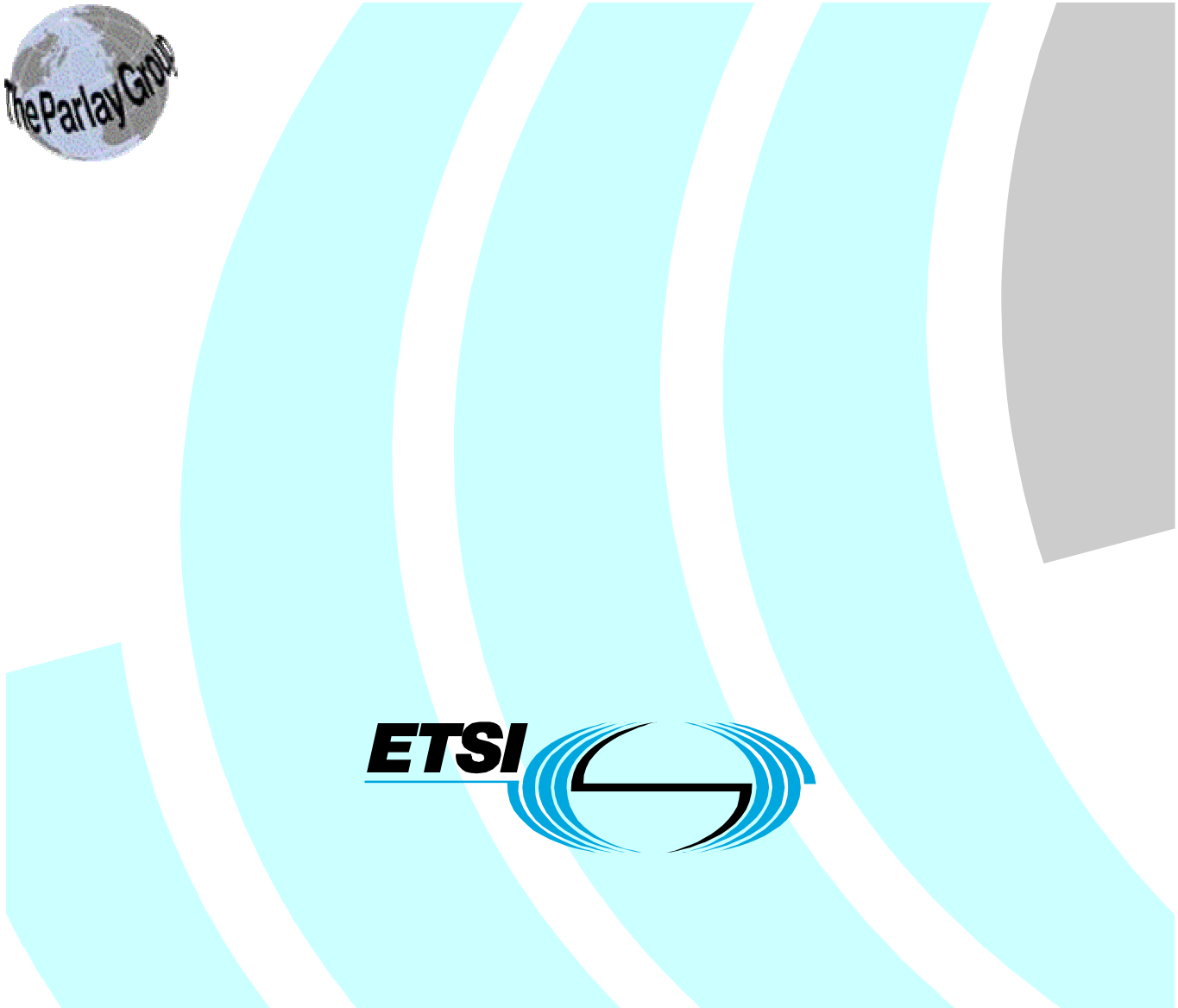
Final draft ETSI ES 202 391-4 V1.1.1 (2005-01)

ETSI Standard

Open Service Access (OSA);
Parlay X Web Services;
Part 4: Short Messaging



Reference
DES/TISPAN-01007-04-OSA

Keywords
API, OSA, service

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

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2005.
© The Parlay Group 2005.
All rights reserved.

DECT™, **PLUGTESTS**™ and **UMTS**™ are Trade Marks of ETSI registered for the benefit of its Members.
TIPHON™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights 5
Foreword 5
1 Scope 6
2 References 6
3 Definitions and abbreviations 6
3.1 Definitions 6
3.2 Abbreviations 7
4 Detailed service description 7
5 Namespaces 9
6 Sequence diagrams 9
6.1 Send SMS and report status 9
7 XML Schema data type definition 10
7.1 DeliveryStatus enumeration 10
7.2 SmsFormat enumeration 10
7.3 DeliveryInformation structure 10
7.4 SmsMessage structure 10
8 Web Service interface definition 11
8.1 Interface: SendSms 11
8.1.1 Operation: SendSms 11
8.1.1.1 Input message: SendSmsRequest 11
8.1.1.2 Output message: SendSmsResponse 11
8.1.1.3 Referenced faults 11
8.1.2 Operation: SendSmsLogo 12
8.1.2.1 Input message: SendSmsLogoRequest 12
8.1.2.2 Output message: SendSmsLogoResponse 12
8.1.2.3 Referenced faults 12
8.1.3 Operation: SendSmsRingtone 13
8.1.3.1 Input message: SendSmsRingtoneRequest 13
8.1.3.2 Output message: SendSmsRingtoneResponse 13
8.1.3.3 Referenced faults 13
8.1.4 Operation: GetSmsDeliveryStatus 14
8.1.4.1 Input message: GetSmsDeliveryStatusRequest 14
8.1.4.2 Output message: GetSmsDeliveryStatusResponse 14
8.1.4.3 Referenced faults 14
8.2 Interface: SmsNotification 14
8.2.1 Operation: NotifySmsReception 14
8.2.1.1 Input message: NotifySmsReceptionRequest 15
8.2.1.2 Output message: NotifySmsReceptionResponse 15
8.2.1.3 Referenced faults 15
8.3 Interface: ReceiveSms 15
8.3.1 Operation: GetReceivedSms 15
8.3.1.1 Input message: GetReceivedSmsRequest 15
8.3.1.2 Output message: GetReceivedSmsResponse 15
8.3.1.3 Referenced faults 15
9 Fault definitions 16
9.1 ServiceException 16
9.1.1 SVC0280: Message too long 16
9.1.2 SVC0281: Unrecognized data format 16
10 Service policies 16

Annex A (normative): WSDL for Short Messaging 17

Annex B (informative): Bibliography 18

History 19

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 ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 4 of a multi-part deliverable covering 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".

The present document has been defined jointly between ETSI, The Parlay Group (http://www.parlay.org) and the 3GPP.

The present document is equivalent to 3GPP TS 29.199-04 V6.0.0 (Release 6).

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 present document specifies the Short Messaging Web Service. The following are defined here:

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

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 and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

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

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

[2] ETSI ES 202 391-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common".

[3] ETSI TS 123 040: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Short Message Service (SMS) (3GPP TS 23.040)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 391-1 [2] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 202 391-1 [2] and the following apply:

SMS	Short Message Service
SMS-C	Short Message Service - Centre

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.

For sending a message to the network (see clause 8.1 of the present document, Send SMS API), the application invokes a message to send it and must subsequently become active again to poll for delivery status. There is an alternative to this polling mechanism, i.e. an asynchronous notification mechanism implemented with an application-side Web Service. However it was decided not to provide a notification mechanism in the first release, to make the API as simple as possible, even though the polling mechanism is not as network efficient as the notification mechanism.

For receiving a message from the network, the application may use either polling (see clause 8.3 of the present document, Receive SMS API) or notification (see clause 8.2 of the present document, SMS Notification API) mechanisms. The notification mechanism is more common: network-initiated messages are sent to autonomous application-side Web Services. Both mechanisms are supported, but the provisioning of the notification-related criteria is not specified.

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-SMS (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.

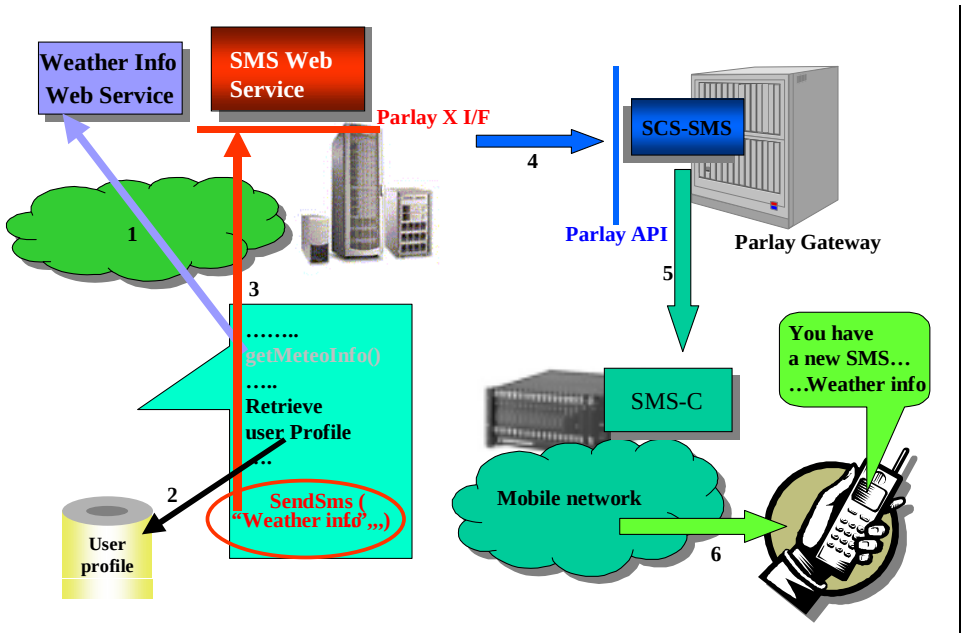


Figure 1: Send SMS Scenario

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 to retrieve 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).

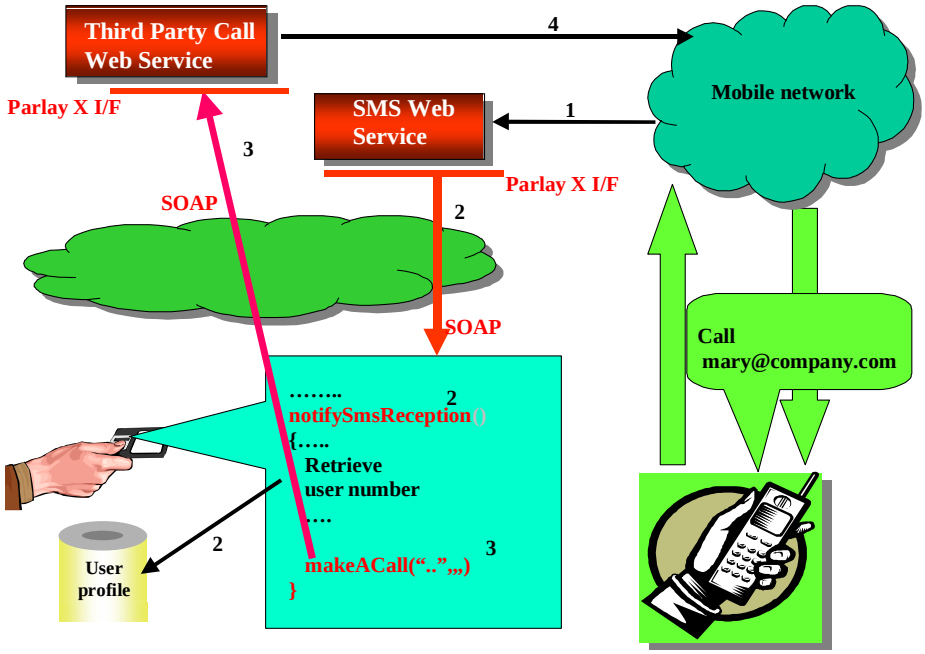


Figure 2: Receive SMS Scenario

5 Namespaces

The SendSms interface uses the namespace:

www.csapi.org/wsdl/parlayx/sms/send/v2_0

The ReceiveSms interface uses the namespace:

www.csapi.org/wsdl/parlayx/sms/receive/v2_0

The SmsNotification interface uses the namespace:

www.csapi.org/wsdl/parlayx/sms/notification/v2_0

The data types are defined in the namespace:

www.csapi.org/schema/parlayx/sms/v2_0

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [1]. 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.

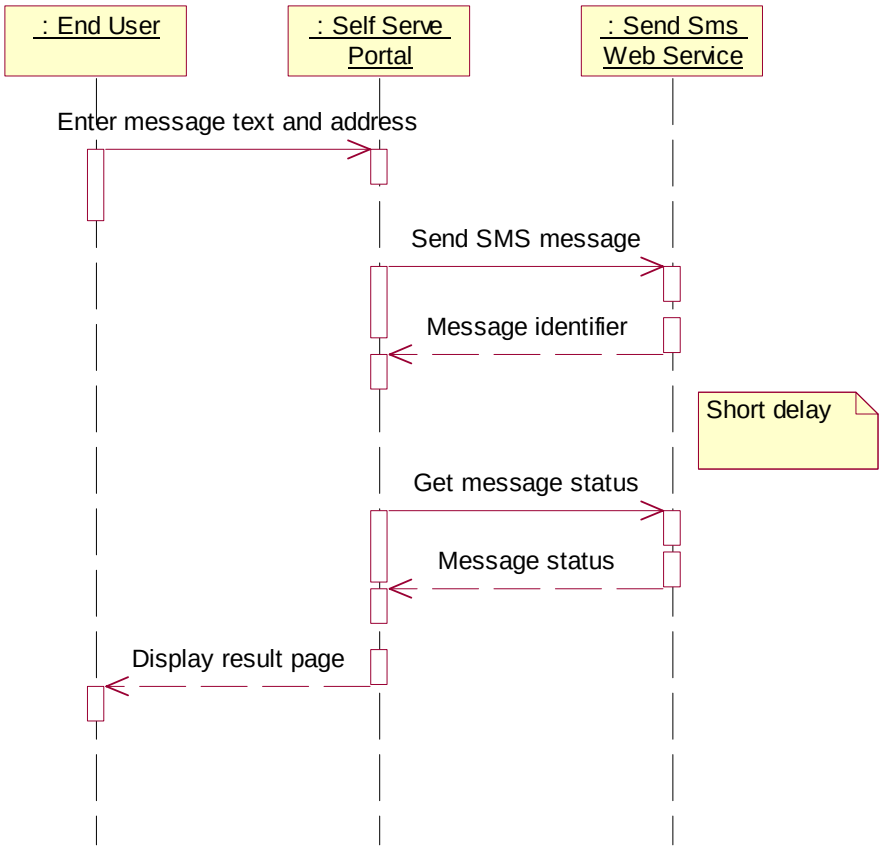


Figure 3

7 XML Schema data type definition

7.1 DeliveryStatus enumeration

List of delivery status values.

Enumeration	Description
Delivered	Successful delivery.
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.

7.2 SmsFormat enumeration

List of SMS format values.

Enumeration	Description
Ems	Enhanced Messaging Service, standardized in TS 123 040 [3], which defines a logo/ringtone format
SmartMessaging™	Defines a logo/ringtone format

7.3 DeliveryInformation structure

Delivery status information.

Element name	Element type	Description
Address	xsd:anyURI	It indicates the destination address to which the notification is related.
DeliveryStatus	DeliveryStatus	Indicates the delivery result for destinationAddress. Possible values are: • 'Delivered'; • 'DeliveryUncertain'; • 'DeliveryImpossible'.

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	Description
Message	xsd:string	Text received in SMS
SenderAddress	xsd:anyURI	It indicates address sending the SMS
SmsServiceActivation Number	xsd:anyURI	Number associated with the invoked Message service, i.e. the destination address used to send the message

8 Web Service interface definition

8.1 Interface: SendSms

8.1.1 Operation: SendSms

The invocation of **sendSms** requests to send an SMS, specified by the String **Message** to the specified address (or address set), specified by **Addresses**. Optionally the application can also indicate the sender name (**SenderName**), i.e. the string that is displayed on the user's terminal as the originator of the message, and the charging information. By invoking this operation the application requires to receive the notification of the status of the SMS delivery. In order to receive this information the application has to explicitly invoke the **getSmsDeliveryStatus**. The **RequestIdentifier**, returned by the invocation, can be used to identify the SMS delivery request.

Addresses 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.

For GSM systems, if **Message** contains characters not in the GSM 7-bit character set, the SMS is sent as a Unicode SMS.

If **Message** is longer than the maximum supported length (e.g. for GSM, 160 GSM 7-bit characters or 70 Unicode characters), the message will be sent as several concatenated short messages.

8.1.1.1 Input message: SendSmsRequest

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Addresses to which the SMS will be sent
SenderName	xsd:string	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	Charge to apply to this message (optional)
Message	xsd:string	Text to be sent in SMS

8.1.1.2 Output message: SendSmsResponse

Part name	Part type	Description
RequestIdentifier	xsd:string	It identifies a specific SMS delivery request

8.1.1.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

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

PolicyException from ES 202 391-1 [2]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.

- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.

8.1.2 Operation: SendSmsLogo

The invocation of **sendSmsLogo** requests to send an SMS logo, specified by the byte array **image** to the specified address (or address set), specified by **destinationAddressSet**. Optionally the application can also indicate the sender name (**senderName**), i.e. the string that is displayed on the user's terminal as the originator of the message, and the charging information (**charging**). By invoking this operation the application requires to receive the notification of the status of the SMS delivery. In order to receive this information the application has to explicitly invoke the **getSmsDeliveryStatus**. The **requestIdentifier**, returned by the invocation, can be used to identify the SMS delivery request.

Addresses 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.

8.1.2.1 Input message: SendSmsLogoRequest

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Addresses to which the SMS logo will be sent
SenderName	xsd:string	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message (optional)
Charging	common:ChargingInformation	Charge to apply to this message (optional)
Image	xsd:base64Binary	The image in jpeg, gif or png format. The image will be scaled to the proper format
SmsFormat	SmsFormat	Possible values are: 'Ems' or 'SmartMessaging'

8.1.2.2 Output message: SendSmsLogoResponse

Part name	Part type	Description
requestIdentifier	String	It identifies a specific SMS delivery request

8.1.2.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

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

PolicyException from ES 202 391-1 [2]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.

8.1.3 Operation: SendSmsRingtone

The invocation of **sendSmsRingtone** requests to send an SMS ringtone, specified by the String **ringtone** (in RTX format) to the specified addresses, specified by **Addresses**. Optionally the application can also indicate the sender name (**senderName**) i.e. the string that is displayed on the user's terminal as the originator of the message, and the charging information (**charging**). By invoking this operation the application requires to receive the notification of the status of the SMS delivery. In order to receive this information the application has to explicitly invoke the **getSmsDeliveryStatus**. The **requestIdentifier**, returned by the invocation, can be used to identify the SMS delivery request.

Addresses 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.

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

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

8.1.3.1 Input message: SendSmsRingtoneRequest

Part name	Part type	Description
Addresses	xsd:anyURI [0..unbounded]	Addresses to which the SMS logo will be sent
SenderName	xsd:string	SMS sender name, i.e. the string that is displayed on the user's terminal as the originator of the message (optional)
Charging	common:ChargingInformation	Charge to apply to this message (optional)
Ringtone	xsd:string	The ringtone in RTX format (see note above). (http://www.logomanager.co.uk/help/Edit/RTX.html)
SmsFormat	SmsFormat	Possible values are: 'Ems' or 'SmartMessaging'

8.1.3.2 Output message: SendSmsRingtoneResponse

Part name	Part type	Description
RequestIdentifier	xsd:string	It identifies a specific SMS delivery request

8.1.3.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

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

PolicyException from ES 202 391-1 [2]:

- POL0001 - Policy error.
- POL0006 - Groups not allowed.
- POL0007 - Nested groups not allowed.
- POL0008 - Charging not allowed.

8.1.4 Operation: GetSmsDeliveryStatus

The invocation of **getSmsDeliveryStatus** requests 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 that should be specified in an off-line configuration step. The following four different SMS delivery status have been identified:

- 'Delivered': in case of concatenated messages, only when all the SMS-parts have been successfully delivered.
- '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.

8.1.4.1 Input message: GetSmsDeliveryStatusRequest

Part name	Part type	Description
RequestIdentifier	xsd:string	It identifies a specific SMS delivery request

8.1.4.2 Output message: GetSmsDeliveryStatusResponse

Part name	Part type	Description
DeliveryStatus	DeliveryInformation [0..unbounded]	It lists the variations on the delivery status of the SMS

8.1.4.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

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

PolicyException from ES 202 391-1 [2]:

- POL0001 - Policy error.

8.2 Interface: SmsNotification

8.2.1 Operation: NotifySmsReception

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 an off-line provisioning step, identified by the **registrationIdentifier**. The criteria must at least include a **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	Description
RegistrationIdentifier	xsd:string	Identifies the off-line provisioning step that enables the application to receive notification of SMS reception according to specified criteria
Message	SmsMessage	Message received

8.2.1.2 Output message: NotifySmsReceptionResponse

Part name	Part type	Description
None		

8.2.1.3 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 specified in an off-line configuration step.

The received SMS messages are returned in **receivedSms**. 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	Description
RegistrationIdentifier	xsd:string	Identifies the off-line 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	Description
ReceivedSms	SmsMessage [0..unbounded]	It lists the received SMS since last invocation

8.3.1.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

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

PolicyException from ES 202 391-1 [2]:

- 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

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

Annex A (normative): WSDL for Short Messaging

The document/literal WSDL representation of this interface specification is compliant to ES 202 391-1 [2] and is contained in text files (contained in archive es_20239104v010101m0.zip) which accompany the present document.

Annex B (informative): Bibliography

ETSI TR 121 905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

History

Document history			
V1.1.1	January 2005	Membership Approval Procedure	MV 20050318: 2005-01-18 to 2005-03-18