ETSI TS 129 199-4 V9.0.0 (2010-01)

Technical Specification

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Open Service Access (OSA);
Parlay X web services;
Part 4: Short messaging
(3GPP TS 29.199-04 version 9.0.0 Release 9)

Reference
RTS/TSGC-0029199-04v900

Keywords
GSM, LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

3GPP acknowledges the contribution of the Parlay X Web Services specifications from The Parlay Group. The Parlay Group is pleased to see 3GPP acknowledge and publish the present document, and the Parlay Group looks forward to working with the 3GPP community to improve future versions of the present document.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part 4 of a multi-part deliverable covering the 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Open Service Access (OSA); Parlay X Web Services, as identified below:

Part 1: "Common"
Part 2: "Third party call"
Part 3: "Call Notification"
Part 4: "Short Messaging"
Part 5: "Multimedia Messaging"
Part 6: "Payment"
Part 7: "Account management"
Part 8: "Terminal Status"
Part 9: "Terminal location"
Part 10: "Call handling"
Part 11: "Audio call"
Part 12: "Multimedia conference"
Part 13: "Address list management"
Part 14: "Presence"
Part 15: "Message Broadcast"
Part 16: "Geocoding"
Part 17: "Application driven Quality of Service (QoS)"
Part 18: "Device Capabilities and Configuration"
Part 19: "Multimedia streaming control"
Part 20: "Multimedia multicast session management"
Part 21: "Content management"
Part 22: "Policy"

1 Scope

The present document is Part 4 of the Stage 3 Parlay X Web Services specification for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the OSA APIs. The concepts and the functional architecture for the OSA are contained in 3GPP TS 23.198 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

The present document specifies the Short Messaging Web Service aspects of the interface. All aspects of the Short Messaging Web Service are defined here, these being:

- Name spaces.
- Sequence diagrams.
- Data definitions.
- Interface specification plus detailed method descriptions.
- Fault definitions.
- Service policies.
- WSDL description of the interfaces.

The present document has been defined jointly between 3GPP TSG CT WG5, ETSI TISPAN and The Parlay Group.

Maintenance of up to 3GPP Rel-8 and new OSA Stage 1, 2 and 3 work beyond Rel-9 was moved to OMA in June 2008.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.127: "Service Requirement for the Open Services Access (OSA); Stage 1".

[3] 3GPP TS 23.198: "Open Service Access (OSA); Stage 2".

[4] 3GPP TS 22.101: "Service aspects; Service principles".

[5] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

NOTE: Available at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[6] 3GPP TS 29.199-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common".

[7] 3GPP TS 23.040: "Technical realization of Short Message Service (SMS)".

[8] RFC2822: "Internet Message Format".

NOTE: Available at htpp://www.ietf.org/rfc/rfc2822.txt

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 29.199-1 [6] apply.

Additionally the following definition is needed:

Whitespace: see definition for CFWS as defined in RFC2822 [8].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TS 29.199-1 [6] and the following apply:

SMS	Short Message Service
SMS-C	Short Message Service - Center

4 Detailed service description

Currently, in order to programmatically receive and send SMS it is necessary to write applications using specific protocols to access SMS functions provided by network elements (e.g. SMS-C). This approach requires a high degree of network expertise. Alternatively it is possible to use the Parlay/OSA approach, invoking standard interfaces (e.g. User Interaction or Messaging Service Interfaces) to gain access to SMS capabilities, but these interfaces are usually perceived to be quite complex by IT application developers. Developers must have advanced telecommunication skills to use OSA interfaces.

In this clause is described a Parlay X Web Service, for sending and receiving SMS messages. The overall scope of this Web Service is to provide to application developers primitives to handle SMS in a simple way. In fact, using the SMS Web Service, application developers can invoke SMS functions without specific Telco knowledge.

ShortMessaging provides operations (see clause 8.1, Send SMS API) for sending a SMS message to the network and a polling mechanism for monitoring the delivery status of a sent SMS message. It also provides an asynchronous notification mechanism for delivery status (see clause 8.2, SmsNotification API). In addition, a mechanism is provided to start and stop the notification of delivery receipts (see clause 8.4, SmsNotificationManager API).

ShortMessaging also allows an application to receive SMS messages. Both a polling (see clause 8.3, ReceiveSMS API) and an asynchronous notification mechanism (see clause 8.2, SMSNotification API and clause 8.4, SmsNotificationManager API) are available.

Figure 1 shows a scenario using the SMS Web Service to send an SMS message from an application. The application invokes a Web Service to retrieve a weather forecast for a subscriber (1) and (2) and a Parlay X Interface (3) to use the SMS Web Service operations (i.e. to send an SMS). After invocation, the SMS Web Service invokes a Parlay API method (4) using the Parlay/OSA SCS (Generic User Interaction) interface. This SCS handles the invocation and sends an UCP operation (5) to an SMS-C. Subsequently the weather forecast is delivered (6) to the subscriber.

In an alternative scenario, the Parlay API interaction involving steps (4) and (5) could be replaced with a direct interaction between the SMS Web Service and the Mobile network.

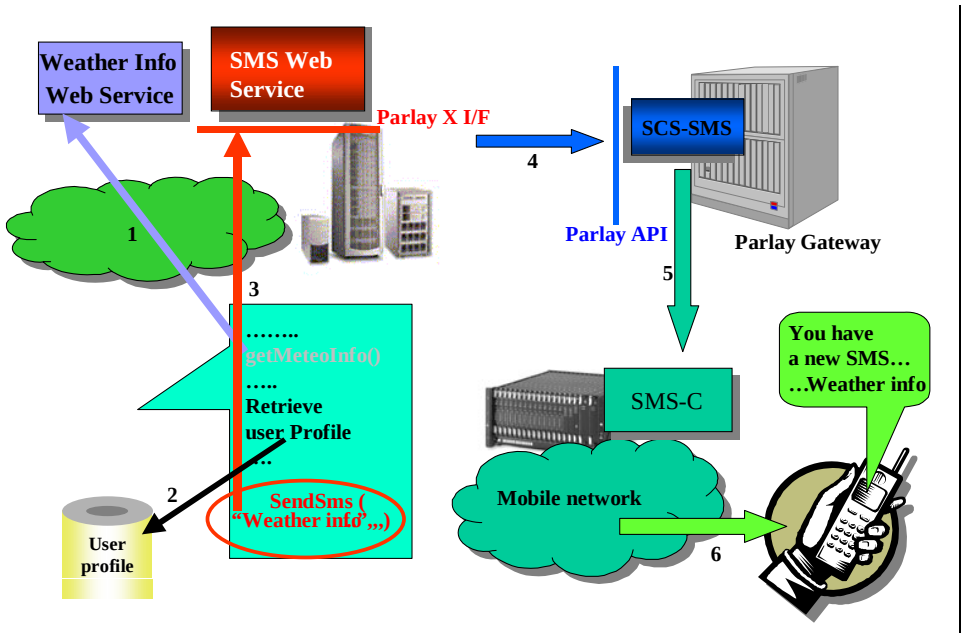

Figure 1: Send SMS Scenario

Figure 2 shows a scenario using the SMS Web Service to deliver a received SMS message to an application. The application receives a Parlay X Web Service invocation for an SMS sent by a subscriber (1) and (2). The SMS message contains the e-mail address of the person the user wishes to call. The application invokes a Parlay X Interface (3) to the Third Party Call Web Service in order to initiate the call (4).

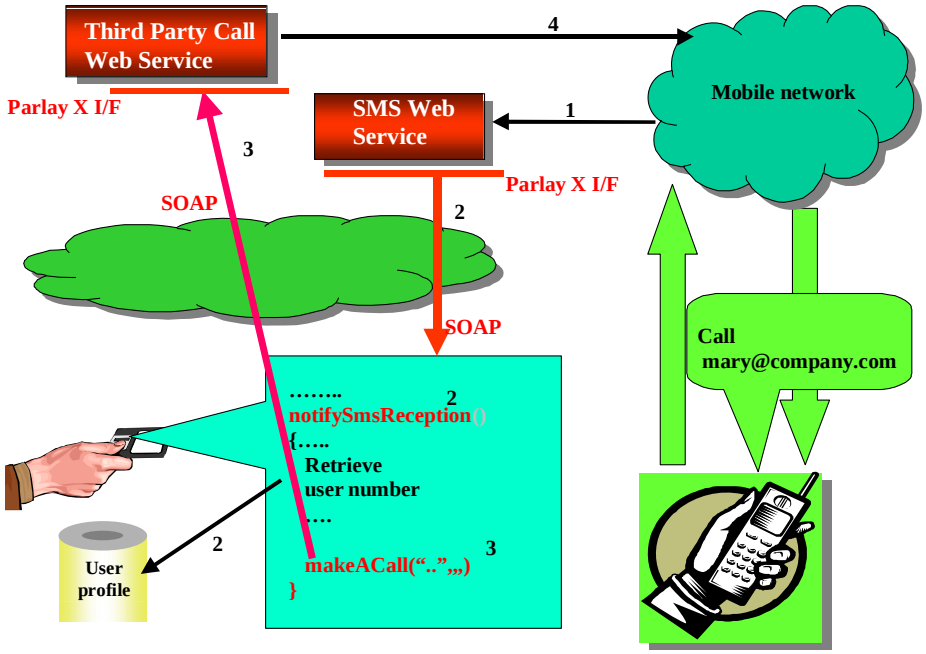

Figure 2: Receive SMS Scenario

5 Namespaces

The SendSms interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/sms/send/v4_0

The ReceiveSms interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/sms/receive/v4_0

The SmsNotification interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/sms/notification/v4_0

The SmsNotificationManager interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/sms/notification_manager/v4_0

The data types are defined in the namespace:

http://www.csapi.org/schema/parlayx/sms/v4_0

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [5]. The use of the name 'xsd' is not semantically significant.

6 Sequence Diagrams

6.1 Send SMS and report status

Sending SMS message from Web portals is a common capability offered by Service Providers. This sequence diagram shows a portal providing this service.

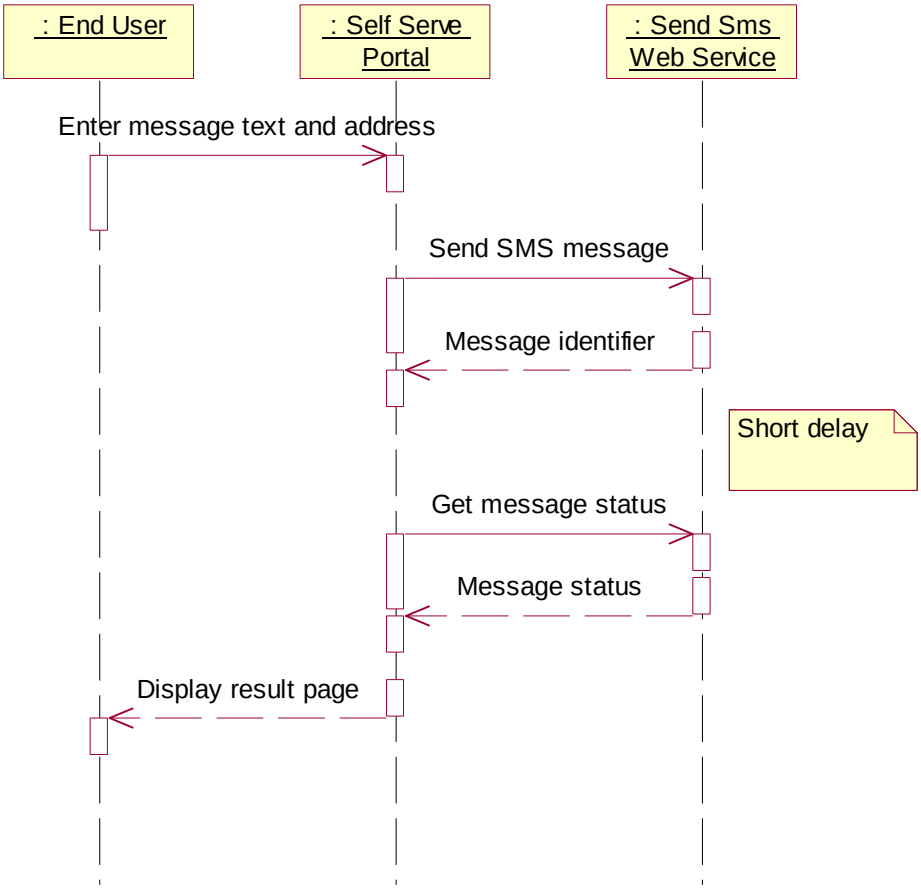

Figure 3

7 XML Schema data type definition

7.1 DeliveryStatus enumeration

List of delivery status values.

Enumeration	Description
DeliveredToNetwork	Successful delivery to network
DeliveryUncertain	Delivery status unknown: e.g. because it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.
MessageWaiting	The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.
DeliveredToTerminal	Successful delivered to Terminal
DeliveryNotificationNotSupported	Unable to provide delivery receipt notification. NotifySMSDeliveryReceipt function will provide 'DeliveryNotificationNotSupported' to indicate that delivery receipt for the specified address in a SendSMSRequest is not supported.

7.2 SmsFormat enumeration

List of SMS format values.

Enumeration	Description
Ems	Enhanced Messaging Service, standardized in TS 23.040 [7], which defines a logo/ringtone format
SmartMessaging™	Defines a logo/ringtone format

7.3 DeliveryInformation structure

Delivery status information.

Element name	Element type	Optional	Description
Address	xsd:anyURI	No	It indicates the destination address to which the notification is related.
DeliveryStatus	DeliveryStatus	No	Indicates the delivery result for the destination address.
Description	xsd:string	Yes	Used together with delivery status (e.g.DeliveryImpossible) to provide additional information.

7.4 SmsMessage structure

SMS message information. The SenderAddress is the address from which the message was actually sent, which may or may not match the senderName value provided in the SendSms operation.

Element name	Element type	Optional	Description
Message	xsd:string	No	Text received in SMS
SenderAddress	xsd:anyURI	No	It indicates address sending the SMS
SmsServiceActivation Number	xsd:anyURI	No	Number associated with the invoked Message service, i.e. the destination address used to send the message
DateTime	xsd:dateTime	Yes	Time when message was received by operator

7.5 ScheduledDeliveryStatus enumeration

List of scheduled SMS delivery status values.

Enumeration	Description
Scheduled	The Message has been scheduled, the scheduled time has not started.
NotSent	Message could not be sent before end of scheduled time.
Sent	The Message has been sent within the scheduled time.
Cancelled	Message has been cancelled. Some messages may have been sent.
PartiallySent	Message is sent to some, but not to all the recipients.
StatusUnavailable	Unable to provide delivery information.

7.6 ScheduledDeliveryInformation structure

Scheduled delivery information.

Element name	Element type	Optional	Description
DeliveryStatus	ScheduledDeliveryStatus	No	Indicates the delivery result for the destination address.
NumberOf MessagesSent	xsd:int	Yes	If applicable, the number of messages already sent.

8 Web Service interface definition

8.1 Interface: SendSms

This interface defines operations to send and schedule various types of Short Messages and to subsequently poll for delivery status. The Short Message types are:

- SMS message, as described in 0 and 0
- SMS logo, as described in 0 and 0
- SMS ringtone, as described in 0. and 0

The send operations for the Short Message types are similar. A description of the common message parts follows:

- **addresses** specifies the destination address or address set for the Short Message. It may include group URIs as defined in the Address List Management specification. If groups are not supported, a **PolicyException** (POL0006) will be returned to the application.
- **senderName** is optional and specifies the sender"s name: i.e. the string that is displayed on the user's terminal as the originator of the message
- **charging** specifies the charging information
- **receiptRequest** is optional and is specified when the application requires to receive notification of the status of the SMS delivery. It is a **SimpleReference** structure that indicates the application endpoint, interface used for notification of delivery receipt and a correlator that uniquely identifies the sending request.
    - If the notification mechanism is not supported by a network, a **ServiceException** (SVC0283) will be returned to the application and the message will not be sent to the addresses specified.
    - The correlator provided in the r**eceiptRequest** must be unique for this Web Service and application at the time the notification is initiated, otherwise a **ServiceException** (SVC0005) will be returned to the application.
    - Notification to the application is done by invoking the **notifySmsDeliveryReceipt** operation at the endpoint specified in **receiptRequest**.
    - This optional message part is not used (or will be overridden) in case the **startDeliveryReceiptNotification** operation is used when the application requires to receive delivery receipt notifications. This is to avoid overlapping criteria.
- **requestIdentifier** is specified in the response message associated with each send operation. The application can use it to invoke the **getSmsDeliveryStatus** operation to poll for the delivery status.

8.1.1 Operation: SendSms

The application invokes the **sendSms** operation to send an SMS message, specified by the String **message**.

If **message** is longer than the maximum supported length, the message content will be sent as several concatenated short messages.

8.1.1.1 Input message: SendSmsRequest

Part name	Part type	Optional	Description
Addresses	xsd:anyURI [1..unbounded]	No	Addresses to which the SMS will be sent
SenderName	xsd:string	Yes	If present, it indicates the SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Yes	Charge to apply to this message
Message	xsd:string	No	Text to be sent in SMS
ReceiptRequest	common:SimpleReference	Yes	It defines the application endpoint, interfaceName and correlator that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible. It is not used (or will be overridden) in case the startDeliveryReceiptNotification operation is used.

8.1.1.2 Output message : SendSmsResponse

Part name	Part type	Optional	Description
result	xsd:string	No	It identifies a specific SMS delivery request

8.1.1.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0280 - Message too long.
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.
- POL0012 - Too many description entries specified
- POL0013 – Addresses duplication

8.1.2 Operation: SendSmsLogo

The application invokes the **sendSmsLogo** operation to send an SMS logo, specified by the byte array **image.**

8.1.2.1 Input message: SendSmsLogoRequest

Part name	Part type	Optional	Description
Addresses	xsd:anyURI [1..unbounded]	No	Addresses to which the SMS logo will be sent
SenderName	xsd:string	Yes	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Yes	Charge to apply to this message
Image	xsd:base64Binary	No	The image in jpeg, gif or png format. The image will be scaled to the proper format
SmsFormat	SmsFormat	No	Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'
ReceiptRequest	common:SimpleReference	Yes	It defines the application endpoint, interfaceName and correlator that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible

8.1.2.2 Output message: SendSmsLogoResponse

Part name	Part type	Optional	Description
result	String	No	It identifies a specific SMS delivery request

8.1.2.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.
- POL0012 - Too many description entries specified
- POL0013 – Addresses duplication

8.1.3 Operation: SendSmsRingtone

The application invokes the **sendSmsRingtone** operation to send an SMS ringtone, specified by the String **ringtone** (in RTX format).

Depending on the length of the ringtone, it may be sent as several concatenated short messages.

NOTE: In the RTX Ringtone Specification,an RTX file is a text file, containing the ringtone name, a control subclause and a subclause containing a comma separated sequence of ring tone commands.

8.1.3.1 Input message: SendSmsRingtoneRequest

Part name	Part type	Optional	Description
Addresses	xsd:anyURI [1..unbounded]	No	Addresses to which the SMS ringtone will be sent
SenderName	xsd:string	Yes	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Yes	Charge to apply to this message
Ringtone	xsd:string	No	The ringtone in RTX format (see note above). (http://www.logomanager.co.uk/help/Edit/RTX.html)
SmsFormat	SmsFormat	No	Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'
ReceiptRequest	common:SimpleReference	Yes	It defines the application endpoint, interfaceName and correlator that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible

8.1.3.2 Output message: SendSmsRingtoneResponse

Part name	Part type	Optional	Description
result	xsd:string	No	It identifies a specific SMS delivery request

8.1.3.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.
- POL0012 - Too many description entries specified
- POL0013 – Addresses duplication

8.1.4 Operation: GetSmsDeliveryStatus

The application invokes the **getSmsDeliveryStatus** operation to request the status of a previous SMS delivery request identified by **requestIdentifier**. The information on the status is returned in **deliveryStatus**, which is an array of status related to the request identified by **requestIdentifier**. The status is identified by a couplet indicating a user address and the associated delivery status. This method can be invoked multiple times by the application even if the status has reached a final value. However, after the status has reached a final value, status information will be available only for a limited period of time as defined by a service policy. The following five different SMS delivery status values have been identified:

- 'DeliveredToNetwork': in case of concatenated messages, only when all the SMS-parts have been successfully delivered to the network.
- 'DeliveryUncertain': e.g. because it was handed off to another network.
- 'DeliveryImpossible': unsuccessful delivery; the message could not be delivered before it expired.
- 'MessageWaiting': the message is still queued for delivery.
- 'DeliveredToTerminal': in case of concatenated messages, only when all the SMS-parts have been successfully delivered to the terminal.

8.1.4.1 Input message: GetSmsDeliveryStatusRequest

Part name	Part type	Optional	Description
RequestIdentifier	xsd:string	No	It identifies a specific SMS delivery request

8.1.4.2 Output message : GetSmsDeliveryStatusResponse

Part name	Part type	Optional	Description
result	DeliveryInformation [0..unbounded]	Yes	It lists the variations on the delivery status of the SMS. Possible values are: • DeliveredToNetwork • DeliveryUncertain • DeliveryImpossible • MessageWaiting • DeliveredToTerminal

8.1.4.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.

PolicyException from [6]:

- POL0001 - Policy error.
- POL0010 – Retention time interval expired

8.1.5 Operation: ScheduleSms

The application invokes the **scheduleSms** operation to schedule sending of an SMS message, specified by the String **message.**

8.1.5.1 Input message: ScheduleSmsRequest

Part name	Part type	Optional	Description
Addresses	xsd:anyURI [1..unbounded]	No	Addresses to which the SMS will be sent
SenderName	xsd:string	Yes	If present, it indicates the SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Yes	Charge to apply to this message
Message	xsd:string	No	Text to be sent in SMS
StartTime	xsd:dateTime	No	Specifies the time to start sending out the scheduled SMS.
StopTime	xsd:dateTime	No	Specifies the time to stop sending out the SMS. Any message not sent before StopTime will not be sent.

8.1.5.2 Output message : ScheduleSmsResponse

Part name	Part type	Optional	Description
result	xsd:string	No	It identifies a specific SMS schedule request

8.1.5.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0280 - Message too long.

PolicyException from [6]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.
- POL0012 - Too many description entries specified
- POL0013 – Addresses duplication

8.1.6 Operation: ScheduleSmsLogo

The application invokes the **ScheduleSmsLogo** operation to schedule sending of an SMS logo, specified by the byte array **image.**

8.1.6.1 Input message: ScheduleSmsLogoRequest

Part name	Part type	Optional	Description
Addresses	xsd:anyURI [1..unbounded]	No	Addresses to which the SMS logo will be sent
SenderName	xsd:string	Yes	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Yes	Charge to apply to this message
Image	xsd:base64Binary	No	The image in jpeg, gif or png format. The image will be scaled to the proper format
SmsFormat	SmsFormat	No	Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'
StartTime	xsd:dateTime	No	Specifies the time to start sending out the scheduled SMS.
StopTime	xsd:dateTime	No	Specifies the time to stop sending out the SMS. Any message not sent before StopTime will not be sent.

8.1.6.2 Output message: ScheduleSmsLogoResponse

Part name	Part type	Optional	Description
result	xsd:string	No	It identifies a specific SMS schedule request

8.1.6.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.

PolicyException from [6]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.
- POL0012 - Too many description entries specified
- POL0013 – Addresses duplication

8.1.7 Operation: ScheduleSmsRingtone

The application invokes the **ScheduleSmsRingtone** operation to schedule sending of an SMS ringtone, specified by the String **ringtone** (in RTX format).

Depending on the length of the ringtone, it may be sent as several concatenated short messages.

NOTE: In the RTX Ringtone Specification,an RTX file is a text file, containing the ringtone name, a control subclause and a subclause containing a comma separated sequence of ring tone commands.

8.1.7.1 Input message: ScheduleSmsRingtoneRequest

Part name	Part type	Optional	Description
Addresses	xsd:anyURI [1..unbounded]	No	Addresses to which the SMS ringtone will be sent
SenderName	xsd:string	Yes	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message
Charging	common:ChargingInformation	Yes	Charge to apply to this message
Ringtone	xsd:string	No	The ringtone in RTX format (see note above). (http://www.logomanager.co.uk/help/Edit/RTX.html)
SmsFormat	SmsFormat	No	Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'
StartTime	xsd:dateTime	No	Specifies the time to start sending out the scheduled SMS.
StopTime	xsd:dateTime	No	Specifies the time to stop sending out the SMS. Any message not sent before StopTime will not be sent.

8.1.7.2 Output message: ScheduleSmsRingtoneResponse

Part name	Part type	Optional	Description
result	xsd:string	No	It identifies a specific SMS schedule request

8.1.7.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.
- SVC0004 - No valid addresses.
- SVC0006 - Invalid group.
- SVC0281 - Unrecognized data format.

PolicyException from [6]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.
- POL0012 - Too many description entries specified
- POL0013 – Addresses duplication

8.1.8 Operation: CancelScheduledSms

The invocation of cancelScheduledSmsRequest cancels the previously scheduled SMS request identified by requestIdentifier. If the period scheduled for sending SMS has started, some or all of the messages may have been sent.

8.1.8.1 Input message: CancelScheduledSmsRequest

Part name	Part type	Optional	Description
RequestIdentifier	xsd:string	No	It identifies a specific SMS schedule request

8.1.8.2 Output message : CancelScheduledSmsResponse

Part name	Part type	Optional	Description
None			

8.1.8.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.

PolicyException from [6]:

- POL0001 - Policy error.

8.1.9 Operation: GetScheduledSmsStatus

Gets the schedule and status of a scheduled SMS message.

8.1.9.1 Input message: GetScheduledSmsStatusRequest

Part name	Part type	Optional	Description
RequestIdentifier	xsd:string	No	It identifies a specific SMS schedule request

8.1.9.2 Output message : GetScheduledSmsStatusResponse

Part name	Part type	Optional	Description
result	ScheduledDeliveryInformation	No	Indicates the delivery result for the destination addresses and, if applicable, the number of messages already sent.

8.1.9.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.

PolicyException from [6]:

- POL0001 - Policy error.
- POL0010 – Retention time interval expired

8.2 Interface: SmsNotification

SmsNotification is the application side notification interface to which short messages are delivered.

8.2.1 Operation: NotifySmsReception

The notification is used to send a short message to the application. The notification will occur only if the SMS fulfils the criteria specified when starting the SMS notification.

The **notifySmsReception** method must be implemented by a Web Service at the ***application side***. It will be invoked by the Parlay X server to notify the application of the reception of an SMS. The notification will occur if and only if the SMS received fulfils the criteria specified in a provisioning step, identified by the **correlator**. The criteria must at least include an **smsServiceActivationNumber**, i.e. the SMS destination address that can be "monitored" by the application. The parameter **senderAddress** contains the address of the sender. The application can apply the appropriate service logic to process the SMS.

8.2.1.1 Input message: NotifySmsReceptionRequest

Part name	Part type	Optional	Description
correlator	xsd:string	No	Correlator provided in request to set up this notification
Message	SmsMessage	No	Message received

8.2.1.2 Output message : NotifySmsReceptionResponse

Part name	Part type	Optional	Description
None			

8.2.1.3 Referenced faults

None.

8.2.2 Operation: NotifySmsDeliveryReceipt

The **notifySmsDeliveryReceipt** method must be implemented by a Web Service at the ***application side*** if it requires notification of SMSdelivery receipt. It will be invoked by the Parlay X server to notify the application when a SMS sent by an application has been delivered to the terminal of the recipient or if delivery is impossible. The notification will occur if and only if the status of the sent SMS is "DeliveredToTerminal" or "DeliveryImpossible" and the application has specified interest in notification when sending an SMS message using one of the following mutually exclusive mechanisms:

- by specifying the optional receiptRequest parameter. The correlator returned corresponds to the identifier specified by the application in the **receiptRequest** of the original **sendSMS** request
- by invoking the **startDeliveryReceiptNotification** operation requesting to receive delivery receipt notifications. The correlator returned corresponds to the identifier specified by the application in the **reference** of the original **startDeliveryReceiptNotification** request.

When a SMS message is sent to multiple addresses, the notification from the server will send notification for each terminal as and when a SMS message is delivered to a terminal.

The following three different SMS delivery status will be returned in NotifySMSDeliveryReceiptResponse:

- 'DeliveryImpossible': unsuccessful delivery; the message could not be delivered before it expired.
- 'DeliveredToTerminal': in case of concatenated messages, only when all the SMS-parts have been successfully delivered to the terminal.
- "DeliveredNotificationNotSupported" - If notification is supported by the network but it does not support delivery receipt for one or more addresses specified in the **sendSMS** message. The service will send this status for those addresses.

8.2.2.1 Input message: NotifySmsDeliveryReceiptRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	The identifier defining the original SendRequest. This correlator was passed by the application during the SendSMS request
DeliveryStatus	DeliveryInformation	No	It lists the variations on the delivery status of the SMS to a terminal. Possible values are: • DeliveryImpossible • DeliveredToTerminal • DeliveryNotificationNotSupported

8.2.2.2 Output message: NotifySmsDeliveryReceiptResponse

Part name	Part type	Optional	Description
None			

8.2.2.3 Referenced faults

None.

8.3 Interface: ReceiveSms

8.3.1 Operation: GetReceivedSms

The invocation of **getReceivedSms** retrieves all the SMS messages received that fulfil the criteria identified by **registrationIdentifier**. The method returns only the list of SMS messages received since the previous invocation of the same method, i.e. each time the method is executed the messages returned are removed from the server. Moreover, each SMS message will be automatically removed from the server after a maximum time interval as defined by a service policy.

The received SMS messages are returned in the **getReceivedSmsResponse** message. An SMS message is identified by a structure indicating the sender of the SMS message and the content.

8.3.1.1 Input message: GetReceivedSmsRequest

Part name	Part type	Optional	Description
RegistrationIdentifier	xsd:string	No	Identifies the provisioning step that enables the application to receive notification of SMS reception according to specified criteria

8.3.1.2 Output message : GetReceivedSmsResponse

Part name	Part type	Optional	Description
result	SmsMessage [0..unbounded]	Yes	It lists the received SMS since last invocation

8.3.1.3 Referenced faults

ServiceException from [6]:

- SVC0001 - Service error.
- SVC0002 - Invalid input value.

PolicyException from [6]:

- POL0001 - Policy error.
- POL0010 – Retention time interval expired

8.4 Interface: SmsNotificationManager

The short message notification manager enables applications to set up and tear down notifications for short messages, online.

8.4.1 Operation: StartSmsNotification

Start notifications to the application for a given SMS Service activation number and criteria.

The SMS Service activation number is an Address Data item e.g. Shortcode, as defined in 3GPP TS 29.199-1 [6].

The correlator provided in the reference must be unique for the application Web Service at the time the notification is initiated, otherwise a ServiceException (SVC0005) will be returned to the application..

If specified, criteria will be used to filter messages that are to be delivered to an application. If criteria are not provided, or is an empty string, then all messages for the SmsServiceActivationNumber will be delivered to the application. The SmsServiceActivationNumber and criteria combination must be unique. If a criteria overlaps then SVC0008 will be returned to the application and the notification will not be set up. Note that the use of criteria will allow different notification endpoints to receive notifications for the same SmsServiceActivationNumber. The combination of SmsServiceActivationNumber and criteria must be unique, so that a notification will be delivered to only one notification endpoint. If no match is found, the message will not be delivered to the application.

8.4.1.1 Input message: StartSmsNotificationRequest

Part name	Part type	Optional	Description
Reference	common:SimpleReference	No	Notification endpoint definition
SmsServiceActivation Number	xsd:anyURI [1…unbounded]	No	The destination address or addresses to the short message
Criteria	xsd:string	Yes	The text to match against to determine the application to receive the notification. This text is matched against the first word in the message, defined as the initial characters after discarding any leading whitspaceWhitespace and ending with a whitespaceWhitespace or end of message. The matching shall be case-insensitive.

8.4.1.2 Output message: StartSmsNotificationResponse

Part Name	Part Type	Optional	Description
none			

8.4.1.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value
- SVC0005 – Duplicate correlator
- SVC0008 – Overlapping Criteria

PolicyException from [6]

- POL0001 – Policy error

8.4.2 Operation: StopSmsNotification

The application may end a short message notification using this operation

8.4.2.1 Input message: StopSmsNotificationRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator of request to end

8.4.2.2 Output message: StopSmsNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.4.2.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value

PolicyException from [6]

- POL0001 – Policy error

8.4.3 Operation: StartDeliveryReceiptNotification

Start notifications to the application for delivery receipts. The reference will be where to send the delivery receipts. The **notifySmsDeliveryReceipt** method (see clause 8.2.2) must be implemented by a Web Service at the application side if it requires notification of SMSdelivery receipt. When the **startDeliveryReceiptNotification** operation is supported by the Service Provider, its use overrides the delivery receipting mechanism supported in the SendSMS API (see clause 8.1: **sendSMS** operation).

8.4.3.1 Input message: StartDeliveryReceiptNotificationRequest

Part name	Part type	Optional	Description
Reference	common:SimpleReference	No	Notification endpoint definition
FilterCriteria	xsd:string	No	The FilterCriteria will allow the service to filter flexibly. One example would be for the Service Provider to filter based on first 4 digits in MSISDN. This however is implementation specific and will be left to the Service Provider.

8.4.3.2 Output message: StartDeliveryReceiptNotificationResponse

Part Name	Part Type	Optional	Description
none			

8.4.3.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value
- SVC0005 – Duplicate correlator
- SVC0008 – Overlapping Criteria
- SVC0283 – Delivery Receipt Notification not supported

PolicyException from [6]

- POL0001 – Policy error

8.4.4 Operation: StopDeliveryReceiptNotification

The application may end delivery receipt notification using this operation

8.4.4.1 Input message: StopDeliveryReceiptNotificationRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator of request to end

8.4.4.2 Output message: StopDeliveryReceiptNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.4.4.3 Referenced Faults

ServiceException from [6]

- SVC0001 – Service error
- SVC0002 – Invalid input value

PolicyException from [6]

- POL0001 – Policy error

9 Fault definitions

9.1 ServiceException

9.1.1 SVC0280: Message too long

Name	Description
Message Id	SVC0280
Text	Message too long. Maximum length is %1 characters
Variables	%1 Number of characters allowed in a message

9.1.2 SVC0281: Unrecognized data format

Name	Description
Message Id	SVC0281
Text	Data format not recognized for message part %1
Variables	%1 Message part with the unrecognized data

9.1.3 Void

The fault code (SVC0282) is reserved and shall not be used,

9.1.4 SVC0283: Delivery Receipt Notification not supported

Name	Description
Message Id	SVC0283
Text	Delivery Receipt Notification not supported
Variables	

10 Service policies

Service policies for this service.

Name	Type	Description
GroupSupport	xsd:boolean	Groups may be included with addresses
NestedGroupSupport	xsd:boolean	Are nested groups supported in group definitions
ChargingSupported	xsd:boolean	Is charging supported for send operations
StatusRetentionTime	common:TimeMetric	A time interval that begins after the status of a short message delivery request has reached a final value. During this interval, the delivery status information remains available for retrieval by the application.
MessageRetentionTime	common:TimeMetric	A time interval that begins after the the receipt of a short message. During this interval, the short message remains available for retrieval by the application.
MaximumDescriptions	xsd:int	Maximum number of Descriptions that can be charged simultaneously
MessagingAddressesDuplicationSupport	xsd:boolean	Is duplication addresses supported for send operations

NOTE: For service policy – "MessagingAddressesDuplicationSupport", if aliases or group addresses are used:

1. Parlay X Gateway with Identity Management Framework support can verify that indeed there is a duplicate.

2. If network capability supports aliases or group addresses and the Parlay X Gateway without Identity Management Framework supporting, then the policy exception of addresses duplication may not have effect fully.

3. If network capability does not support aliases or group addresses and the Parlay X Gateway without Identity Management Framework supporting, the Parlay-X Gateway should reject the aliases and group addresses.

Annex A (normative): WSDL for Short Messaging

The document/literal WSDL representation of this interface specification is compliant to 3GPP TS 29.199-1 [6] and is contained in text files:

- parlayx_sms_notification_interface_4_0.wsdl
- parlayx_sms_notification_manager_interface_4_0.wsdl
- parlayx_sms_notification_manager_service_4_0.wsdl
- parlayx_sms_notification_service_4_0.wsdl
- parlayx_sms_receive_interface_4_0.wsdl
- parlayx_sms_receive_service_4_0.wsdl
- parlayx_sms_send_interface_4_0.wsdl
- parlayx_sms_send_service_4_0.wsdl
- parlayx_sms_types_4_0.xsd

which accompany the present document.

The WSDL files have been verified using the following files:

- 4_wsdl2Java_axis-1_4.bat
- 4_wsdl2Java_axis2-1_4_1.bat

which accompany the present document.

Annex B (informative): Description of Parlay X Web Services Part 4: Short messaging for 3GPP2 cdma2000 networks

This annex is intended to define the OSA Parlay X Web Services Stage 3 interface definitions and it provides the complete OSA specifications. It is an extension of OSA Parlay X Web Services specifications capabilities to enable operation in cdma2000 systems environment. They are in alignment with 3GPP2 Stage 1 requirements and Stage 2 architecture defined in:

[1] 3GPP2 X.S0011-D: 'cdma2000 Wireless IP Network Standard ", Version 1.1

[2] 3GPP2 S.R0037-0: "IP Network Architecture Model for cdma2000 Spread Spectrum Systems", Version 3.0

[3] 3GPP2 X.S0013-A: "All-IP Core Network Multimedia Domain"

These requirements are expressed as additions to and/or exclusions from the 3GPP specification.
The information given here is to be used by developers in 3GPP2 cdma2000 network architecture to interpret the 3GPP OSA specifications.

B.1 General Exceptions

The terms 3GPP and UMTS are not applicable for the cdma2000 family of standards. Nevertheless these terms are used (3GPP TR 21.905) mostly in the broader sense of "3G Wireless System". If not stated otherwise there are no additions or exclusions required.

CAMEL mappings are not applicable for cdma2000 systems.

B.2 Specific Exceptions

B.2.1 Clause 1: Scope

There are no additions or exclusions.

B.2.2 Clause 2: References

There are no additions or exclusions.

B.2.3 Clause 3: Definitions and abbreviations

There are no additions or exclusions.

B.2.4 Clause 4: Detailed service description

There are no additions or exclusions.

B.2.5 Clause 5: Namespaces

There are no additions or exclusions.

B.2.6 Clause 6: Sequence diagrams

There are no additions or exclusions.

B.2.7 Clause 7: XML Schema data type definition

There are no additions or exclusions.

B.2.8 Clause 8: Web Service interface definition

There are no additions or exclusions.

B.2.9 Clause 9: Fault definitions

There are no additions or exclusions.

B.2.10 Clause 10: Service policies

There are no additions or exclusions.

B.2.11 Annex A (normative): WSDL for Short Messaging

There are no additions or exclusions.

Annex B (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Dec 2006	CT_34	CP-060600	0011	--	Add support for Scheduled SMS	B	6.6.0	7.0.0
Mar 2007	CT_35	CP-070045	0013	--	Add OSA Parlay Web Services support for 3GPP2 networks	A	7.0.0	7.1.0
Mar 2007	--	--	--	--	Editorial: Aligned 5 Namespaces	--	7.1.0	7.1.1
Jun 2007	CT_36	CP-070346	0015	--	Correction to align notification mechanisms for Delivery Receipting in Short Messaging	F	7.1.1	7.2.0
Dec 2008	CT_42	CP-080892	0016		Extend information on delivery failure notifications - SMS	F	7.2.0	8.0.0
Dec 2008	CT_42	CP-080894	0017		Remove limitation of number of notification subscriptions - SMS	F	7.2.0	8.0.0
Sep 2009	CT_45	CP-090592	0018		Completion of Parlay X Part 4 – Short Messaging for Release 8	F	8.0.0	8.1.0
2009-12	-	-	-	-	Update to Rel-9 version (MCC)		8.1.0	9.0.0

History

Document history		
V9.0.0	January 2010	Publication