INTERNATIONAL STANDARD

ISO/IEC 29341-26-15

First edition
2017-09

Information technology — UPnP Device Architecture —

Part 26-15: Telephony device control protocol — Level 2 — Calendar service

Technologies de l'information — Architecture de dispositif UPnP —

Partie 26-15: Protocole de contrôle de dispositif de téléphonie — Niveau 2 — Service de calendrier

Reference number
ISO/IEC 29341-26-15:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see http://www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of Standard, the meaning of the ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword – Supplementary information

ISO/IEC 29341-26-15 was prepared by UPnP Forum and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of ISO/IEC 29341 series, under the general title *Information technology — UPnP Device Architecture*, can be found on the ISO web site.

Introduction

ISO and IEC draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights. The holders of -these patent rights have assured ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Huawei Technologies Co., Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Huawei Technologies Co., Ltd.
Administration Building, Bantian Longgang District
Shenzhen – China 518129

Qualcomm Incorporated has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA – USA 92121

Telecom Italia S.p.A.has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Telecom Italia S.p.A.
Via Reiss Romoli, 274
Turin - Italy 10148

Cisco Systems informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA – USA 95134

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Original UPnP Document

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation and later by UPnP Forum. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

UPnP Document Title	ISO/IEC 29341 Part
UPnP Device Architecture 1.0	ISO/IEC 29341-1:2008
UPnP Device Architecture Version 1.0	ISO/IEC 29341-1:2011
UPnP Device Architecture 1.1	ISO/IEC 29341-1-1:2011
UPnP Device Architecture 2.0	ISO/IEC 29341-1-2
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2008
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2011
UPnP AVTransport:1 Service	ISO/IEC 29341-3-10
UPnP ConnectionManager:1 Service	ISO/IEC 29341-3-11
UPnP ContentDirectory:1 Service	ISO/IEC 29341-3-12
UPnP RenderingControl:1 Service	ISO/IEC 29341-3-13
UPnP MediaRenderer:1 Device	ISO/IEC 29341-3-2
UPnP MediaRenderer:2 Device	ISO/IEC 29341-3-2:2011
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2008
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2011
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2008
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2011
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2008
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2011
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP ScheduledRecording:2	ISO/IEC 29341-4-14:2011
UPnP MediaRenderer:2 Device	ISO/IEC 29341-4-2
UPnP MediaServer:2 Device	ISO/IEC 29341-4-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2008
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2011
UPnP DigitalSecurityCamera:1 Device	ISO/IEC 29341-5-1
UPnP DigitalSecurityCameraMotionImage:1 Service	ISO/IEC 29341-5-10
UPnP DigitalSecurityCameraSettings:1 Service	ISO/IEC 29341-5-11
UPnP DigitalSecurityCameraStillImage:1 Service	ISO/IEC 29341-5-12
UPnP HVAC_System:1 Device	ISO/IEC 29341-6-1
UPnP ControlValve:1 Service	ISO/IEC 29341-6-10
UPnP HVAC_FanOperatingMode:1 Service	ISO/IEC 29341-6-11
UPnP FanSpeed:1 Service	ISO/IEC 29341-6-12
UPnP HouseStatus:1 Service	ISO/IEC 29341-6-13
UPnP HVAC_SetpointSchedule:1 Service	ISO/IEC 29341-6-14
UPnP TemperatureSensor:1 Service	ISO/IEC 29341-6-15
UPnP TemperatureSetpoint:1 Service	ISO/IEC 29341-6-16

UPnP HVAC_UserOperatingMode:1 Service	ISO/IEC 29341-6-17
UPnP HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2
UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
UPnP LANHostConfigManagement:1 Service	ISO/IEC 29341-8-10
UPnP Layer3Forwarding:1 Service	ISO/IEC 29341-8-11
UPnP LinkAuthentication:1 Service	ISO/IEC 29341-8-12
UPnP RadiusClient:1 Service	ISO/IEC 29341-8-13
UPnP WANCableLinkConfig:1 Service	ISO/IEC 29341-8-14
UPnP WANCommonInterfaceConfig:1 Service	ISO/IEC 29341-8-15
UPnP WANDSLLinkConfig:1 Service	ISO/IEC 29341-8-16
UPnP WANEthernetLinkConfig:1 Service	ISO/IEC 29341-8-17
UPnP WANIPConnection:1 Service	ISO/IEC 29341-8-18
UPnP WANPOTSLinkConfig:1 Service	ISO/IEC 29341-8-19
UPnP LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
UPnP WANDevice:1 Device	ISO/IEC 29341-8-3
UPnP WANConnectionDevice:1 Device	ISO/IEC 29341-8-4
UPnP WLANAccessPointDevice:1 Device	ISO/IEC 29341-8-5
UPnP Printer:1 Device	ISO/IEC 29341-9-1
UPnP ExternalActivity:1 Service	ISO/IEC 29341-9-10
UPnP Feeder:1.0 Service	ISO/IEC 29341-9-11
UPnP PrintBasic:1 Service	ISO/IEC 29341-9-12
UPnP Scan:1 Service	ISO/IEC 29341-9-13
UPnP Scanner:1.0 Device	ISO/IEC 29341-9-2
UPnP QoS Architecture:1.0	ISO/IEC 29341-10-1
UPnP QosDevice:1 Service	ISO/IEC 29341-10-10
UPnP QosManager:1 Service	ISO/IEC 29341-10-11
UPnP QosPolicyHolder:1 Service	ISO/IEC 29341-10-12
UPnP QoS Architecture:2	ISO/IEC 29341-11-1
UPnP QosDevice:2 Service	ISO/IEC 29341-11-10
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11
UPnP ContentDirectory:3 Service	ISO/IEC 29341-14-12:2011
UPnP MediaServer:3 Device	ISO/IEC 29341-14-3:2011

UPnP ContentSync:1	ISO/IEC 29341-15-10:2011
UPnP Low Power Architecture:1	ISO/IEC 29341-16-1:2011
UPnP LowPowerProxy:1 Service	ISO/IEC 29341-16-10:2011
UPnP LowPowerDevice:1 Service	ISO/IEC 29341-16-11:2011
UPnP QoS Architecture:3	ISO/IEC 29341-17-1:2011
UPnP QosDevice:3 Service	ISO/IEC 29341-17-10:2011
UPnP QosManager:3 Service	ISO/IEC 29341-17-11:2011
UPnP QosPolicyHolder:3 Service	ISO/IEC 29341-17-12:2011
UPnP QosDevice:3 Addendum	ISO/IEC 29341-17-13:2011
UPnP RemoteAccessArchitecture:1	ISO/IEC 29341-18-1:2011
UPnP InboundConnectionConfig:1 Service	ISO/IEC 29341-18-10:2011
UPnP RADAConfig:1 Service	ISO/IEC 29341-18-11:2011
UPnP RADASync:1 Service	ISO/IEC 29341-18-12:2011
UPnP RATAConfig:1 Service	ISO/IEC 29341-18-13:2011
UPnP RAClient:1 Device	ISO/IEC 29341-18-2:2011
UPnP RAServer:1 Device	ISO/IEC 29341-18-3:2011
UPnP RADiscoveryAgent:1 Device	ISO/IEC 29341-18-4:2011
UPnP SolarProtectionBlind:1 Device	ISO/IEC 29341-19-1:2011
UPnP TwoWayMotionMotor:1 Service	ISO/IEC 29341-19-10:2011
UPnP AV Architecture:2	ISO/IEC 29341-20-1
UPnP AVTransport:3 Service	ISO/IEC 29341-20-10
UPnP ConnectionManager:3 Service	ISO/IEC 29341-20-11
UPnP ContentDirectory:4 Device	ISO/IEC 29341-20-12
UPnP RenderingControl:3 Service	ISO/IEC 29341-20-13
UPnP ScheduledRecording:2 Service	ISO/IEC 29341-20-14
UPnP MediaRenderer:3 Service	ISO/IEC 29341-20-2
UPnP MediaServer:4 Device	ISO/IEC 29341-20-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-20-4
UPnP InternetGatewayDevice:2 Device	ISO/IEC 29341-24-1
UPnP WANIPConnection:2 Service	ISO/IEC 29341-24-10
UPnP WANIPv6FirewallControl:1 Service	ISO/IEC 29341-24-11
UPnP WANConnectionDevice:2 Service	ISO/IEC 29341-24-2
UPnP WANDevice:2 Device	ISO/IEC 29341-24-3
UPnP Telephony Architecture:2	ISO/IEC 29341-26-1
UPnP CallManagement:2 Service	ISO/IEC 29341-26-10
UPnP MediaManagement:2 Service	ISO/IEC 29341-26-11
UPnP Messaging:2 Service	ISO/IEC 29341-26-12
UPnP PhoneManagement:2 Service	ISO/IEC 29341-26-13
UPnP AddressBook:1 Service	ISO/IEC 29341-26-14
UPnP Calendar:1 Service	ISO/IEC 29341-26-15
UPnP Presense:1 Service	ISO/IEC 29341-26-16
UPnP TelephonyClient:2 Device	ISO/IEC 29341-26-2
UPnP TelephonyServer:2 Device	ISO/IEC 29341-26-3
UPnP Friendly Info Update:1 Service	ISO/IEC 29341-27-1
UPnP MultiScreen MultiScreen Architecture:1	ISO/IEC 29341-28-1
UPnP MultiScreen Application Management:1 Service	ISO/IEC 29341-28-10

UPnP MultiScreen Screen:1 Device	ISO/IEC 29341-28-2
UPnP MultiScreen Application Management:2 Service	ISO/IEC 29341-29-10
UPnP MultiScreen Screen:2 Device	ISO/IEC 29341-29-2
UPnP IoT Management and Control Architecture Overview:1	ISO/IEC 29341-30-1
UPnP DataStore:1 Service	ISO/IEC 29341-30-10
UPnP IoT Management and Control Data Model:1 Service	ISO/IEC 29341-30-11
UPnP IoT Management and Control Transport Generic:1 Service	ISO/IEC 29341-30-12
UPnP IoT Management and Control:1 Device	ISO/IEC 29341-30-2
UPnP Energy Management:1 Service	ISO/IEC 29341-31-1

1 Scope

This service definition is compliant with [1]. It defines a service type referred to herein as *Calendar*.

This service provides the following functionalities:

- Manage the family calendar events and share calendar events between the family members.
- Calendar events includes Reminders/Alarms/Free-BusyTime.
- Synchronizing the local calendar events with family calendar.
- Registrating for receiving the events notification on WAN side using any kinds of messaging service mechanism.
- Memo handling : Posting simple memos, retrieveing memos.
- Registrating for receiving the memo notifications on the WAN side using any kind of messaging service.

This service does not provide the following functionalities:

- Interaction between Telephony Server and network side calendar service

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] – UPnP Device Architecture, version 1.0, UPnP Forum, October 15, 2008. Available at: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20081015.pdf. Latest version available at: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf.

[2] – Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 21, 2000. Available at: ISO 8601:2000.

[3] – IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, 1997. Available at: http://www.faqs.org/rfcs/rfc2119.html.

[4] – IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, MIT/LCS, R. Fielding, U.C. Irvine, L. Masinter, Xerox Corporation, August 1998. Available at: http://www.ietf.org/rfc/rfc2396.txt.

[5] – IETF RFC 3339, Date and Time on the Internet: Timestamps, G. Klyne, Clearswift Corporation, C. Newman, Sun Microsystems, July 2002. Available at: http://www.ietf.org/rfc/rfc3339.txt.

[6] – IETF RFC 3966, The tel URI for Telephone Numbers, H. Schulzrinne, Columbia University, December 2004. Available at: http://www.ietf.org/rfc/rfc3966.txt.

[7] – IETF RFC 5545, Internet Calendaring and Scheduling Core Object Specification (iCalendar), B. Desruisseaux, Sept 2009. Available at: http://tools.ietf.org/html/rfc5545.

[8] – Extensible Markup Language (XML) 1.0 (Third Edition), François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004. Available at: http://www.w3.org/TR/2004/REC-xml-20040204.

[9] – XML Schema Part 2: Data Types, Second Edition, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028.

[10] – *PhoneManagement:2*, UPnP Forum, December 10, 2012. Available at: http://upnp.org/specs/phone/UPnP-phone-PhoneManagement-v2-Service-20121210.pdf. Latest version available at: http://upnp.org/specs/phone/UPnP-phone-PhoneManagement-Service.pdf.

3 Terms, definitions, symbols and abbreviated terms

For the purposes of this document, the terms and definitions given in [1] and the following apply.

3.1 Provisioning terms

3.1.1

conditionally allowed
CA
The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is allowed, otherwise it is not allowed.

3.1.2

conditionally required
CR
The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is required, otherwise it is not allowed.

3.1.3

not allowed
The definition or behavior is prohibited by this specification. Opposite of required.

3.2 Symbols

3.2.1

::
signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

3.3 Abbreviated terms

3.3.1

GUI
Graphical User Interface

3.3.2

ID
Identifier

3.3.3

TC
Telephony Client

3.3.4

TelCP
Telephony Control Point

3.3.5

TS
Telephony Server

3.3.6

VoIP
Voice over IP

3.3.7

WAN
Wide Area Network

4 Notations and conventions

4.1 Text conventions

- Strings that are to be taken literally are enclosed in “double quotes”.
- Words that are emphasized are printed in *italic*.
- Keywords that are defined by the UPnP Working Committee are printed using the *forum* character style.
- Keywords that are defined by [1] are printed using the **arch** character style.

4.1.1 Data Types

This specification uses data type definitions from two different sources. Data types from [1] are used to define state variable and action argument data types [1]. The XML Schema namespace is used to define property data types [9].

For Boolean data types from [1], it is strongly recommended to use the value “**0**” for false, and the value “**1**” for true. The values “**true**”, “**yes**”, “**false**”, or “**no**” may also be used but are not recommended. The values “**yes**” and “**no**” are deprecated and shall not be sent out by devices but shall be accepted on input.

For XML Schema defined Boolean data types, it is strongly recommended to use the value “*0*” for false, and the value “*1*” for true. The values “*true*”, “*yes*”, “*false*”, or “*no*” may also be used but are not recommended. The values “*yes*” and “*no*” are deprecated and shall not be sent out by devices but shall be accepted on input.

4.2 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation shall follow the naming conventions and XML rules as specified in [1], 2.5, “Description: Non-standard vendor extensions”.

5 Service Modeling Definitions

5.1 Service Type

The following service type identifies a service that is compliant with this specification:

urn:schemas-upnp-org:service:*Calendar:1*

Calendar service is used herein to refer to this service type.

5.2 *Calendar* Service Architecture

The *Calendar* service provides in the UPnP network the overall set of calendar functionalities of a phone (Smart Phone, IP Phone, VOIP Gateways, etc), as a role of a TS.

Accroding to the Telephony architecture in [11], the *Calendar* service is included in the TS device. More than one *Calendar* service can coexist in the same UPnP network, it’s up to the TelCP to manage multiple *Calendar* service.

The *Calendar* service provides a TelCP with the following features:

- Managing the family calendar: managing reminders/ToDo list/Alarms/General Notes.
- Family user can register for events when user is outside of the home via any messaging mechanism.
- Synching the events with the network calendar service if available.

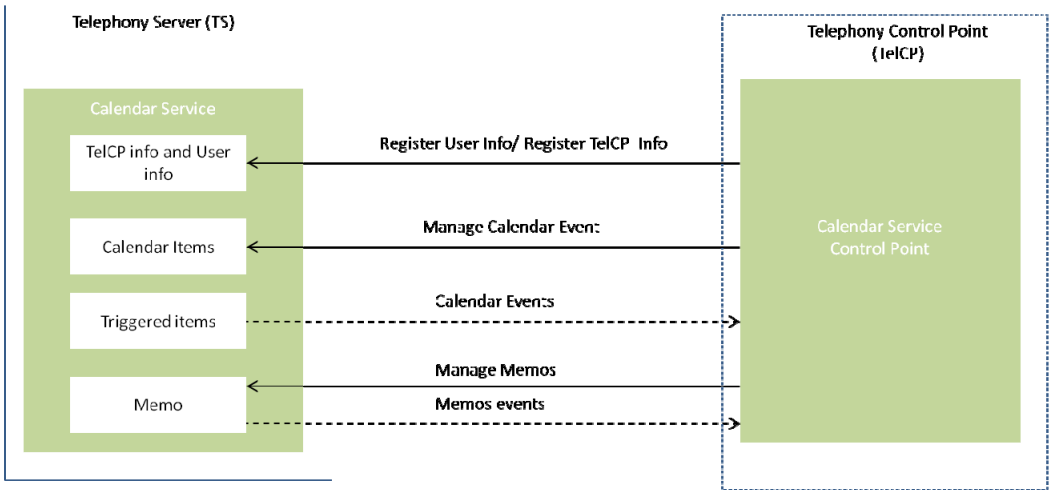

Figure 1 — *Calendar* Service Architecture

The UPnP device that implements the *Calendar* service is capable of accessing features provided by network calendar system through some telecommunication means (e.g. internet based calendar service or network calendar etc.). The interaction with such network based calendar service is out of the scope of this specification.

It is possible to have a local user interface for the *Calendar* service in UPnP server. The local management of *Calendar* service and related fetaures by the end user should be taken into consideration when implementing the *Calendar* service behavour.

The *Calendar* service consists of three main components.

- Calendar Objects: This entity stores all the calendar events
- Delivery mechanism/TelCP Info: This entity stores the delivery mechanism (e.g., contact information of the user) to deliver the memo or triggered calendar events when a user is outside the home network. The TelCP info is stored to specify which TelCP controls the particular triggered event.
- *TriggeredEvent*: This state variable will be used to notify all triggered event of the calendar for e.g. Alarm Event etc.
- *MemoInfo*: This state variable will be used to notify the new memo.

The calendar objects stores all the calendar event in a XML structure. The *Calendar* service follows the structure in [7] for storing the calendar objects. This allows interoperability with the existing calendar application on the device.

5.3 State Variables

Note: For first-time reader, it may be more insightful to read the theory of operations first and then the action definitions before reading the state variable definitions.

5.3.1 State Variable Overview

Table 1 — State Variables

Variable Name	R/A [a]	Data Type	Reference
CalendarItem	*R*	**String** (XML fragment)	See 5.3.2
TriggeredItem	*R*	**String** (XML fragment)	See 5.3.3
MemoInfo	*R*	**String** (XML fragment)	See 5.3.4
A_ARG_TYPE_TriggeredItems	*A*	**String** (XML fragment)	See 5.3.5
A_ARG_TYPE_CalItems	*R*	**String** (XML fragment)	See 5.3.6
A_ARG_TYPE_ItemIDs	*R*	**String** (XML fragment)	See 5.3.7
A_ARG_TYPE_Memo	*R*	**String**	See 5.3.8
A_ARG_TYPE_MemoID	*R*	**String**	See 5.3.9
A_ARG_TYPE_ItemDeliveryMethod	*A*	**String** (XML fragment)	See 5.3.10
A_ARG_TYPE_MemoInfoList	*A*	**String** (XML fragment)	See 5.3.11
A_ARG_TYPE_TelCPName	*A*	**String**	See 5.3.12
A_ARG_TYPE_TelCPNameList	*A*	**String**	See 5.3.13
A_ARG_TYPE_Expires	*A*	**ui4**	See 5.3.14

[a] *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

5.3.2 *CalendarItem*

This state variable includes information of a calendar item. The state variable is an XML document. The calendar item is stored in the format given by [7]. This state variable is evented out when a new calendar event is set or existing calendar item is updated.

5.3.2.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *CalendarItem* in the XML namespace "urn:schemas-upnp-org:phone:messaging" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

5.3.2.2 Description of fields in the *CalendarItem* structure

```
<?xml version="1.0" encoding="UTF-8"?>
<calendar:calendarItem
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:item Itemid="ID of the item" format="RFC5545" TelCPIDs ="ID of the
TelCPs">
      actual event as per RFC 5545.(textual format)
   </calendar:item>
</calendar:calendarItem>
```

<xml>

Required. Case Sensitive.

<calendarItem>

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas-upnp-org:phone:CalendarService”). This namespace “urn:schemas-upnp-org:phone:CalendarService” defines the following elements and attribute

<item>

Required. includes the individual calendar event(object) stored in the format of [7]. The calendar oject is stored in the texual format and follow the rule specified in [7].

ItemID

Required. includes the unique identifierfor each calendat item stored.

format

Required. specifies the format used to store the calendar item. The value of this attribute shall be “RFC5545”.

TelCPID

Allowed. specifies the TelCP who is associated with the calendar item.

5.3.3 *TriggeredItem*

This state variable contains the identifier of the triggered calendar items. When a calendar item is triggered (e.g. Birthday reminder generated) the particular calendar item will be evented to the TelCP(s).

The *Calendar* service includes triggered calendar item ID in this state variable. So any new triggered events will be evented to all subscribed TelCP. This state variable can be cleared

periodically to clear off the already evented calendar items but this will be up to the implementation of the service.

5.3.3.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *TriggeredItem* in the XML namespace "urn:schemas-upnp-org:phone:messaging" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

5.3.3.2 Description of fields in the *TriggeredItem* structure

```
<?xml version="1.0" encoding="utf-8"?>
<calendar:triggeredItem
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:itemID>ID of the item</calendar:itemID>
</calendar:triggeredItem>
```

<xml>

Required. Case Sensitive.

<triggeredItem>

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas upnp org:phone:calendar”). This namespace “urn:schemas upnp org:phone:calendar” defines the following elements and attributes:

<itemID>

Required. includes the triggered calendar item identifier.

5.3.4 *MemoInfo*

This state variable is evented when a memo is posted to the Telephony Server (TS). When a memo is posted at the TS, the TS returns the *memoID*. Once the memo is posted, the TS notifies the memo to all subscribed TelCP. The TS also delivers the memo to the devices that registered with the TS by means of delivery mechanisms.

5.3.4.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *MemoInfo* in the XML namespace "urn:schemas-upnp-org:phone:calendar" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

5.3.4.2 Description of fields in the *MemoInfo* structure

```
<?xml version="1.0" encoding="utf-8"?>
<calendar:memoInfo
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:memo>
      <calendar:memoID>ID of the memo</calendar:memoID>
      <calendar:memoText>Content of the memo</calendar:memoText>
   </calendar:memo>
</calendar:MemoInfo>
```

`<xml>`

Required. Case Sensitive.

`<MemoInfo>`

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas upnp org:phone:calendar”). This namespace “urn:schemas upnp org:phone:calendar” defines the following elements and attributes:

`<Memo>`

Required. contains the newly posted memo. It includes the following sub elements:

`<MemoID>`

Required, ID of the memo.

`<MemoText>`

Required, the content of the memo in text.

5.3.5 *A_ARG_TYPE_TriggeredItems*

This state variable contains the identifier of all the triggered calendar items. This state variable can be cleared periodically to clear off the already evented calendar items but this will be up to the implementation of the service.

5.3.5.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_TriggeredItems* in the XML namespace "`urn:schemas-upnp-org:phone:calendar`" which is located at "`http://www.upnp.org/schemas/phone/calendar-v1.xsd`".

5.3.5.2 Description of fields in the *A_ARG_TYPE_TriggeredItems* structure

```
<?xml version="1.0" encoding="UTF-8"?>
<calendar:triggeredItems
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:itemID>ID of the item</calendar:itemID>
      <!-- Any other items (if any) go here.-->
</calendar:triggeredItems>
```

`<xml>`

Required. Case Sensitive.

`<triggeredItem>`

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas upnp org:phone:calendar”). This namespace “urn:schemas upnp org:phone:calendar” defines the following elements and attributes:

`<itemID>`

Required. includes the triggered calendar item identifiers.

5.3.6 *A_ARG_TYPE_CalItems*

This state variable contains all the calendar items in the TS and their information as defined in [7].

5.3.6.1 Description of fields in the *A_ARG_TYPE_CallItems*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_CallItems* in the XML namespace "urn:schemas-upnp-org:phone:calendar" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

```
<?xml version="1.0" encoding="UTF-8"?>
<calendar:calendarItems
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:item ItemID="item ID" format="RFC5545" TelCPIDs="ID of the TelCPs">
      actual event as per RFC 5545.(textual format)
   </calendar:item>
      <!-- Any other items (if any) go here.-->
</calendar:calendarItems>
```

<xml>

Required. Case Sensitive.

<calendarItem>

Required. shall include the namespace declaration for the *Calendar* service Schema ("urn:schemas-upnp-org:phone:CalendarService"). This namespace "urn:schemas-upnp-org:phone:CalendarService" defines the following elements and attribute.

<item>

Required. includes the individual calendar event(object) stored in the format of [7]. The calendar oject is stored in the texual format and follow the rule specified in [7].

ItemID

Required, includes the unique identifierfor each calendat item stored.

format

Required, specifies the format used to store the calendar item. The value of this attribute shall be "RFC5545".

TelCPID

Allowed, specifies the TelCP who is associated with the calendar item.

5.3.7 *A_ARG_TYPE_ItemIDs*

This state variable contains unique identifier for the calendar item to be set in the TS and their information as defined in the *CalendarItem* state variable.

5.3.7.1 Description of fields in the *A_ARG_TYPE_ItemIDs*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_ItemIDs* in the XML namespace "urn:schemas-upnp-org:phone:calendar" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

```
<?xml version="1.0" encoding="utf-8"?>
<calendar:calendarItemID
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:itemID>ID of the item</calendar:itemID>
      <!-- Any other items (if any) go here.-->
</calendar:calendaritemID>
```

<xml>

Required. Case Sensitive.

<calendaritemID>

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas upnp org:phone:calendar”). This namespace “urn:schemas upnp org:phone:calendar” defines the following elements and attributes:

<itemID>

Required. includes the triggered calendar item identifier.

5.3.8 *A_ARG_TYPE_Memo*

This state variable contains the memo text. The memo is stored as an normal string value.

5.3.9 *A_ARG_TYPE_MemoID*

This state variable contains an unique identifier for the memo.

5.3.10 *A_ARG_TYPE_ItemDeliveryMethod*

This state variable specifies the delivery method for a triggered calendar Item or a memo.

5.3.10.1 Description of fields in the *A_ARG_TYPE_ItemDeliveryMethod*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_ItemDeliveryMethod* in the XML namespace "urn:schemas-upnp-org:phone:calendar" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

```
<?xml version="1.0" encoding="utf-8"?>
<calendar:itemDeliveryMethod
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:cams="urn:schemas-upnp-org:phone:calendar">
   <calendar:deliveryMethod itemID="MemoID or CalendarID”>
      <calendar:email>email address</calendar:email>
         <!-- Any other email address (if any) go here.-->
      <calendar:sms>tell uri</calendar:sms>
         <!-- Any other sms id (if any) go here.-->
      <calendar:sip>tell uri</calendar:sip>
         <!-- Any other sip address (if any) go here.-->
      <calendar:sns>
         <calendar:snsName>name of the social network service</calendar:snsName>
         <calendar:snsID>user id of the social network service</calendar:snsID>
      </calendar:sns>
         <!-- Any other sns address (if any) go here.-->
   </calendar:deliveryMethod>
</calendar:ItemDeliveryMethod>
```

<xml>

Required. Case Sensitive.

<ItemDeliveryMethod>

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas upnp org:phone:calendar”). This namespace “urn:schemas upnp org:phone:calendar” defines the following elements and attributes:

<deliveryMethod>

Required. contains mechanisms to deliver memo. This may include the attribute called itemID which will specify the calendar item id or memo id to which this delivery method is applicable. It includes the following sub elements.

ItemID

Allowed. includes the unique identifier for the calendar item or for the Memo.This attribute is included when the TelCP wants to set the delivery method for particular calendar event or particular Memo.

<email>

Allowed, the email id to which the memo will be delivered. This is a multi-valued property.

<sms>

Allowed, the tel-uri to deliver the memo via sms. This is a multi-valued property.

<sip>

Allowed, the sip address to deliver the memo via sip delivery mechanism. This is a multi-valued property.

<sns>

Allowed, the social network service name and address to deliver the memo. This is a multi-valued property.

5.3.11 *A_ARG_TYPE_MemoInfoList*

This state variable contains the list of memos posted on a TS.

5.3.11.1 Description of fields in the *A_ARG_TYPE_MemoInfoList*

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for *A_ARG_TYPE_MemoInfoList* in the XML namespace "urn:schemas-upnp-org:phone:calendar" which is located at "http://www.upnp.org/schemas/phone/calendar-v1.xsd".

```
<?xml version="1.0" encoding="utf-8"?>
<calendar:memoInfoList
   xsi:schemaLocation="urn:schemas-upnp-org:phone:calendar
   http://www.upnp.org/schemas/phone/calendar-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:calendar="urn:schemas-upnp-org:phone:calendar">
   <calendar:memoList>
      <calendar:memo>
         <calendar:memoID>ID of the memo</calendar:memoID>
         <calendar:memoText>Content of the memo</calendar:memoText>
      </calendar:memo>
         <!-- Any other memo (if any) go here.-->
   </calendar:memoList>
</calendar:MemoInfoList>
```

<xml>

Required. Case Sensitive.

<MemoInfoList>

Required. shall include the namespace declaration for the *Calendar* service Schema (“urn:schemas upnp org:phone:calendar”). This namespace “urn:schemas upnp org:phone:calendar” defines the following elements and attributes:

<memoList>

Required. contains list of memos posted in the TS. It includes the following sub elements.

<memo>

Required. containts information as deined in following sub-elements of a memo.

<memoID>

Required. the ID of the memo.

<memoText>

Required. the content of the memo in texts.

5.3.12 *A_ARG_TYPE_TelCPName*

This state variable describes a unique name for a TelCP. Each TelCP has its own name. A TelCP can register its name to the *Calendar* service by invoking the *RegisterTelCPName()* action. It is recommended that this state variable includes human readable and understanding string.

5.3.13 *A_ARG_TYPE_TelCPNameList*

This state variable contains the names of all the TelCPs. The format of this state variable is comma separated values of string.

5.3.14 *A_ARG_TYPE_Expires*

This state variable contains the duration as an integer value.

5.4 Eventing and Moderation

Table 2 — Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate [a] (seconds)	Logical Combination	Min Delta per Event [b]
CalendarItem	*YES*	*YES*	1		
TriggeredItem	*YES*	*YES*	1		
MemoInfo	*YES*	*YES*	1		
A_ARG_TYPE_TriggeredItems	*NO*	*NO*			
A_ARG_TYPE_CallItems	*NO*	*NO*			
A_ARG_TYPE_ItemIDs	*NO*	*NO*			
A_ARG_TYPE_Memo	*NO*	*NO*			
A_ARG_TYPE_MemoID	*NO*	*NO*			
A_ARG_TYPE_ItemDeliveryMethod	*NO*	*NO*			
A_ARG_TYPE_MemoInfoList	*NO*	*NO*			
A_ARG_TYPE_TelCPName	*NO*	*NO*			
A_ARG_TYPE_TelCPNameList	*NO*	*NO*			
A_ARG_TYPE_Expires	*NO*	*NO*			

a Determined by N, where Rate = (Event)/(N secs).

b (N) * (allowedValueRange Step).

5.4.1 Eventing of *CalendarItem*

This state variable can be evented when a new calendar item is set in the *Calendar* service in the TS. However, this state variable shall not be evented more than once in 1 s. If multiple request arrives within 1 s, then all the events will be accumulated into a single event message and the event message will be sent after 1 s.

5.4.2 Eventing of *TriggeredItem*

This state variable can be evented when the calendar item is triggered. However, this state variable shall not be evented more than once in 1 s. If multiple request arrives within 1 s, then all the events will be accumulated into a single event message and the event message will be sent after 1 s.

5.4.3 Eventing of *MemoInfo*

This state variable can be evented when a new memois added in the *Calendar* service in the TS. However, this state variable shall not be evented more than once in 1 s. If multiple request arrives within 1 s, then all the events will be accumulated into a single event message and the event message will be sent after 1 s.

5.5 Actions

Table 3 lists the actions of the *Calendar* service.

The *Calendar* service requires the implementation of all the actions necessary for delivering the complete set of functions.

Table 3 — Actions

Name	Device R/A [a]	Control Point R/A [b]
AddCalendarItems()	*R*	*R*
DeleteCalendarItems()	*A*	*A*
UpdateCalendarItems()	*A*	*A*
GetCalendarItems()	*A*	*A*
GetTriggeredItems()	*A*	*A*
PostMemo()	*R*	*R*
RegisterItemDelivery()	*A*	*A*
GetMemo()	*A*	*A*
RegisterTelCPName()	*A*	*A*
UnregisterTelCPName()	*A*	*A*
GetTelCPNameList()	*A*	*A*

a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

b For a control pont this column indicates whether a control point shall be capable of invoking this action, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

5.5.1 *AddCalendarItems()*

This action allows aTelCP to store the new calendar items into the TS. The calendar items are stored in the format defined in [7]. This will update the *CalendarItem* state variable with the new calendar items. The newly added calendar items will be evented to all the subscribed TelCP(s).

5.5.1.1 Arguments

Table 4 — Arguments for *AddCalendarItems()*

Argument	Direction	relatedStateVariable
CalItems	*IN*	*A_ARG_TYPE_CalItems*
ItemIDs	*OUT*	*A_ARG_TYPE_ItemIDs*

5.5.1.2 Argument Descriptions

The input argument *CalItems* contains the calendar item to be set. The calendar item is stored in the format specified in [7].

The output argument *ItemID* is returned by this service which uniquely identifies the calendar items in the *CalendarItem* state variable.

5.5.1.3 Service Requirements

None.

5.5.1.4 Control Point Requirements When Calling The Action

None.

5.5.1.5 Dependency on Device State

None.

5.5.1.6 Effect on Device State

None.

5.5.1.7 Errors

Table 5 — Error Codes for *AddCalendarItems()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Calendar Format Not Supported	calendar item format not supported.
702	Error in Calendar item	Represents errors in the calendar items.
703	Internal Error	Not able to create the calendar item due to internal errors.

5.5.2 *DeleteCalendarItems()*

This action allows a TelCP to delete existing calendar items from the TS. The input argument *ItemIDs* identify the calendar items to be deleted.

5.5.2.1 Arguments

Table 6 — Arguments for *DeleteCalendarItems()*

Argument	Direction	relatedStateVariable
ItemIDs	*IN*	*A_ARG_TYPE_ItemIDs*

5.5.2.2 Argument Descriptions

The input argument *ItemIDs* identifies the calendar items to be deleted.

5.5.2.3 Service Requirements

None.

5.5.2.4 Control Point Requirements When Calling The Action

None.

5.5.2.5 Dependency on Device State

None.

5.5.2.6 Effect on Device State

None.

5.5.2.7 Errors

Table 7 — Error Codes for *DeleteCalendarItems()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
704	Calendar Item does not exist	Calendar item represented by itemID does not exist.

5.5.3 *UpdateCalendarItems()*

This action allows a TelCP to update the existing calendar items from the the TS. The input argument *CalItems* identifies the calendar items to be updated.

5.5.3.1 Arguments

Table 8 — Arguments for *UpdateCalendarItems()*

Argument	Direction	relatedStateVariable

Argument	Direction	relatedStateVariable
CalItems	*IN*	*A_ARG_TYPE_CalItems*

5.5.3.2 Argument Descriptions

The input argument *CalItems* contains the calendar item to be set. The calendar item is stored in the format specified in [7]. The *CalItems* includes the *ItemID* of the caltendar item to be updated.

5.5.3.3 Service Requirements

None.

5.5.3.4 Control Point Requirements When Calling The Action

None.

5.5.3.5 Dependency on Device State

None.

5.5.3.6 Effect on Device State

None.

5.5.3.7 Errors

Table 9 — Error Codes for *UpdateCalendarItems()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Calendar Format Not Supported	calendar item format not supported.
702	Error in Calendar item	Represents errors in the calendar items.
703	Internal Error	Not able to create the calendar item due to internal errors.
704	Calendar Item does not exist	Calendar item to be updated does not exist.

5.5.4 *GetCalendarItems()*

This action allows a TelCP to retrieve existing calendar items from the TS. The input argument *ItemIDs* identifie the calendar items to be retrieved.

5.5.4.1 Arguments

Table 10 — Arguments for *GetCalendarItems()*

Argument	Direction	relatedStateVariable

Argument	Direction	relatedStateVariable
ItemIDs	*IN*	*A_ARG_TYPE_ItemIDs*
CalItems	*OUT*	*A_ARG_TYPE_CalItems*

5.5.4.2 Argument Descriptions

The input argument *ItemIDs* identifie the calendar items to be retrieved.

The output argument *CalItems* include all the calendar items requested by the TelCP.

5.5.4.3 Service Requirements

None.

5.5.4.4 Control Point Requirements When Calling The Action

None.

5.5.4.5 Dependency on Device State

None.

5.5.4.6 Effect on Device State

None.

5.5.4.7 Errors

Table 11 — Error Codes for *GetCalendarItems()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
704	Calendar Item does not exist	Calendar Item represented by ItemIDs does not exist.

5.5.5 *GetTriggeredItems()*

This action allows a TelCP to retrieve the triggered calendar items from the TS.

5.5.5.1 Arguments

Table 12 — Arguments for *GetTriggeredItems()*

Argument	Direction	relatedStateVariable
TriggeredItemIDs	*OUT*	*A_ARG_TYPE_TriggeredItems*

5.5.5.2 Argument Descriptions

The output argument *TriggeredItemIDs* identifies the triggered calendar items.

5.5.5.3 Service Requirements

None.

5.5.5.4 Control Point Requirements When Calling The Action

None.

5.5.5.5 Dependency on Device State

None.

5.5.5.6 Effect on Device State

None.

5.5.5.7 Errors

Table 13 — Error Codes for *GetTriggeredItems()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.

5.5.6 *PostMemo()*

This action allows a TelCP to post a memo on the TS which will be notified to all subscribed TelCPs by the TS. A TS may decide to store the memo. However, this behavior is implementation specific and so is also how long the memo will be kept on the TS.

5.5.6.1 Arguments

Table 14 — Arguments for *PostMemo()*

Argument	Direction	relatedStateVariable
Memo	*IN*	*A_ARG_TYPE_Memo*
MemoID	*OUT*	*A_ARG_TYPE_MemoID*

5.5.6.2 Argument Descriptions

The input argument *Memo* contains the content of the memo to be notified to all subscribed TelCPs.

The output argument *MemoID* contains an unique ID assigned to the memo by the TS for the specific memo.

5.5.6.3 Service Requirements

The behavior of the TS about storing the memo is implementation specific. The TS may decide to delete the memo instantaneously as soon as it is notified to all subscribed TelCPs. In which case, if the UPnP action is invoked to retrieve that specific memo will fail.

5.5.6.4 Control Point Requirements When Calling The Action

The TelCP invoke this action when it wants to send a memo to all subscribed TelCPs.

5.5.6.5 Dependency on Device State

None.

5.5.6.6 Effect on Device State

None.

5.5.6.7 Errors

Table 15 — Error Codes for *PostMemo()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
705	Could not post the memo	The TS is unable to post the memo due to some internal problem.

5.5.7 *RegisterItemDelivery()*

This action allows a TelCP to register a delivery method to receive memo or triggered calendar notifications from the home network. A user may want to receive memo or triggered calendar notifications to a mobile device while the user is outside of the home network. This action allows the user to register address and delivery mechanism with the TS to receive memo or triggered calendar notifications to a user preferred delivery mechanism when the user is outside the home network. A user can select more than one delivery methods such SMS and email etc.

5.5.7.1 Arguments

Table 16 — Arguments for *RegisterItemDelivery()*

Argument	Direction	relatedStateVariable
ItemDeliveryMethod	*IN*	*A_ARG_TYPE_ItemDeliveryMethod*
Expires	*IN*	*A_ARG_TYPE_Expires*

5.5.7.2 Argument Descriptions

The input argument *ItemDeliveryMethod* contains the delivery method by which memo or triggered calendar items will be notified to the registered device. The delivery method could be email, SMS etc.

The input argument *Expires* contains the duration for which this registration is valid. The registration will expire and the user will no longer receive the notifications once the duration expires with his preferred memo delivery mechanism. If the value of Expires is set to “0” then the TS unregistered the delivery mechanism.

5.5.7.3 Service Requirements

None

5.5.7.4 Control Point Requirements When Calling The Action

The TelCP shall invoke this action periodically to keep registration active before the duration expires.

5.5.7.5 Dependency on Device State

None.

5.5.7.6 Effect on Device State

None.

5.5.7.7 Errors

Table 17 — Error Codes for *RegisterItemDelivery()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
704	Calendar Item does not exist	The calendar item does not exist.
706	Invalid Delivery Methods	One or more delivery methods weren't registered.

5.5.8 *GetMemo()*

This action allows a TelCP to retrieve memo from the TS. This action will return the list of memos posted on the TS if the input argument is *MemoID* null otherwise this action will return only the memo identified by the *MemoID*.

5.5.8.1 Arguments

Table 18 — Arguments for *GetMemo()*

Argument	Direction	relatedStateVariable
MemoID	*IN*	*A_ARG_TYPE_MemoID*
MemoInfoList	*OUT*	*A_ARG_TYPE_MemoInfoList*

5.5.8.2 Argument Descriptions

The input argument *MemoID* contains the identification of the memo. A null value is allowed.

The output argument *MemoInfoList* contains the list of memos posted in the TS. If the input argument *MemoID* contains a valid id for a memo, then the output argument only returns the memo identified by the *MemoID* otherwise the output contains all the memos stored in the TS.

5.5.8.3 Service Requirements

None.

5.5.8.4 Control Point Requirements When Calling The Action

None.

5.5.8.5 Dependency on Device State

None.

5.5.8.6 Effect on Device State

None.

5.5.8.7 Errors

Table 19 — Error Codes for *GetMemo()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
707	Memo does not exist	The memo identified by the memo id does not exist.

5.5.9 *RegisterTelCPName()*

This action allows a TelCP to register its user friendly name tothe TS. The input argument *TelCPName* includes the TelCPName to be registered to the TS.

5.5.9.1 Arguments

Table 20 — Arguments for *RegisterTelCPName()*

Argument	Direction	relatedStateVariable
TelCPName	*IN*	*A_ARG_TYPE_TelCPName*

5.5.9.2 Argument Descriptions

The input argument *TelCPName* contains the user friendly name to be registered with the *Calendar* service.

5.5.9.3 Service Requirements

This service shall check the uniqueness of the TelCP name when a TelCP invokes this action to register a *TelCPName*.

5.5.9.4 Control Point Requirements When Calling The Action

None.

5.5.9.5 Dependency on Device State

None.

5.5.9.6 Effect on Device State

None.

5.5.9.7 Errors

Table 21 — Error Codes for *RegisterTelCPName()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
708	Invalid TelCP	The specified TelCPName is not valid.
709	TelCPName already exists	The specified TelCPName is already in use.

5.5.10 *UnregisterTelCPName()*

This action allows a TelCP to unregister its TelCP name. The input argument *TelCPName* includes the user friendly name to be unregistered from the TS. If this action is invoked without registering the *TelCPName* then the action fails with an error code.

5.5.10.1 Arguments

Table 22 — Arguments for *UnregisterTelCPName()*

Argument	Direction	relatedStateVariable
TelCPName	*IN*	*A_ARG_TYPE_TelCPName*

5.5.10.2 Argument Descriptions

The input argument *TelCPName* contains the name to be unregister.

5.5.10.3 Service Requirements

None.

5.5.10.4 Control Point Requirements When Calling The Action

None.

5.5.10.5 Dependency on Device State

None.

5.5.10.6 Effect on Device State

None.

5.5.10.7 Errors

Table 23 — Error Codes for *UnregisterTelCPName()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
708	Invalid TelCPName	The specified TelCPName is invalid.

5.5.11 *GetTelCPNameList()*

This action allows a TelCP to get all registered TelCP names with *Calendar* service.

5.5.11.1 Arguments

Table 24 — Arguments for *GetTelCPNameList()*

Argument	Direction	relatedStateVariable
TelCPName	*OUT*	*A_ARG_TYPE_TelCPNameList*

5.5.11.2 Argument Descriptions

The output argument *TelCPNameList* includes the list of TelCP name registered with the *Calendar* service.

5.5.11.3 Service Requirements

None.

5.5.11.4 Control Point Requirements When Calling The Action

None.

5.5.11.5 Dependency on Device State

None.

5.5.11.6 Effect on Device State

None.

5.5.11.7 Errors

Table 25 — Error Codes for *GetTelCPNameList()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.

5.5.12 Error Code Summary

Table 26 lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 26 — Error Code Summary

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
700		Reserved for future extensions.
701	Calendar Format Not Supported	calendar item format not supported.
702	Error in Calendar item	Represents errors in the calendar items.
703	Internal Error	Not able to create the calendar item due to internal errors.
704	Calendar Item does not exist	Calendar item does not exist.
705	Could not post the memo	The TS is unable to post the memo due to some internal problem.
706	Invalid Delivery Methods	One or more delivery methods weren't registered.
707	Memo does not exist	The memo identified by the memo id does not exist.
708	Invalid TelCP	The specified TelCPName is not valid.
709	TelCPName already exists	The specified TelCPName is already in use.

Note: 800-899 Error Codes are not permitted for standard actions. See Control clause in [1] for more details.

5.6 Service Behavioral Model

None.

6 XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

   <specVersion>
      <major>1</major>
      <minor>0</minor>
   </specVersion>

   <actionList>

      <action>
         <name>AddCalendarItems</name>
         <argumentList>
            <argument>
               <name>CalItems</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_CalItems
               </relatedStateVariable>
            </argument>

            <argument>
               <name>ItemIDs</name>
```



```
            <direction>out</direction>
            <relatedStateVariable>
               A_ARG_TYPE_ItemIDs
            </relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>DeleteCalendarItems</name>
      <argumentList>
         <argument>
            <name>ItemIDs</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_ItemIDs
            </relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>UpdateCalendarItems</name>
      <argumentList>
         <argument>
            <name>Caltems</name>
            <direction>in</direction>

            <relatedStateVariable>
               A_ARG_TYPE_CalItems
            </relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>GetCalendarItems</name>
      <argumentList>
         <argument>
            <name>ItemIDs</name>
            <direction>in</direction>
            <relatedStateVariable>
               A_ARG_TYPE_ItemIDs
            </relatedStateVariable>
         </argument>
         <argument>
            <name>Caltems</name>
            <direction>out</direction>
            <relatedStateVariable>
               A_ARG_TYPE_CalItems
            </relatedStateVariable>
         </argument>
      </argumentList>
   </action>
   <action>
      <name>GetTriggeredItems</name>
      <argumentList>
         <argument>
            <name>TriggeredItemIDs</name>
            <direction>out</direction>
            <relatedStateVariable>
               A_ARG_TYPE_TriggeredItems
            </relatedStateVariable>
```

```
            </argument>
         </argumentList>
      </action>
      <action>
         <name>PostMemo</name>
         <argumentList>
            <argument>
               <name>Memo</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_Memo
               </relatedStateVariable>
            </argument>
            <argument>
               <name>MemoID</name>
               <direction>out</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_MemoID
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>RegisterItemDelivery</name>
         <argumentList>
            <argument>
               <name>ItemDeliveryMethod</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_ItemDeliveryMethod
               </relatedStateVariable>
            </argument>
            <argument>
               <name>Expires</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_Expires
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>GetMemo</name>
         <argumentList>
            <argument>
               <name>MemoID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_MemoID
               </relatedStateVariable>
            </argument>
            <argument>
               <name>MemoInfoList</name>
               <direction>out</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_MemoInfoList
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
```



```
      <action>
         <name>RegisterTelCPName</name>
         <argumentList>
            <argument>
               <name>TelCPName</name>
               <direction>in</direction>
               relatedStateVariable>
                  A_ARG_TYPE_TelCPName
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>UnregisterTelCPName</name>
         <argumentList>
            <argument>
               <name>TelCPName</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_TelCPName
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>GetTelCPNameList</name>
         <argumentList>
            <argument>
               <name>TelCPName</name>
               <direction>out</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_TelCPNameList
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
   </actionList>
   <serviceStateTable>
      <stateVariable sendEvents="yes">
         <name>CalendarItem</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="yes">
         <name>TriggeredItem</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="yes">
         <name>MemoInfo</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_CalItems</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_TriggeredItems</name>
         <dataType>string</dataType>
      </stateVariable>

      <stateVariable sendEvents="no">
```

```
            <name>A_ARG_TYPE_ItemIDs</name>
            <dataType>string</dataType>
         </stateVariable>
         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_Memo</name>
            <dataType>string</dataType>
         </stateVariable>
         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_MemoID</name>
            <dataType>string</dataType>
         </stateVariable>
         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_MemoInfoList</name>
            <dataType>string</dataType>
         </stateVariable>
         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_ItemDeliveryMethod</name>
            <dataType>string</dataType>
         </stateVariable>

         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_TelCPName</name>
            <dataType>string</dataType>
         </stateVariable>
         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_TelCPNameList</name>
            <dataType>string</dataType>
         </stateVariable>

         <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_Expires</name>
            <dataType>ui4</dataType>
         </stateVariable>

      </serviceStateTable>
</scpd>
```

Annex A
(normative)

XML Schema

Annex A provides the global XML Schema for syntactical validation of all the XML fragments used in the *Calendar* service.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:tns="urn:schemas-upnp-org:phone:calendar"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:peer="urn:schemas-upnp-org:phone:peer"
targetNamespace="urn:schemas-upnp-org:phone:calendar" elementFormDefault="unqualified"
attributeFormDefault="unqualified">
    <import namespace="urn:schemas-upnp-org:phone:peer"
schemaLocation="http://www.upnp.org/schemas/phone/peer-v2.xsd"/>
    <element name="calendarItem">
        <annotation>
            <documentation>Corresponding state variable: calendarItem</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="item" type="tns:item"/>
            </sequence>
        </complexType>
    </element>
    <element name="triggeredItem">
        <annotation>
            <documentation>Corresponding state variable: triggeredItem</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="itemID" type="string"/>
            </sequence>
        </complexType>
    </element>
    <element name="memoInfo">
        <annotation>
            <documentation>Corresponding state variable: MemoInfo </documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="memo" type="tns:memo"/>
            </sequence>
        </complexType>
    </element>
    <element name="triggeredItems">
        <annotation>
            <documentation> Corresponding state variable: triggeredItem</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="itemID" type="string"  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </complexType>
    </element>
    <element name="calendarItems">
        <annotation>
            <documentation>Corresponding state variable: A_ARG_TYPE_CalItems</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="item" type="tns:item" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </complexType>
    </element>
    <element name="calendarItemID">
        <annotation>
            <documentation> Corresponding state variable: A_ARG_TYPE_ItemIDs</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="itemID" type="string" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </complexType>
    </element>
    <element name="memo" type="string">
```

```
        <annotation>
            <documentation> Corresponding state variable: A_ARG_TYPE_Memo</documentation>
        </annotation>
    </element>
    <element name="memoID" type="string">
        <annotation>
            <documentation>Corresponding state variable: A_ARG_TYPE_MemoID</documentation>
        </annotation>
    </element>
    <element name="itemDeliveryMethod">
        <annotation>
            <documentation> Corresponding state variable:
A_ARG_TYPE_ItemDeliveryMethod</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="deliveryMethod" type="tns:deliveryMethod" minOccurs="1"/>
            </sequence>
        </complexType>
    </element>
    <element name="memoInfoList">
        <annotation>
            <documentation>Corresponding state variable:
A_ARG_TYPE_MemoInfoList</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="memoList" type="tns:memoList"/>
            </sequence>
        </complexType>
    </element>
    <element name="telCPName" type="string">
        <annotation>
            <documentation>Corresponding state variable: A_ARG_TYPE_TelCPName</documentation>
        </annotation>
    </element>
    <element name="telCPNameList" type="string">
        <annotation>
            <documentation>Corresponding state variable:
A_ARG_TYPE_TelCPNameList</documentation>
        </annotation>
    </element>
    <element name="expires" type="integer">
        <annotation>
            <documentation>Corresponding state variable: A_ARG_TYPE_Expires</documentation>
        </annotation>
    </element>
    <complexType name="item">
        <sequence>
            <element name="item"></element>
        </sequence>
        <attribute name="Itemid"></attribute>
        <attribute name="format"></attribute>
        <attribute name="TelCPID"></attribute>
    </complexType>
    <complexType name="memo">
        <sequence>
            <element name="memoID" type="string"/>
            <element name="memoText" type="string"/>
        </sequence>
    </complexType>
    <complexType name="deliveryMethod">
        <sequence>
            <element name="email" type="string"  minOccurs="0" maxOccurs="unbounded"/>
            <element name="sms" type="string" minOccurs="0" maxOccurs="unbounded"/>
            <element name="sip" type="string" minOccurs="0" maxOccurs="unbounded"/>
            <element name="sns" type="tns:sns" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="itemID" type="string" use="optional"/>
    </complexType>
    <complexType name="sns">
        <sequence>
            <element name="snsName" type="string"/>
            <element name="snsID" type="string"/>
        </sequence>
    </complexType>
    <complexType name="memoList">
        <sequence>
            <element name="memo" type="tns:memo" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
```



```
    </complexType>
</schema>
```

Annex B
(informative)

Theory of Operation

B.1 Memo Handling

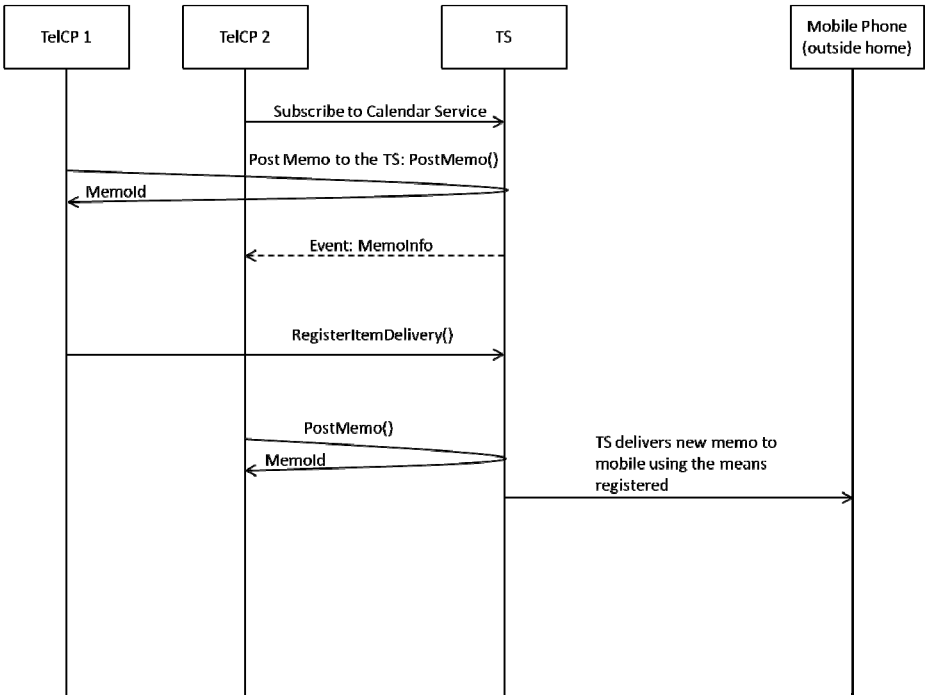

Figure B.1 — Memo handling

The memo feature allows a Telephony Control Point (TelCP) to post a memo on a Telephony Server (TS) which will be notified by the TS to all other subscribed TelCPs. This is realized by invocation of the *PostMemo()* action on the TS. A memo consists of a memo identification information and content of the memo. The content of a memo consists of texts. Once a memo is posted on the TS, the TS notifies the memo to all the subscribed TelCPs by use of eventing. A user may want to receive memo to a mobile device while the user is outside of the home network. The memo feature allows the user to register address and delivery mechanism with the TS to receive memo to a user preferred delivery mechanism when the user is outside the home network. The delivery mechanism could be SMS, email, SIP or others. A Telephony Server (TS) can notify a posted memo to a device outside of the home network by using the mechanism registered by a TelCP with the TS. In the case of SIP address registered with the TS by a device for memo delivery, the TS can use SIP MESSAGE or INVITE method to deliver the memo to a device outside the home network. The TS supporting the memo feature also provides the mechanism for a TelCP to un-register its previously registered notification request with the TS.

B.2 Calendar Object

IETF has defined the structure for the calendar object for interoperability between dissimilar calendaring and scheduling applications in [7].

This specification recommends to use the calendar object specified in [7] for better interoperability.

[7] defines the structure for the calendar object and it is called as iCalendar object. The structure is defined as a texual representation for representing the calendar events. Calendar components defined by [7] are referred to with capitalized, quoted-strings of texts. All calendar components start with the letter "V". For example, "VEVENT" refers to the event calendar component, and "VJOURNAL" refers to the daily journal calendar component. The following example shows an simple way of representing the calendar object using [7].

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
UID:19970610T172345Z-AF23B2@example.com
DTSTAMP:19970610T172345Z
DTSTART:19970714T170000Z
DTEND:19970715T040000Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

The calendar event is represented as a textually defined components. The above example shows the simple calendar event for the “Bastille Day Party”. The calendar object start with the component called VCALENDAR, to present it as a calendar object. The VEVENT component represents the actual start of the event. The DTSTART and DTEND component is used to represent the time information for the event. The SUMMARY component is used to the specify free text related to the event. These components are used to represent the calendar event. There are many other calendar components defined in the [7]. The following are the important components.

- VEVENT: Provide a grouping of component properties that describe an event for e.g. birthday etc.
- VJOURNAL: Provide a grouping of component properties that describe a journal entry.
- VFREEBUSY: Provide a grouping of component properties that describe either a request for free/busy time.
- VTIMEZONE: Provide the timezone information.
- VALARM: Provide a grouping of component properties that define an alarm.

The example below shows structure for storing the alarm event.

```
BEGIN:VALARM
TRIGGER;VALUE=DATE-TIME:19970317T133000Z
REPEAT:4
DURATION:PT15M
ACTION:AUDIO
ATTACH;FMTTYPE=audio/basic:ftp://example.com/pub/ sounds/bell-01.aud
END:VALARM
```

The TRIGGER component specifies when the alarm should be triggered. The other components like REPEAT, DURATION, ACTION, and ATTACH are use to specify the alarm related preferences. The [7] gives a simple texual representation for the calendar object. Please refer to [7] for more details.

B.3 Calendar Event Handling

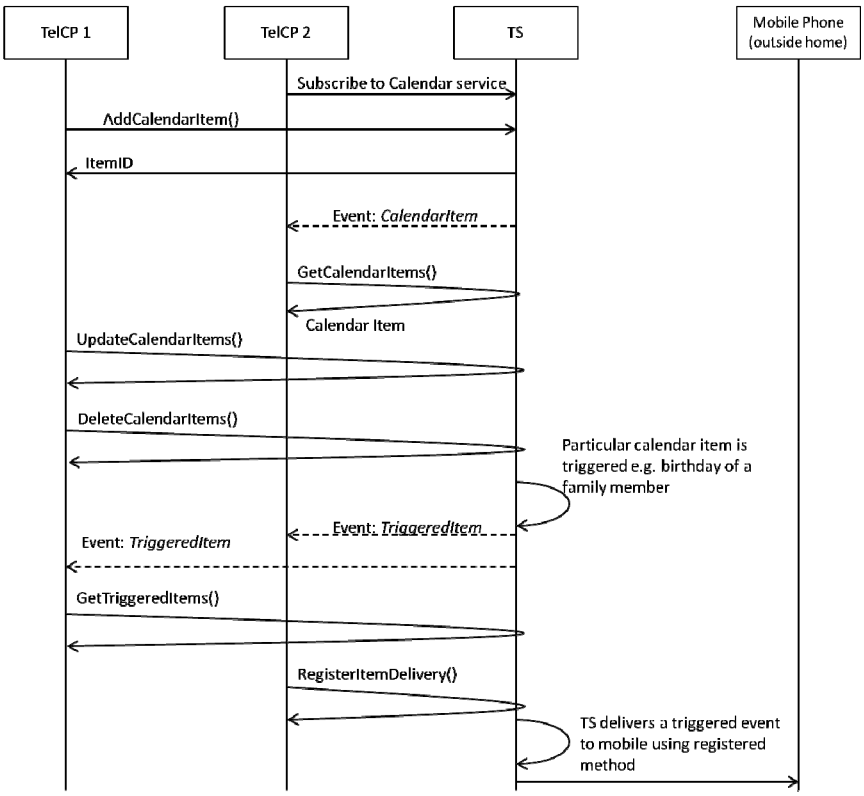

Figure B.2 — Logical flows for calendar service

The *Calendar* service allows a TelCP to set the calendar items into TS and share the calendar items between family members.

The TelCP invokes the *AddCalendarItems()* action to set a calendar item(s) in the TS. Each calendar item is identified by unique identifier *ItemID* which is returned in the *AddCalendarItems()* action.

The TS sends the *CalendarItem* event to all the subscribed TelCP(s) for notifying the new calendar item(s) added in the *Calendar* service. The TelCP(s) can invoke the *GetCalendarItems()* action to retrieve the detail of a particular calendar item.

The TS invokes the *UpdateCalendarItems()* or *DeleteCalendarItems()* action to update an existing calendar item or to delete an existing calendar item respectively.

When a particular calendar item is triggered (e.g., triggering of a birthday reminder), the TS sends the *TriggeredItem* event to all the subscribed TelCP(s). The TelCP(s) can invoke the *GetTriggeredItems()* action to retrieve the details of the triggered calendar item(s).

The user can register to deliver the triggered calendar items when user is outside of the Home Network. The *Calendar* service allows the user to register address and delivery mechanism with the TS to receive triggered calendar item to a user preferred delivery mechanism when the user is outside the home network. The delivery mechanism could be SMS, email, SIP address or others.

B.4 Multiple TelCP Handling for Alarm Based Calendar Items

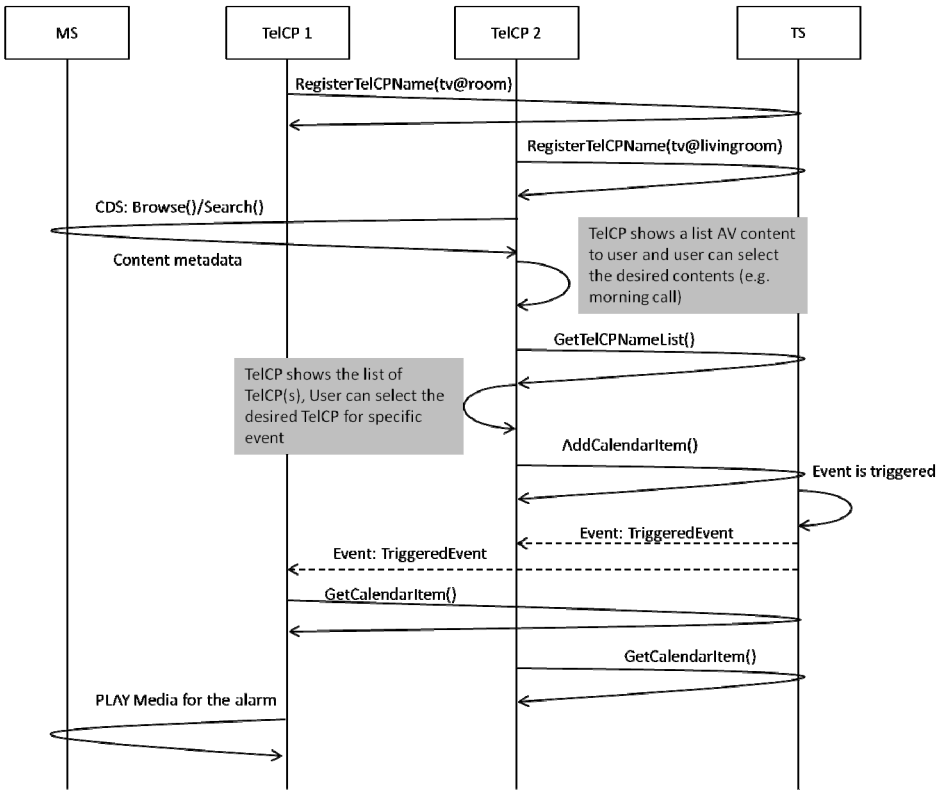

Figure B.3 — Multiple TelCP handling for Alarm use case

The *Calendar* service allows a TelCP to associate any TelCP to a calendar item who will then be able to handle the calendar item when it is triggered (e.g, playing the Alarm for birthday reminder).

The TelCP(s) should register their user-friendly name to the TS by invoking the *RegisterTelCPName()* action.

The TelCP can retrieve the names of all the registered TelCP(s) by invoking the action *GetTelCPNameList()*. To associate any TelCP to a particular calendar item, a TelCP can include the registered name of that TelCP into the calendar object. In the above example the TelCP2 (with registerred name as ”tv@livingroom”) sets the new calendar item and associates TelCP1 with the calendar item (with registerred name as a “tv@room”).

When a particular calendar item is triggered and to associated with TelCP1, the TelCP1 will play the media associated with the alarm event.

Annex C
(informative)

Bibliography

The following documents, in whole or in part, may be useful for understanding this document but they are not essential for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[11] – *TelephonyArchitecture:2*, UPnP Forum, December 10, 2012. Available at: http://www.upnp.org/specs/phone/UPnP-phone-TelephonyArchitecture-v2-20121210.pdf. Latest version available at: http://www.upnp.org/specs/phone/UPnP-phone-TelephonyArchitecture.pdf.

ICS 35.200

Price based on 36 pages