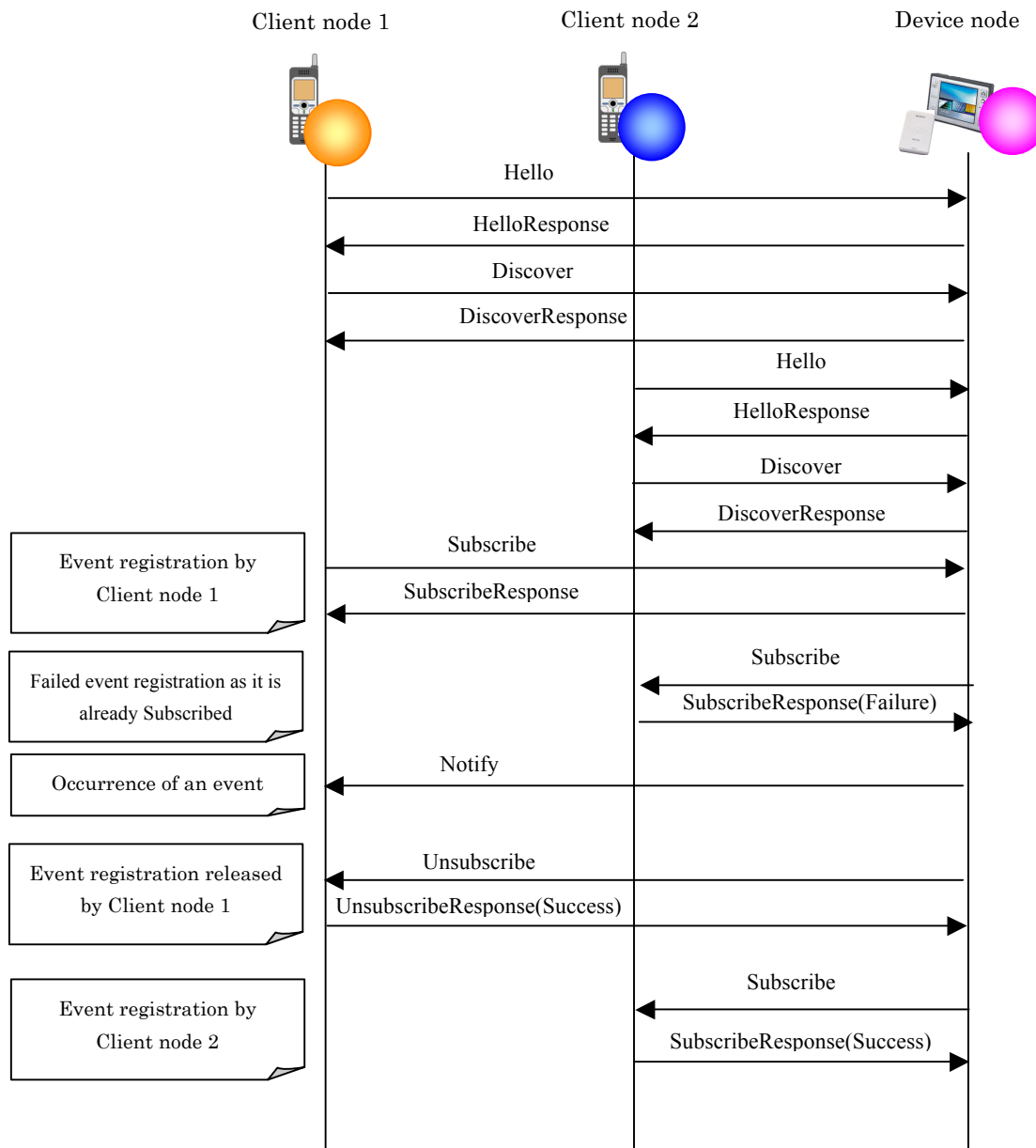
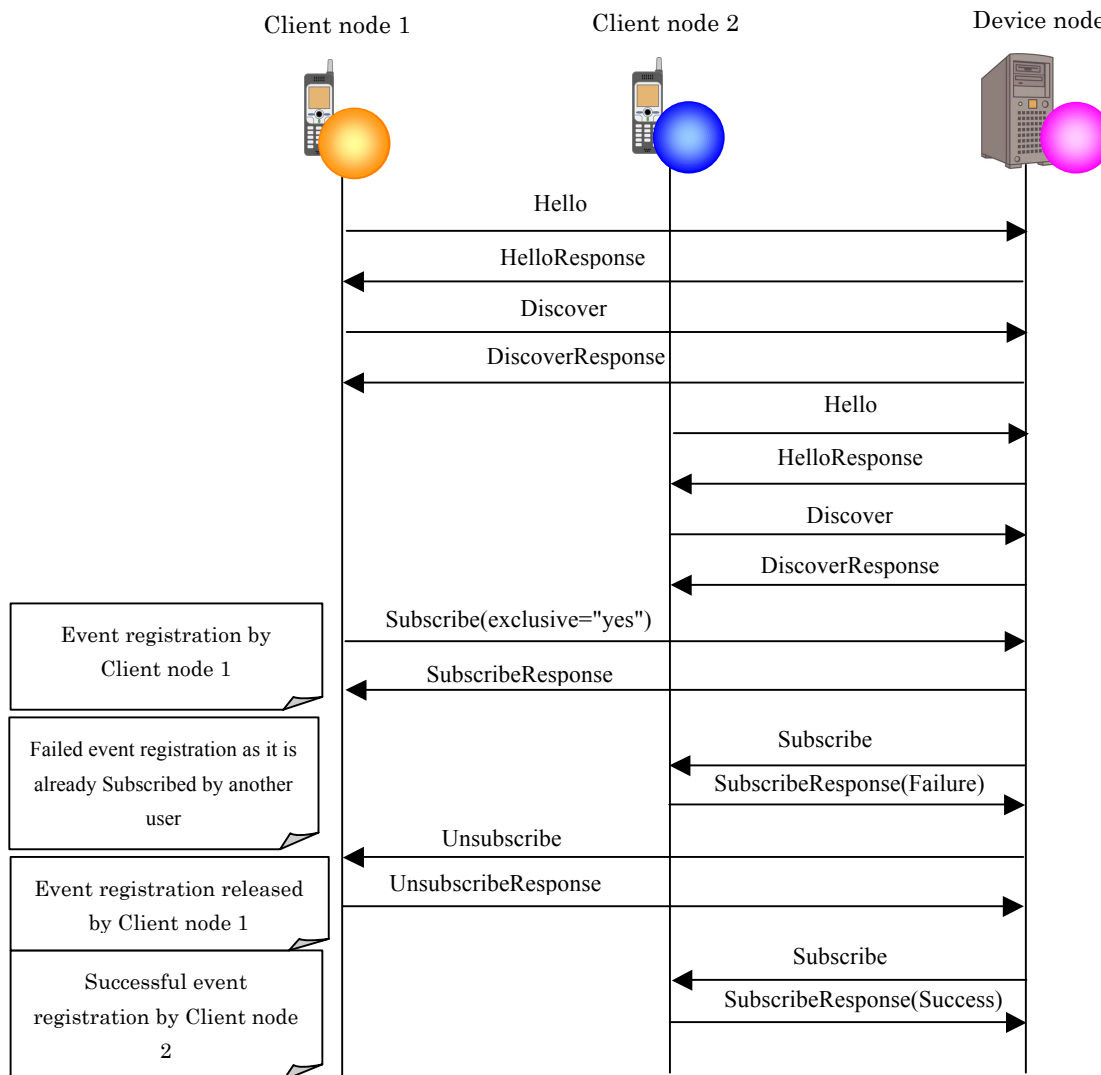


Figure D-2. Event Registration with the Device which cannot accept Subscribe from more than one event subscribers

③ Designation of exclusive control over the device by the event subscriber

If the event subscriber wants to exclude event Subscription from any other user to the device which can be Subscribed to by more than one event subscribers, the event subscriber designates the execution of exclusive control using the exclusive attribute.

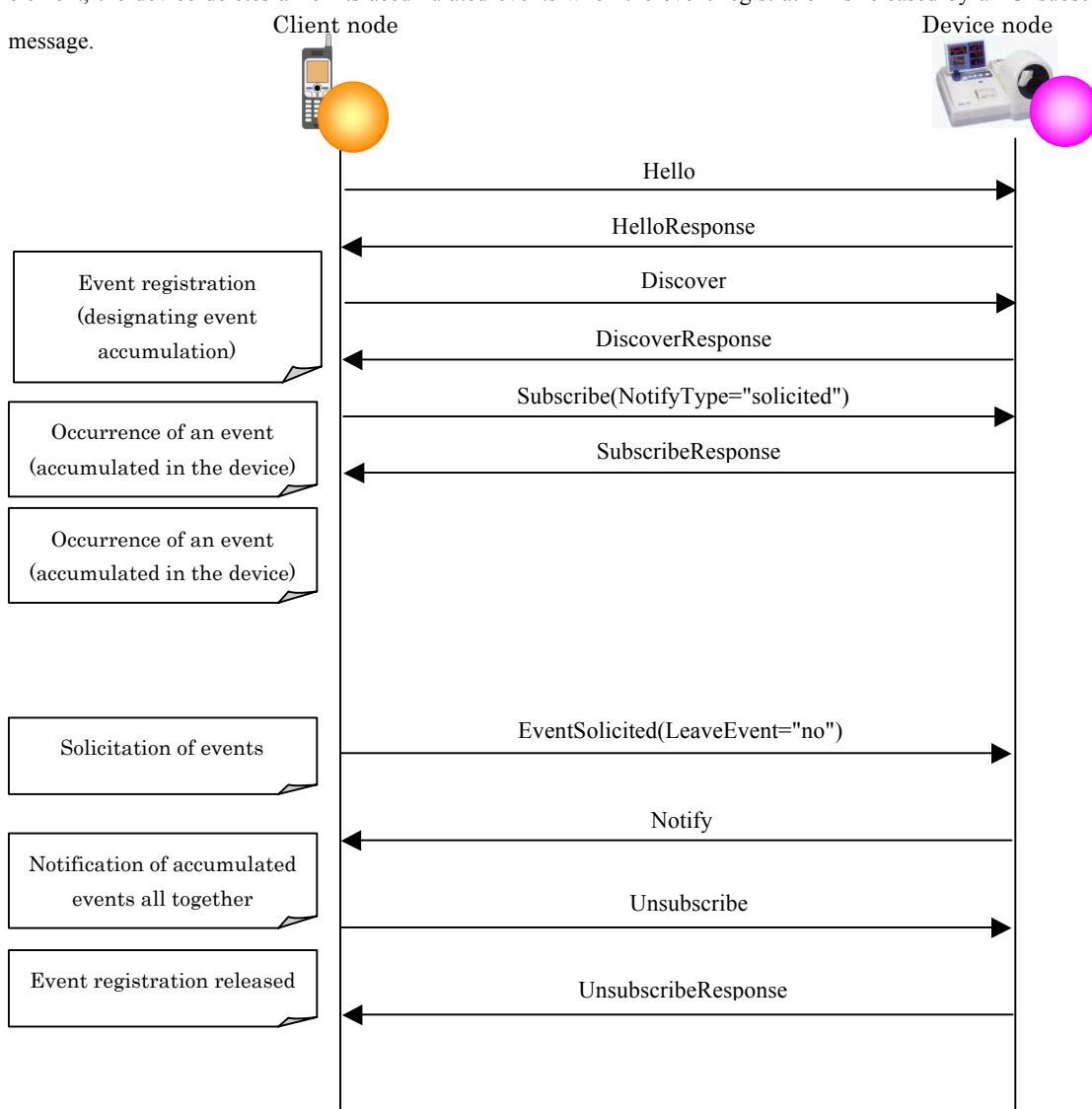


**Figure D-3. Exclusive Control Designated by Event Subscriber**

### Appendix E: Notification of Events Accumulated in the Device

① To delete the events accumulated in the device after they are obtained.

The event subscriber sets the NotifyType element of the Subscribe message to “solicited” when subscribing to events. Then the device accumulates the events that occur until it receives an EventSolicited message from the event subscriber. When receiving an EventSolicited message, the device sends the events accumulated using the Notify message. When the LeaveEvent element in the EventSolicited message is set to “no,” the device deletes the accumulated events after it notifies the event subscriber of the events. Whichever value is set for the LeaveEvent element, the device deletes all of its accumulated events when the event registration is released by an Unsubscribe message.



**Figure E-1. Notification of Events Accumulated in the Device**

## **Appendix F: PUCC Event Monitoring after PUCC connection terminated**

Once the device had invoked event monitoring by Subscribe method, the event monitoring SHOULD be continued until monitoring release is clearly instructed.

When PUCC connection termination occurred between a event subscription source node and the device node after event subscription by Subscribe method, the device SHOULD store all occurred event information with continuing event monitoring.

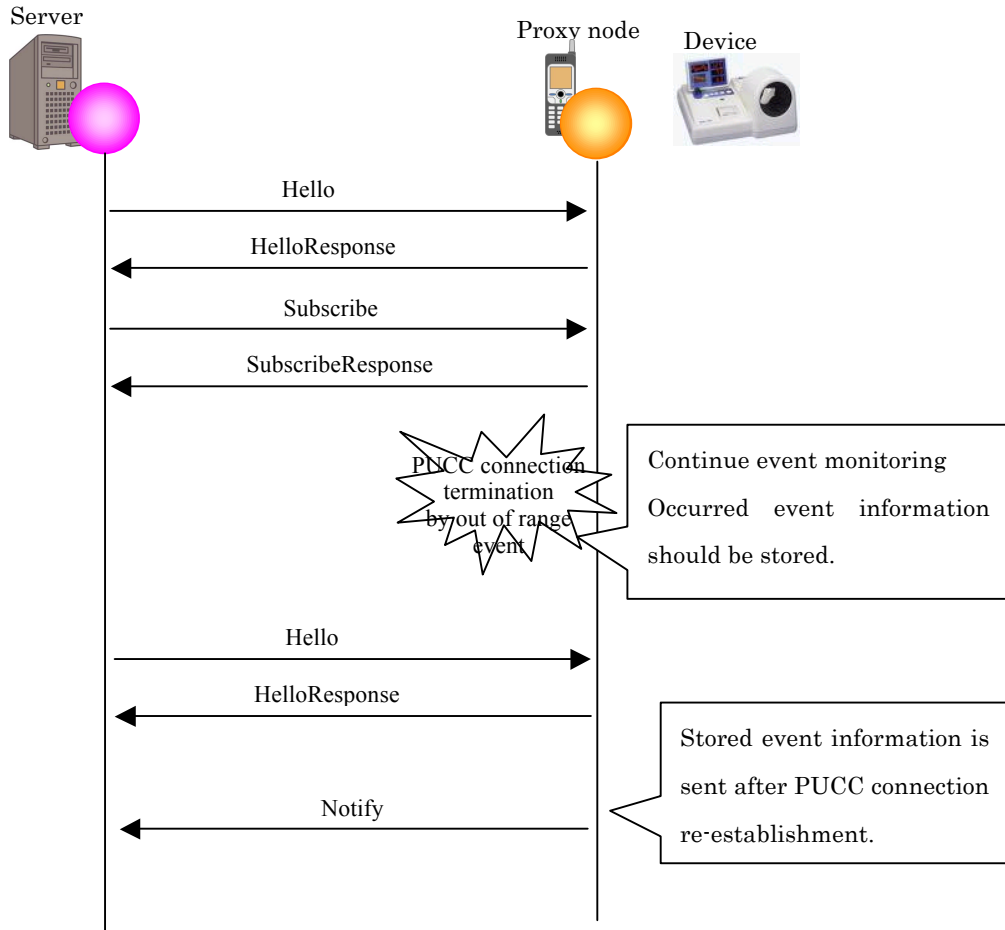
When PUCC connection is re-established, event subscription source node MAY NOT need to request event subscription by Subscribe method to the device that has already subscribed.

After PUCC connection re-established between a event subscription source node and the device, the device SHOULD send all occurred event information stored during PUCC connection termination.

The following is a use case of measured data transfer from a device to a server using a mobile phone as the device proxy.

PUCC connection may need to terminate in the case such that the mobile phone moves out of range after Subscribe method was sent from the server to the mobile phone. In this case, the mobile phone is required to continue event monitoring and store all occurred events as the device proxy.

After PUCC connection is re-established between the mobile phone node and the server node, stored event information in the mobile phone as the device proxy SHOULD be sent to the server.



**Figure F-1. Event monitoring after PUCC connection terminated**

**Appendix G: Access Control Operation Response by ACL (Access Control List)**

The followings show error response procedure when unauthorized operation occurred defined in ACL of a device metadata.

Pucc device metadata ACL allows to describe the access authorization to the following Request/Response type method,

Discover

Invoke

Subscribe

Update.

Discover method returns Response only for the matched Request to the condition, and no response (Response is NOT sent) to the unmatched Request.

Three methods other than Search type return Response message including the success flag and result to each Request.

No	Type	Method	Response for the authorized operation	Response for the unauthorized operation
1	Search type	Discover	Response when the condition is satisfied No response when the condition is NOT satisfied	No response. The same behavior to unmatched condition case.
2	Operation Command type	Invoke	Response with success flag and result information.	Command execution SHOULD be failed and SHOULD send "PermissionDenied" as a response message
3		Subscribe	Subscribe required events and respond with a operation success flag	Command execution SHOULD be failed and SHOULD send "PermissionDenied" as a response message
4		Update	Respond with a operation success flag	Command execution SHOULD be failed and SHOULD send "PermissionDenied" as a response message

**Figure G-1. ACL control and response with Pucc Device Discovery and Service Invocation Protocol**