INTERNATIONAL STANDARD

ISO/IEC 29341-26-14

First edition
2017-09

Information technology — UPnP Device Architecture —

Part 26-14: Telephony device control protocol — Level 2 — Address book service

Technologies de l'information — Architecture de dispositif UPnP —

Partie 26-14: Protocole de contrôle de dispositif de téléphonie — Niveau 2 — Service de carnet d'adresses

Reference number
ISO/IEC 29341-26-14:2017(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 29341-26-14 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 [1]. It defines a service type referred to herein as *AddressBook*.

This service provides the following functionalities:

- Managing the network address book: accessing the network address book, adding new contact information in the network address book, synchronizing the local address book of Telephony Server with the network address book etc.
- Contact Share: sharing the local/network address book's contact information with other contacts of the user.
- Personal Contact Card: enabling mangement of his/her own contact information and sharing his own information with his/her contacts.
- Contact Invitation; allow handling of the Contact Invitation from the remote party.

This service does not provide the following functionalities:

- Interaction between Telephony Server and the network address book.

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 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, N. Freed, Innosoft, N. Borenstein, First Virtual, November 1996. Available at: http://www.ietf.org/rfc/rfc2046.txt.

[4] – 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.

[5] – 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.

[6] – 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.

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

[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 General terms

3.3.1

Personal Contact Card
PCC
collection of personal contact information that a user defines about him/herself. This also consider as the user's own profile information as available in various social networking sites.

3.3.2

Contact Invitation
mechanism for mutual PCC information sharing between the peer.

3.3.3

Contact Sharing
mechanism for sharing contact information from the address book with the remote party.

3.4 Abbreviated terms

3.4.1

GUI
Graphical User Interface

3.4.2

ID
Identifier

3.4.3

TC
Telephony Client

3.4.4

TelCP
Telephony Control Point

3.4.5

TS
Telephony Server

3.4.6

VoIP
Voice over IP

3.4.7

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:*AddressBook:1*

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

5.2 *AddressBook* Service Architecture

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

The *AddressBook* service provides a TelCP with the following address book features:

- Managing network address book (fetching the contact information from the network address book, exporting the network address book contacts in the local address book).
- Contact share: sharing of full or part of contact information with other contacts.
- Managing Personal Contact Card (PCC): managing personal information (e.g. personal profile and contact information, sharing personal information with friends/contacts.
- Contact Invitation handling.

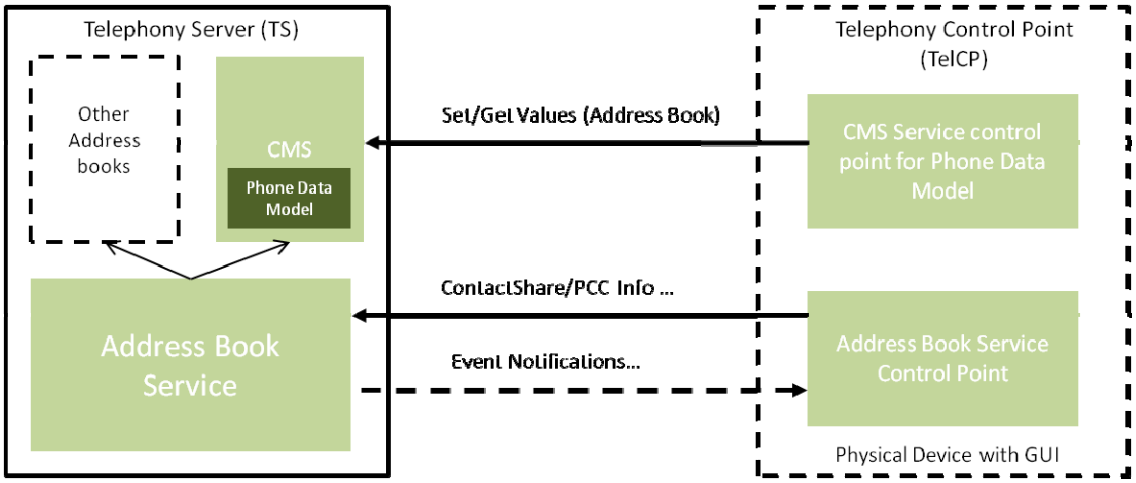

Figure 1 — *AddressBook* Service Architecture

The *AddressBook* service provides the advance features on the address book stored in the Telephony Server. The address book can be implemented on the Telephony Server using either CMS and *Phone Data Model* as defined [10] in or using any other properitory implementation.

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
IncomingRequest	*R*	**String** (XML fragment)	See 5.3.2
A_ARG_TYPE_IncomingRequests	*A*	**String** (XML fragment)	See 5.3.3
A_ARG_TYPE_SharedContacts	*A*	**String** (XML fragment)	See 5.3.4
A_ARG_TYPE_SharedInfo	*A*	**String**	See 5.3.5
A_ARG_TYPE_TargetContacts	*R*	**String** (XML fragment)	See 5.3.6
A_ARG_TYPE_ReqID	*A*	**String**	See 5.3.7
A_ARG_TYPE_NetworkAddressBookID	*R*	**String**	See 5.3.8

[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 *IncomingRequest*

The format of the *IncomingRequest* state variable is an XML document. It includes information about the incoming address book features handling requests from the WAN side such as incoming Contact Sharing request, request to share of a PCC information or contact invition. It contains all the necessary information (like type of a incoming request, initiator of the request, unique request identifier and other addition information etc) about the incoming request to handle (i.e. accept or reject the incoming request) a particular request.

The *IncomingRequest* state variable is evented when *AddressBook* service receives an incoming contact share request, incoming PCC share request, or incoming Contact Invitation request.

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 *IncomingRequest* in the XML namespace "urn:schemas-upnp-org:phone:messaging" which is located at "http://www.upnp.org/schemas/phone/AddressBook-v1.xsd".

5.3.2.2 Description of fields in the *IncomingRequest* structure

```
<?xml version="1.0" encoding="UTF-8"?>
<AddressBook:incomingRequest
   xmlns:AddressBook="urn:schemas-upnp-org:phone:addressbook
   http://www.upnp.org/schemas/phone/addressbook-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:peer=”urn:schemas-upnp-org:phone:peer">
   <request>
      <type>incoming request type</type>
      <reqID>request ID</reqID>
      <fromcontactid>
         <peer:id>contact id of the requester<peer:/id>
      </fromcontactid>
      <status>pending/Accepted/Rejected</status>
      <note>
         informative text information, e.g. sharing jeyoung’s mobile number
      </note>
      <sharedinformationURL>
         URL pointing to additional information for e.g shared contact information…
      </sharedinformationURL>
   </request>
</AddressBook:incomingRequest>
```

`<xml>`

Required. Case Sensitive.

`<incomingRequest>`

Required. shall include the name space declaration for the complex type <peerType> (“urn:schemas-upnp-org:phone:peer”) and the namespace declaration for the *AddressBook* service Schema (“urn:schemas-upnp-org:phone:addressbook”). This namespace “urn:schemas-upnp-org:phone:addressbook” defines the following elements and attributes:

`<request>`

Allowed. includes the required information for all the incoming address book handling requests. This element includes information about the incoming request like type of an incoming request, unique identifier for the request, status of the request etc. This element can appear zero or more times. This element includes following sub-elements:

`<type>`

Required. indicates the type of an incoming requests. The possible type of the incoming requests are “ContactShare”, “PCCShare”, and “Contactinvitation”.

`<reqID>`

Required. unique identifier for the incoming request.

`<fromcontactid>`

Required, peer:peerType. identifies the request originator.

`<status>`

Required. indentifies the status of the incoming request, e.g. “pending”, “Accepted”, “Rejected”.

<note>

Allowed. contains the texual information for corresponding address request information.

<sharedinformationURL>

Allowed. contains the URL which will include the detail information for the incoming request.

5.3.3 *A_ARG_TYPE_IncomingRequests*

The format of the *A_ARG_TYPE_IncomingRequests* state variable is an XML document. It includes information about all the active incoming handling requests from the WAN side such as incoming Contact Sharing request, request to sharing the PCC information or contact invitation.

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 *A_ARG_TYPE_IncomingRequests* in the XML namespace "urn:schemas-upnp-org:phone:messaging" which is located at "http://www.upnp.org/schemas/phone/AddressBook-v1.xsd".

5.3.3.2 Description of fields in the *A_ARG_TYPE_IncomingRequests* structure

```
<?xml version="1.0" encoding="UTF-8"?>
<AddressBook:incomingRequests
   xmlns:AddressBook="urn:schemas-upnp-org:phone:addressbook
   http://www.upnp.org/schemas/phone/addressbook-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:peer=”urn:schemas-upnp-org:phone:peer">
   <request>
      <type>incoming request type</type>
      <reqID>request ID</reqID>
      <fromcontactid>
         <peer:id>contact id of the requester<peer:/id>
      </fromcontactid>
      <status>pending/Accepted/Rejected</status>
      <note>
         informative text information, e.g. sharing jeyoung’s mobile number
      </note>
      <sharedinformationURL>
         URL pointing to additional information for e.g shared contact information…
      </sharedinformationURL>
   </request>
      <!--- Any other incoming requests currently active --->
</AddressBook:incomingRequests>
```

<xml>

Required. Case Sensitive.

<incomingRequests>

Required. shall include the name space declaration for the complex type <peerType> (“urn:schemas-upnp-org:phone:peer”) and the namespace declaration for the AddressBook service Schema (“urn:schemas-upnp-org:phone:addressbook”). This namespace “urn:schemas-upnp-org:phone:addressbook” defines the following elements and attributes:

<request>

Allowed. This element includes the required information for all the incoming address book handling requests. This element includes information about the incoming request like type of an incoming request, unique identifier for the request, status of the request etc. This element can appear zero or more times. This element includes following sub elements:

<type>

Required. indicates the type of an incoming requests. The possible type of the incoming requests are “ContactShare”, “PCCShare”, and “Contactinvitation”.

<reqID>

Required. unique identifier for the incoming request.

<fromcontactid>

Required. peer:peerType. identifies the request originator.

<status>

Required. indentifies the status of the incoming request for e.g. “pending”, “Accepted”, “Rejected”.

<note>

Allowed. contains the texual information for corresponding address request information.

<sharedinformationURL>

Allowed. contains the URL which will include the detail information for the incoming request.

5.3.4 *A_ARG_TYPE_SharedContacts*

This state variable contains the information about the contacts to be shared with one or more other remote parties. The contact information to be shared can be from the address book stored in the Telephony Server, so this element can include the reference to such contacts from the address book. This element is an XML structure of type peer.

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 *A_ARG_TYPE_SharedContacts* in the XML namespace "urn:schemas-upnp-org:phone:messaging" which is located at "http://www.upnp.org/schemas/phone/Addressbook-v1.xsd".

5.3.4.2 Description of fields in the *A_ARG_TYPE_SharedContacts* structure

```
<?xml version="1.0" encoding="UTF-8"?>
<AddressBook:sharedContacts
   xsi:schemaLocation="urn:schemas-upnp-org:phone:AddressBook
   http://www.upnp.org/schemas/phone/addressbook-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:AddressBook="urn:schemas-upnp-org:phone:AddressBook"
   xmlns:peer="urn:schemas-upnp-org:phone:peer">
   <contact>
      <peer:id>ID of the peer</peer:id>
      <peer:contactInstanceId>
         The instance identifier for a contact referred by the Peer
      </peer:contactInstanceId>
         Other Peer elements if required…
   </contact>
      <!-- Any other contact information (if any) go here.-->
</AddressBook:sharedContacts>
```

<xml>

Required. Case Sensitive.

<sharedContacts>

Required. shall include the name space declaration for the complex type <peerType> (“urn:schemas-upnp-org:phone:peer”) and the namespace declaration for the AddressBook service Schema (“urn:schemas-upnp-org:phone:addressbook”). This namespace “urn:schemas-upnp-org:phone:addressbook” defines the following elements and attributes:

<contact>

Required, peer:peerType. identifies the contact to be shared. If contact to be shared is present in the local address book, then the contactInstanceId element may be included to refer the contact from the

address book; otherwise, the Id element will identify the contact. A TelCP can use the any other peer elements (e.g., name etc.) to identify the contact.

5.3.5 *A_ARG_TYPE_SharedInfo*

This state variable contains the contact information elements to be shared with the remote party. This state variable is used as a high level filter in the contact share and PCC share related actions. This element is a string which includes comma separated contact information elements. The empty string will represent null filter to share entire contact information.

The allowed values are as follows:

- “name” : represents the name of the contact.
- “email” : represents the email id of the contact.
- “SIP” : represents the SIP URL of the contact.
- “phone” : represents the phone number.
- “status” : represents the status information of the contact.
- “image” : represents the image of the contact.
- “” : (empty string).

5.3.6 *A_ARG_TYPE_TargetContacts*

This state variable contains the information about the target contacts for the corresponding request like contact share or PCC share. This element is an XML structure of type peer.

5.3.6.1 Description of fields in the *A_ARG_TYPE_TargetContacts*

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

```
<?xml version="1.0" encoding="UTF-8"?>
<AddressBook:targetContacts
   xsi:schemaLocation="urn:schemas-upnp-org:phone:AddressBook
   http://www.upnp.org/schemas/phone/addressbook-v1.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:AddressBook="urn:schemas-upnp-org:phone:AddressBook"
   xmlns:peer="urn:schemas-upnp-org:phone:peer">
   <contact>
      <peer:id>ID of the Peer</peer:id>
      <peer:contactInstanceId>
         The instance identifier for a contact referred by the Peer
      </peer:contactInstanceId>)
         Other peer elements if required…
   </contact>
      <!-- Any other contact information (if any) go here.-->
</AddressBook:targetContacts>
```

<xml>

Required. Case Sensitive.

<targetContacts>

Required. shall include the name space declaration for the complex type <peerType> (“urn:schemas-upnp-org:phone:peer”) and the namespace declaration for the AddressBook service Schema (“urn:schemas-upnp-org:phone:addressbook”). This namespace “urn:schemas-upnp-org:phone:addressbook” defines the following elements and attributes:

<contact>

Required. Peer:peerType. identifies the contact to be shared. If contact to be shared is present in the local address book then,the contactInstanceId element may be included to refer the contact from the address book otherwise the Id element will identify the contact. A TelCP can use the any other peer elements (e.g., name etc) to identify the contact.

5.3.7 *A_ARG_TYPE_RegID*

This state variable contains the unique identifier for the incoming address book handling requests.

5.3.8 *A_ARG_TYPE_NetworkAddressBookID*

This state variable contains the unique identifier for the network address book for e.g. “gmail”.

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]
IncomingRequest	*YES*	*YES*	1		
A_ARG_TYPE_IncomingRequests	*No*	*No*			
A_ARG_TYPE_SharedContacts	*No*	*No*			
A_ARG_TYPE_SharedInfo	*No*	*No*			
A_ARG_TYPE_TargetContacts	*No*	*No*			
A_ARG_TYPE_RegID	*No*	*No*			
A_ARG_TYPE_NetWorkAddressBookID	*No*	*No*			

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

[b] (N) * (allowedValueRange Step).

5.4.1 Eventing of *IncomingRequest*

This state variable can be evented when a new incoming address book handling related request arrives from the WAN side. However, this state variable shall not be evented more than 1 s. If multiple request arrives within a 1 s interval, then all the events shall 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 *AddressBook* service.

The *AddressBook* 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]
ShareContacts()	*A*	*A*
FetchContactInfo()	*R*	*R*

Name	Device R/A [a]	Control Point R/A [b]
SharePCC()	A	A
ImportContacts()	R	R
Accept()	A	A
Reject()	A	A
RetrieveIncomingRequests()	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 *ShareContacts()*

This action allows a TelCP to share a set of contacts to a remote party. The contacts can be shared from the local address book stored in the TS. The TelCP can share only part of the contact information instead of the entire contact.

5.5.1.1 Arguments

Table 4 — Arguments for *ShareContacts()*

Argument	Direction	relatedStateVariable
SharedContacts	*IN*	*A_ARG_TYPE_SharedContacts*
SharedInfo	*IN*	*A_ARG_TYPE_SharedInfo*
TargetContacts	*IN*	*A_ARG_TYPE_TargetContacts*

5.5.1.2 Argument Descriptions

The input argument *SharedContacts* contains the contact information to be shared with the remote party. The *SharedContacts* is an XML structure which include the contact information such as contact address, name etc. The contact information can be stored in the local address book in the TS. This argument can contain reference to the entry in the address book.

The input argument *SharedInfo* contain the list of contact information elements to be shared (e.g., name, email etc). This argument can contain empty string to share entire contact information from the local address book.

The input argument *TargetContacts* contain the target remote party identifier with whom the contacts need to be shared.

5.5.1.3 Service Requirements

The *AddressBook* service first retrieves the contact information to be shared specified by the input argument the *SharedContacts*. The shared contact can include the reference to the contact from the local address book. In case if the local address book is implemented using the *Phone Data Model* specified in [10], the references will be a instance identifier of the contacts stored in the TS *Phone Data Model*. The *AddressBook* service generates the shared contact information using filter specified in the input argument *SharedInfo.* Then the *AddressBook* generates the contact share request to the remote party specified in the input

argument *Targetcontacts* using the appropriate WAN protocol or the network address book associated with the user.

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 *ShareContacts()*

ErrorCode	errorDescription	Description
400-499	TBD	See Control clause in [1].
500-599	TBD	See Control clause in [1].
606	Action not Authorized	The CP does not have privileges to invoke this action.
714	Identity does not exist	The identity of the TS has not been assigned yet by the telephony service in the WAN side.
704	Invalid Filter	The filter element identified is invalid
705	Non Successful	The contact share request can not be executed due to internal errors

5.5.2 *FetchContactInfo()*

This action allows a TelCP to fetch the contact identified by the input argument *TargetContacts* from the network address book.

5.5.2.1 Arguments

Table 6 — Arguments for *FetchcontactInfo()*

Argument	Direction	relatedStateVariable
TargetContacts	*IN*	*A_ARG_TYPE_TargetContacts*
ShareInfo	*IN*	*A_ARG_TYPE_SharedInfo*

5.5.2.2 Argument Descriptions

The input argument *TargetContacts* contain the target contact identifiers for which the updated information needs to be fetched. The *TargetContacts* includes the reference to the contact in the local address book.

The input argument *ShareInfo* contain the list of contact information element to be fetched for e.g. name, email etc.This element contains the comma separated contact information element. This argument can contain empty string to share entire contact information from the local address book.

5.5.2.3 Service Requirements

The *AddresBook* service checks for the contacts in the local address book referenced by the input argument *TargetContacts*, and then identifies the network address book for the corresponding contact. The TS initiates the fetch information request on the network address book on the behalf of the TelCP where the target contact is actually stored.

Once the contact information is fetched from the network address book, then the local address book is modified with the latest contact information. The information to fetch from the network address book is identify by the input argument *ShareInfo*. Once the any updates to the address book is done it will be evented.

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 *FetchContactInfo()*

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.
703	Invalid Target Contacts	The destination contacts does not exist or contactInstanceID is an invalid value
704	Invalid Filter	The filter element identified is invalid
706	Error in Fetch operation	The TS can not fetch the contact information due to network error or WAN side error

5.5.3 *SharePCC()*

This action allows a TelCP to share the user personal contact information stored in the Personal Contact Card in the TS with the other contacts. The TelCP can share only the part for the PCC information for this TelCP can use the *ShareInfo* input argument to specify the required share elements.

5.5.3.1 Arguments

Table 8 — Arguments for *SharePCC()*

Argument	Direction	relatedStateVariable
TargetContacts	*IN*	*A_ARG_TYPE_TargetContacts*
ShareInfo	*IN*	*A_ARG_TYPE_SharedInfo*

5.5.3.2 Argument Descriptions

The input argument *TargetContacts* identifies the target contacts with whom the PCC information needs to be shared. The *TargetContacts* is an XML structure which identifies the contact information like contact communication address, name etc. The contact information can be stored in the local address book in the TS. This argument can contain reference to the entry in the address book in such cases.

The input argument *ShareInfo* contains the list of information element to be shared (e.g., name, email).This element contains the comma separated contact information elements. This argument can contain the empty string to share the entire PCC information.

5.5.3.3 Service Requirements

The *AddressBook* service shares the PCC information with the set of conacts identified in the input arguments. The target contact can belong to the network address book, in such cases the service should initiate the PCC share information through the network address book. Some contacts may not be associated with the network address book in such cases the TS can decide contact share mechanism based on the target contact information (e.g., target contact information with mobile number as the communication address, then *AddressBook* service can send the PCC information as a SMS). Note the mechanism used for sharing the PCC information is based on the implementation of the TS.

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 *SharePCC()*

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.
703	Invalid Target Contacts	The destination contacts does not exist or contactInstanceID is an invalid value
704	Invalid Filter	The filter element identified is invalid
707	Error in Share operation	The TS can not share the PCC information to certain contacts due to network error or any WAN side errors

5.5.4 *Accept()*

This action allows a TelCP to accept an incoming request for a contact. There are three different kinds of incoming requests (Contact Sharing, PCC sharing, and Contact Invitation). The Contact Sharing request means that remote party is willing to share set of contact information. The PCC information sharing request the remote party is requesting for the PCC information of the user. The Contact Invitation request is to share each others contact information. When the incoming request is evented by the *AddressBook* service via

IncomingRequest state variable, the TelCP can accept the particular request by invoking this action.

5.5.4.1 Arguments

Table 10 — Arguments for *Accept()*

Argument	Direction	relatedStateVariable
RequestID	*IN*	*A_ARG_TYPE_ReqID*

5.5.4.2 Argument Descriptions

The input argument *RequestID* contains unique identifier for the incoming request.

5.5.4.3 Service Requirements

The *AddressBook* service execute the incoming request which is identified by the *RequestID*.

In case of Contact Sharing request from the remote party, when the *AddressBook* service receives this action, then the service accept the contact share request and store the set of contacts into the local address book shared by the remote party.

In case of the PCC sharing request, the *AddressBook* service accept the PCC share request and share the PCC information of the user to the remote party.

In case of the Contact Invitation request, the *AddressBook* service accept the mutual contact information sharing. The service includes the remort party contact information into the local address book in the TS and share contact information of a user to the remote party.

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 *Accept()*

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 Request	The request identified by the ReqID does not exist.

5.5.5 *Reject()*

This action allows a TelCP to reject the incoming request identified by the unique identifier for the request in the input argument *RequestID*.

5.5.5.1 Arguments

Table 12 — Arguments for *Reject()*

Argument	Direction	relatedStateVariable
RequestID	*IN*	*A_ARG_TYPE_ReqID*

5.5.5.2 Argument Descriptions

The input argument *RequestID* contains unique identifier for the incoming request.

5.5.5.3 Service Requirements

The *AddressBook* service ignores the incoming request identified by the *RequestID*.

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 *Reject()*

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 Request	The request identified by the ReqID does not exist.

5.5.6 *ImportContacts()*

This action allows a TelCP to import the contacts from a particular network address book. The input argument *NetworkAddressBookID* identifies the network address book.

5.5.6.1 Arguments

Table 14 — Arguments for *ImportContacts()*

Argument	Direction	relatedStateVariable

Argument	Direction	relatedStateVariable
NetworkAddressBookID	*IN*	*A_ARG_TYPE_NetworkAddressBookID*

5.5.6.2 Argument Descriptions

The input argument *NetworkAddressBookId* gives an unique identifier for the network address book.

5.5.6.3 Service Requirements

When TS receives this action, then the TS validates the network address book. Once the TS successfully identifies the network address book, the TS fetches the contact information of the network address book and store the information into the local address book in the TS. If the local address book is associated with Address Book in the *Phone Data Model*, then the TS updates the Address Book in the *Phone Data Model*.

5.5.6.4 Control Point Requirements When Calling The Action

None.

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 *ImportContacts()*

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.
709	Invalid Network Address Book	The network address book identified does not exist.
710	Not successfull	The TS is not able to import the contact due to error in the WAN side.
711	Non authroized	The authorization failed.

5.5.7 *RetrieveIncomingRequests()*

This action allows a TelCP to retrieve all the active incoming requests.

5.5.7.1 Arguments

Table 16 — Arguments for *RetrieveIncomingRequests()*

Argument	Direction	relatedStateVariable

Argument	Direction	relatedStateVariable
ActiveIncomingRequests	*OUT*	*A_ARG_TYPE_IncomingRequests*

5.5.7.2 Argument Descriptions

The output argument *ActiveIncomingRequests* includes the information of the active incoming requests.

5.5.7.3 Service Requirements

None.

5.5.7.4 Control Point Requirements When Calling The Action

None.

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 *RetrieveIncomingRequests()*

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.

5.5.8 Error Code Summary

Table 18 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 18 — Error Code Summary

ErrorCode	errorDescription	Description
400-499	TBD	See Control clause in [1].
500-599	TBD	See Control clause in [1].
606	Action not Autohrized	The CP does not have privileges to invoke this action.
700		Reserved for future extensions.
702	Invalid Shared contact	The contacts to be shared does not exist or contactInstanceId is an invalid value.
703	Invalid Target Contacts	The destination contacts does not exist or contactInstanceID is an invalid value.
704	Invalid Filter	The filter element identified is invalid.

ErrorCode	errorDescription	Description
705	Non Successful	The contact share request can not be executed due to internal errors.
706	Error in Fetch operation	The TS can not fetch the contact information due to network error or WAN side error.
707	Error in Share operation	The TS can not share the PCC information to certain contacts due to network error or any WAN side errors.
708	Invalid Request	The request identified by the ReqID does not exist.
709	Invalid Network Address Book	The network address book identified does not exist.
710	Not successfull	The TS is not able to import the contact due to error in the WAN side.
711	Non authroized	The authorization failed.

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>ShareContacts</name>
         <argumentList>
            <argument>
               <name>SharedContacts</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_SharedContacts
               </relatedStateVariable>
            </argument>
            <argument>
               <name>SharedInfo</name>
               <direction>in</direction>
               <retval/>
               <relatedStateVariable>
                  A_ARG_TYPE_SharedInfo
               </relatedStateVariable>
            </argument>
            <argument>
               <name>TargetContacts</name>
               <direction>in</direction>
               <retval/>
               <relatedStateVariable>
                  A_ARG_TYPE_TargetContacts
               </relatedStateVariable>
            </argument>
         </argumentList>
```

```
    </action>
    <action>
       <name>FetchcontactInfo</name>
       <argumentList>
          <argument>
             <name>Targetcontacts</name>
             <direction>in</direction>
             <relatedStateVariable>
                A_ARG_TYPE_TargetContacts
             </relatedStateVariable>
          </argument>
          <argument>
             <name>ShareInfo</name>
             <direction>in</direction>
             <retval/>
             <relatedStateVariable>
                A_ARG_TYPE_SharedInfo
             </relatedStateVariable>
          </argument>
       </argumentList>
    </action>
    <action>
       <name>SharePCC</name>
       <argumentList>
          <argument>
             <name>TargetContacts</name>
             <direction>in</direction>
             <relatedStateVariable>
                A_ARG_TYPE_TargetContacts
             </relatedStateVariable>
          </argument>
          <argument>
             <name>ShareInfo</name>
             <direction>in</direction>
             <retval/>
             <relatedStateVariable>
                A_ARG_TYPE_SharedInfo
             </relatedStateVariable>
          </argument>
       </argumentList>
    </action>
    <action>
       <name>Accept</name>
       <argumentList>
          <argument>
             <name>RequestID</name>
             <direction>in</direction>
             <relatedStateVariable>
                A_ARG_TYPE_ReqID
             </relatedStateVariable>
          </argument>
       </argumentList>
    </action>
    <action>
       <name>Reject</name>
       <argumentList>
          <argument>
             <name>RequestID</name>
             <direction>in</direction>
             <relatedStateVariable>
```



```
                  A_ARG_TYPE_ReqID
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>ImportContacts</name>
         <argumentList>
            <argument>
               <name>NetworkAddressBookID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_NetworkAddressBookID
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>RetrieveIncomingRequests</name>
         <argumentList>
            <argument>
               <name>ActiveIncomingRequests</name>
               <direction>out</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_IncomingRequests
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
   </actionList>
   <serviceStateTable>
      <stateVariable sendEvents="yes">
         <name>IncomingRequest</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_SharedContacts</name>
         <dataType>String</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_SharedInfo</name>
         <dataType>String</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_TargetContacts</name>
         <dataType>String</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_ReqID</name>
         <dataType>String</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_NetworkAddressBookID</name>
         <dataType>String</dataType>
      </stateVariable>
   </serviceStateTable>
</scpd>
```

Annex A
(normative)

XML complex type *peerType*

A communication means the exchange of an information between two or more end entities. These end entities are herein referred as Peers. The Peer can be a caller of a phone call, recipient of an email message, or group of participants in a communication session, or a contact in an Address book.

In order to have a uniform representation of a Peer across all the services in the UPnP Telephony, the XML complex type peerType is defined. The same XML complex type can be reused by other UPnP Telephony services.

The complex type peerType contains the information to properly identify a contact and its communication address for e.g. a phone call needs a telephone number, an email message needs an email address etc. Along with the communication address it is also important to include additional information about the Peer for e.g. photo, location information of user etc. If TS supports the PhoneManagement profile, then the correspondence between the Peer element and either a contact or a group of contacts in the Address book is also included in the complex peerType element.

A.1 Using the *peerType* within XML Schemas

The complex type `peerType` can be used in the XML schemas by including the following statement:

```
<import
   namespace="urn:schemas-upnp-org:phone:peer"
   schemaLocation="http://www.upnp.org/schemas/phone/peer-v2.xsd"/>
```

where the `schemaLocation` refers to the last updated schema file for the *Peer.*

A.2 Description of fields of a `peerType` complex type

Clause A.2 gives a description of the elements defined in the peerType complex type.

```
<?xml version="1.0" encoding="UTF-8"?>
<peer:peer
   xsi:schemaLocation="urn:schemas-upnp-org:phone:peer
   http://www.upnp.org/schemas/phone/peer-v2.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:peer="urn:schemas-upnp-org:phone:peer">
   <peer:id>
      The identifier of the Peer (e.g., phone number, etc.)
   </peer:id>
   <peer:name>A user friendly name for the Peer</peer:name>
   <peer:contactInstanceId>
      The instance identifier for a contact referred by the Peer
   </peer:contactInstanceId>
   <peer:groupInstanceId>
      The instance identifier for a group referred by the Peer
   </peer:groupInstanceId>
   <peer:image type="URL"/"thumbnail">
      <peer:contentType>
         content type information for the image data
      </peer:contentType>
      <peer:contentTransferEncoding>
         Encoding information
```



```
      </peer:contentTransferEncoding>
      <peer:imageData>actual image information</peer:imageData>
   </peer:image>
   <peer:location type="map"/"coordinates">
      Either a URL points to the map information; URL for a the map image OR it
could include longitude and latitude information in ["LAT LON"] order.
   </peer:location>
</peer:peer>
```

`id`

Required, xsd:string. Indicates the communication address or the identifier for the Peer (e.g., a telephone number, an e-mail address, an identifier of a group of contacts, etc).

`name`

Allowed, xsd:string. Indicates a user friendly name for the Peer.

`contactInstanceId`

Allowed, xsd:unsignedInt. Is the instance identifier of the contact present in the *Phone Data Model's* Address Book for the referenced *Peer*. The value of the contactInstanceId is an unsigned integer. If there is no Instance in the *Address Book* for the referenced *Peer,* then the contactInstanceId value shall be 0 (no match with the list of contacts in the Address Book). If the PhoneManagement profile is not supported or the relationship between the *Address Book* and the *Peer* is not used by the service, then this element shall not be used. The contactInstanceId and groupInstanceId are mutually exclusive elements.

`groupInstanceId`

Allowed, xsd:unsignedInt. Is the instance identifier of a group present in the *Phone Data Model's Address Book* for the referenced Peer. The value of groupInstanceId is an unsigned integer. If there is no Instance in the *Address Book* for this referenced *Peer*, then the groupInstanceId value shall be 0 (no match with the list of groups in the *Address Book*). If the PhoneManagement profile is not supported or the relationship between the *Address Book* and the *Peer* is not used by the service, then this element shall not be used. The contactInstanceId and groupInstanceId are mutually exclusive elements.

`image`

Allowed,This element represents the image information for the contact. The image can be represented as an URL pointing to the image or small thumbunail image data information. This element may include following attributes and elements.

`type`

Allowed, xsd:string. This attribute indicates how image information of the contact is represented. The image information can be represented either an URL to the image or actual thumbnail image data. This attribute can have value either "URL" or "thumbnail".

`contentType`

Allowed, xsd:string. This element indicates MIME type information for the image as defined by [3]. This element should be present if type attribute is set to "thumbnail".

`contentTransferEncoding`

Allowed, xsd:string. This element indicates encoding mechanism for the image data as defines by [3]. This element should be present if *type* attribute is set to "thumbnail".

`ImageData`

Allowed, xsd:string. This element carries actual image information either as an URL to the image or actual image encoded information. If *type* attribute is set to "URL" then this element carries an URL to the image else *type* attribute is set to "Thumbnail" then it carries an actual encoded image information.

`location`

Allowed, xsd:string. This element carries the location information of the contact. The location information is represented eighter in longitude and latitude format or as an URL pointing to map information which includes location of the contact. The URL can also point to the map image. The information in this element is interpreted based on the *type* attribute of this element.

type

Allowed, xsd:string. This attribute indicates how to interpret the location information. If the location information is to be represented as a map information or as a map image, then this attribute is set to “map” and *location* element wll carry an URL of the map information. Else it is set to “coordinates” to represent the location information in [lattitude, Longitude] format.

any

Allowed. Attachment point for custom extensions.

A.3 peerType Schema

The following XML schema defines the peerType complex type.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:peer="urn:schemas-upnp-org:phone:peer"
targetNamespace="urn:schemas-upnp-org:phone:peer" elementFormDefault="qualified"
attributeFormDefault="qualified" version="1">
   <complexType name="peerType">
      <sequence>
         <element name="id" nillable="0">
            <annotation>
               <documentation>Id of the peer. The content depends on the context. For example
it can be a phone number, ad e-mail address and so on.</documentation>
            </annotation>
            <complexType>
               <simpleContent>
                  <extension base="string"/>
               </simpleContent>
            </complexType>
         </element>
         <element name="name" type="string" nillable="0" minOccurs="0">
            <annotation>
               <documentation>Textual name of the peer. In case the Phone Data Model is
supported, this element shall be the FormattedName in the address book.</documentation>
            </annotation>
         </element>
         <choice minOccurs="0">
            <element name="contactInstanceId" nillable="0">
               <annotation>
                  <documentation>The Instance Identifier of a Contact in the PDM address
book.</documentation>
               </annotation>
               <complexType>
                  <simpleContent>
                     <extension base="unsignedInt"/>
                  </simpleContent>
               </complexType>
            </element>
            <element name="groupInstanceId">
               <annotation>
                  <documentation>The Instance Identifier of a Group in the PDM address
book.</documentation>
               </annotation>
            </element>
         </choice>
         <element name="image" minOccurs="0">
           <complexType>
             <sequence>
               <element name="contentType" type="string" minOccurs="0"/>
               <element name="contentTransferEncoding" type="string" minOccurs="0"/>
               <element name="imageData" type="string" minOccurs="0"/>
             </sequence>
             <attribute name="type" use="optional">
               <simpleType>
                 <restriction base="string">
                   <enumeration value="URL"/>
                   <enumeration value="thumbnail"/>
                 </restriction>
               </simpleType>
             </attribute>
           </complexType>
         </element>
         <element name="location" minOccurs="0">
           <annotation>
```



```
            <documentation>Either a URL points to the map information; URL for a the map
image OR it could include longitude and latitude information in ["LAT LON"]
order.</documentation>
          </annotation>
          <complexType>
            <simpleContent>
              <extension base="string">
                <attribute name="type" use="optional" default="map">
              <simpleType>
                <restriction base="string">
                  <enumeration value="map"/>
                  <enumeration value="coordinates"/>
                </restriction>
              </simpleType>
            </attribute>
              </extension>
            </simpleContent>
          </complexType>
        </element>
        <any namespace="##other" minOccurs="0">
           <annotation>
              <documentation>Vendor defined extensions attachment point.</documentation>
           </annotation>
        </any>
      </sequence>
   </complexType>
</schema>
```

Annex B
(normative)

XML Schema

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

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:tns="urn:schemas-upnp-org:phone:AddressBook"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:peer="urn:schemas-upnp-org:phone:peer"
targetNamespace="urn:schemas-upnp-org:phone:AddressBook" elementFormDefault="unqualified"
attributeFormDefault="unqualified">
  <import namespace="urn:schemas-upnp-org:phone:peer"
schemaLocation="http://www.upnp.org/schemas/phone/peer-v2.xsd"/>
  <element name="incomingRequest">
    <annotation>
      <documentation>Corresponding state variable: IncomingRequest</documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="request" type="tns:request" minOccurs="0"/>
      </sequence>
    </complexType>
  </element>
  <element name="incomingRequests">
    <annotation>
      <documentation>Corresponding state variable: A_ARG_TYPE_IncomingRequests</documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="request" type="tns:request" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="sharedContacts">
    <annotation>
      <documentation>Corresponding state variable: A_ARG_TYPE_SharedContacts</documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="contact" type="peer:peerType" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="sharedInfo" type="string">
    <annotation>
      <documentation>Corresponding state variable: A_ARG_TYPE_SharedInfo</documentation>
    </annotation>
  </element>
  <element name="targetContacts">
    <annotation>
      <documentation>Corresponding state variable: A_ARG_TYPE_TargetContacts</documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="contact" type="peer:peerType" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="reqID" type="string">
    <annotation>
      <documentation>Corresponding state variable: A_ARG_TYPE_ReqID</documentation>
    </annotation>
  </element>
  <element name="networkAddressBookID" type="string">
    <annotation>
      <documentation>Corresponding state variable:
A_ARG_TYPE_NetworkAddressBookID</documentation>
    </annotation>
  </element>
  <complexType name="request">
    <sequence>
      <element name="type" type="string"/>
      <element name="reqID" type="string"/>
      <element name="fromcontactid" type="tns:fromcontactid"/>
```



```
      <element name="status" type="tns:status"/>
      <element name="note" type="string" minOccurs="0"/>
      <element name="sharedinformationURL" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="fromcontactid">
    <sequence>
      <element name="id" type="peer:peerType"/>
    </sequence>
  </complexType>
  <simpleType name="status">
    <restriction base="string">
      <enumeration value="Pending"/>
      <enumeration value="Accepted"/>
      <enumeration value="Rejected"/>
    </restriction>
  </simpleType>
</schema>
```

Annex C
(informative)

Theory of Operation

C.1 Sharing set of contacts with WAN side User

A TelCP can share a set of contact information to remote party using the interface provided by the network address book or messaging service.

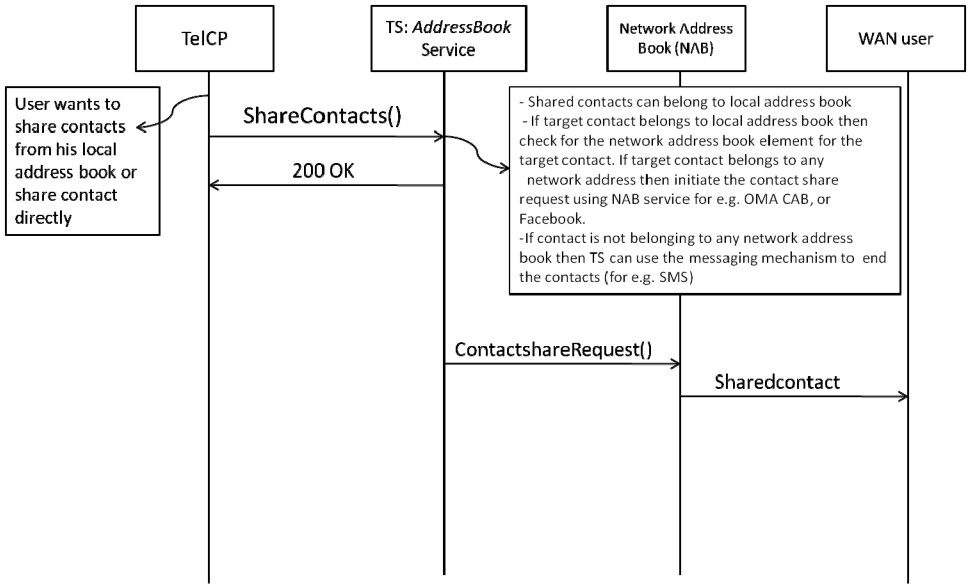

Figure C.1 — Contact Share with WAN User

- A TelCP wants to share set of contact information with a WAN user. A TelCP invokes the *ShareContacts()* action to the *AddressBook* service.

- The *AddressBook* service validates the action and responds to TelCP. The *AddressBook* service also initiates a contact share request on WAN side for each target contact specified in the action. If target contact is belogging to network address book which support contact share, then the *AddressBook* should initiate the contact share request via network address book. If target contact is not associated with any network address book then, the *AddressBook* should initiate the any other messaging service and share the contact information for e.g. SMS, or Email.

C.2 Handling of incoming contact share request from WAN side

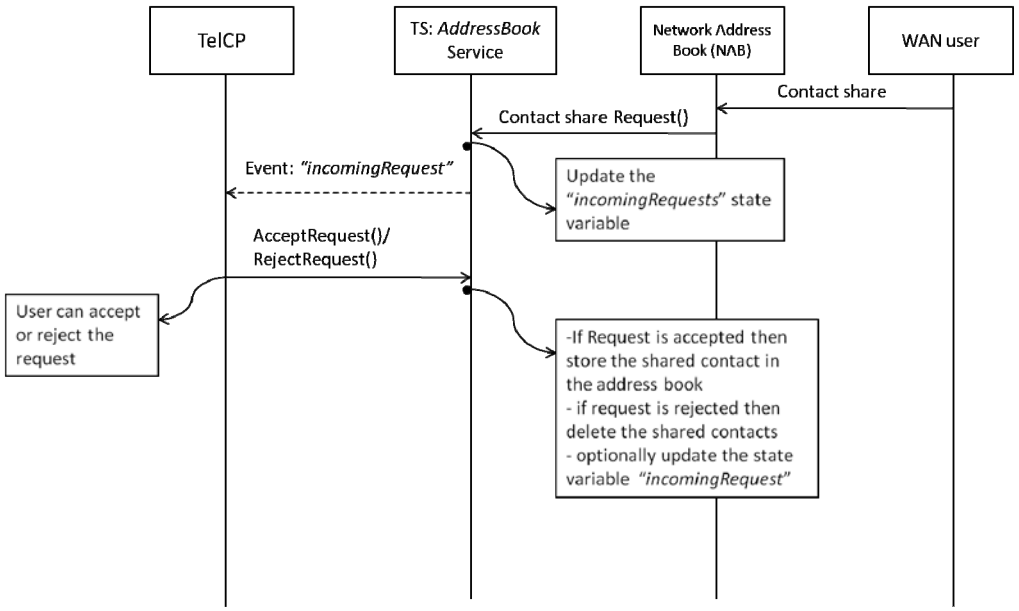

Figure C.2 — Handling of incoming contact share request

- When the *AddressBook* service receives a contact share request from the WAN side, the service capture the contact share request into the state variable "*IncomingRequest*".

- The *AddressBook* service events out the *IncomingRequest* state variable to the subscribed TelCP(s).

- TelCP can accept or reject the incoming request by invoking the *Accept()* or *Reject()* action.

- If the TelCP accepts the contact share request, then the *AddressBook* service stores the shared contact information into the local address book of the user.

C.3 Fetching updated contact information from the Network Address Book

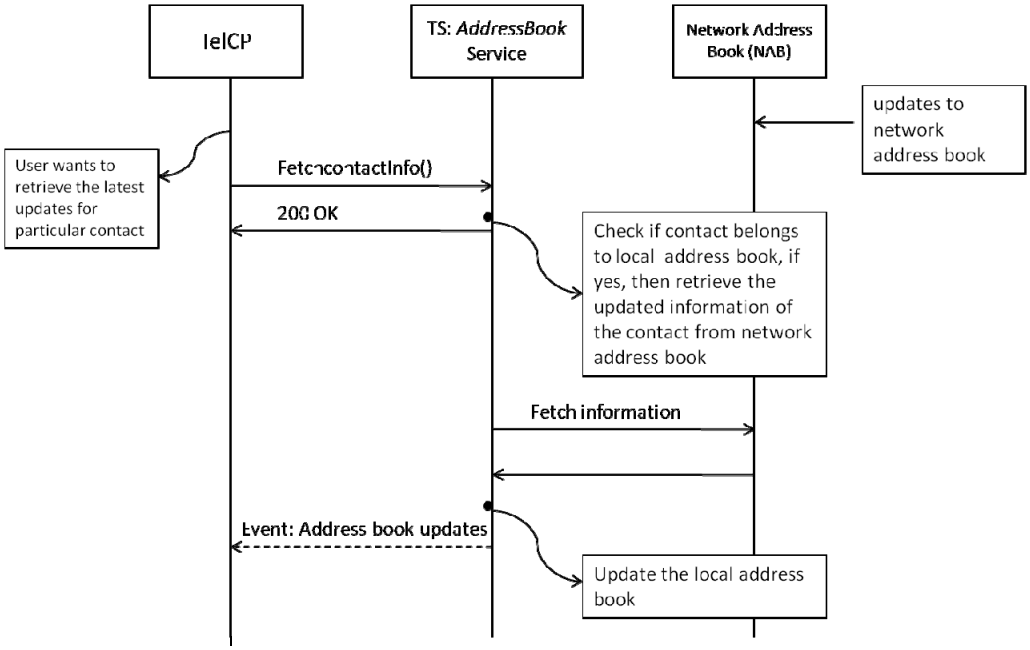

Figure C.3 — Fetching contact information from network address book

- The TelCP invokes the *Fetchcontactinfo()* action to retrieve the contact information for a particular contact from the network address book.

- The *AddressBook* service fetch the contact information from the network address book where this contact belongs. The fetched contact information will be updated into the local address book. If a contact does not belong to any network address book then the *AddressBook* service returns an error to the TelCP.

C.4 Sharing Personal Contact Information (PCC) with WAN user

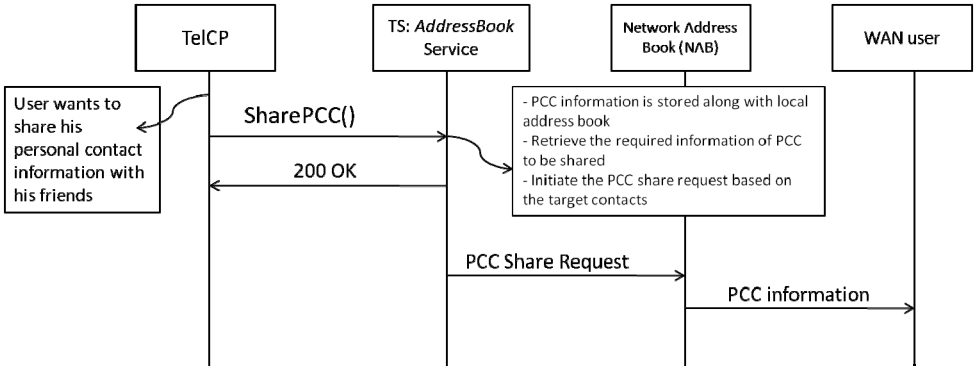

Figure C.4 — Sharing PCC information

- TelCP invokes the *SharePCC()* action to share the personal contact information (PCC) which is stored in the local address book.

- The *AddressBook* service retrieves the PCC information from the local address book. If the target contact is associated with a particular network address book, then the

AddressBook service will be initiate a PCC share request trough the network address book.

C.5 Contact Invitation Request handling

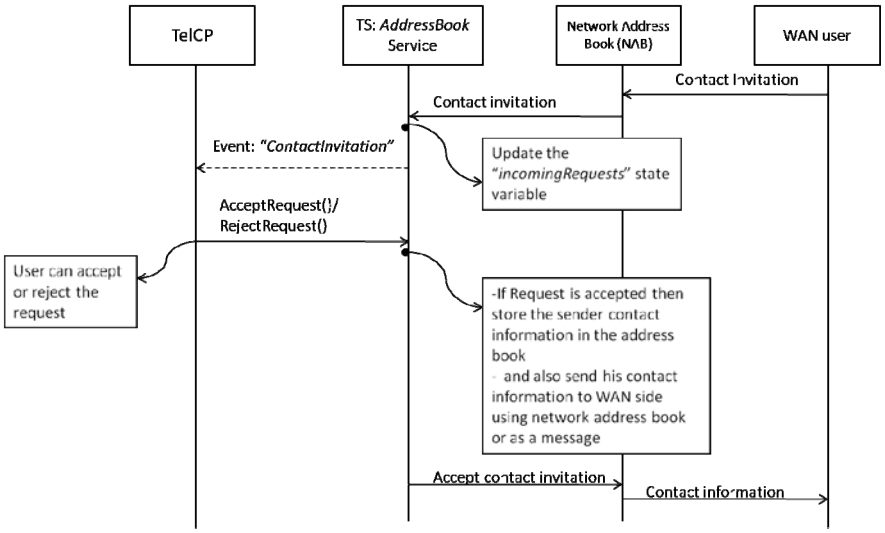

Figure C.5 — Contact Invitation request handling

- When the *AddressBook* service receives a Contact Invitation request from the WAN side, the service captures the Contact Invitation request into the state variable "*IncomingRequest*".

- The *AddressBook* service events out the *IncomingRequest* state variable to the subscribed TelCP(s).

- A TelCP can accept or reject the incoming request by invoking the *Accept()* or *Reject()* action.

If a TelCP accepts the Contact Invitation request, then the *AddressBook* service stores the contact information into the local address book of the user and also sends the PCC information or contact information to the WAN user.

Annex D
(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.

[12] – OMA Converged Address Book v1.0, Open Mobile Alliance, October 19, 2010. Available at: http://www.openmobilealliance.com/Technical/release_program/cab_v1_0.aspx.

ICS 35.200

Price based on 32 pages