

Source: CN5
Title: Report of CN5#25bis (Messaging) meeting
Agenda item: 6.5.1
Document for: INFORMATION

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-041012
Meeting #25bis, Sophia Antipolis, FRANCE, 29-30 January 2004

Source: CN5 Chair (Chelo.Abarca@alcatel.fr)
Title: Report of CN5#25bis (Messaging) meeting

Contents

1	Brief introduction of each contribution, focusing on its support of one of the five options	1
2	Discussion of options based on arguments on the contributions	4
3	Drafting session aiming at the specifications to implement the option chosen, based on the relevant contributions	5
3.1	Requirements	5
3.2	Capabilities	6
4	Participants list.....	11

1 Brief introduction of each contribution, focusing on its support of one of the five options

Only contributions directly relevant to this discussion are presented now; discussion on requirements and specifications of proposed future functionality is postponed for later in the agenda.

N5-041003 Design requirements for Messaging (Ericsson)

Presentation:

Current GMS is very complex – it has many layer of objects (mailbox -> folder -> messages), so interactions get very complicated; it also conflict with the tendency to flatness in Web Services technologies. In order to improve that we can

- improve GMS (and forget BC)
- leave GMS except error corrections and introduce a simpler, flatter new SCF.

In any case we'd be defining a new SCF.

Disadvantage of replacing GMS: applications may already be in the market that use it.

New SCF could be asynchronous, leaving the current one synchronous => a choice for developers.

Rest of the contribution is a list of requirements for the new SCF.

Discussion:

Question: GMS inconveniently complex

Comment: the problem is not the OO complexity, but it is flawed – some methods are useless. Compared to e.g. Javamail, the learning curve is steep at the beginning but after a while it is very straightforward to use. Problem with this contribution: asynchronous interfaces are complex to use, so the proposal is on one hand proposing a simpler API, on the other a more complex one. For the non-skilled developers asynchronous

Reply: developer feedback says retrieving a message is complex because of the layered structure. Agreed that it depends on the application (for example for notifications it is too complex, but many applications would anyway have to go through this structure). Agreed that asynchronous is more complex but it is necessary because of possible delays in the network.

Comment (IBM): there is complexity in GMS because for example there are lots of unnecessary data types, some objects are created in the wrong interface. Most messaging systems today are asynchronous. It would be useful to see how other systems work, instead of trying to incrementally improve GMS.

Question (Alcatel): all opinions so far indicate that there are problems

Answer (Ericsson): some BC improvements can be made but they won't meet all requirements.

Question (Marconi): if these improvements were made, and at the same time a new SCF were defined, would anybody use the fixed GMS, or would we be wasting our time?

Question (Apium): trying to improve the current GMS is not letting us include key functionality.

Answer (Aepona): there is a third way – deprecating GMS gives the wrong message to the market (that it is bad). There is a distinction between improving and rewriting.

Comment (IBM): there is a precedence in GCC, where a replacement has been done between version 2.1 and versions 3 and later, .

Comment (Alcatel): our market is more mature now, and we strive to a maturity and stability of our APIs that is usually our message..

Comment (IBM): from SP feedback it seems the current level of satisfaction with GMS is low, so there are not looking for stability.

Comment (Apium): our market message could be that GMS was developed in a time where unified messaging was not considered.

Comment (Aepona): we're spending lots of meeting time on this, and we're a technical group. The Parlay Group surely has an opinion on the market message.

Reply (Alcatel): we also need to decide whether we want GMS in 3GPP specs or not.

Suggestion (Ericsson): we need to make a whole list of requirements (like whether we want synch or asynch), and based on this decide how to go forward.

N5-041007 Generic Messaging Requirements and Capabilities (IBM)

Presentation:

Describes the requirements of a Messaging SCG (called GMS) from IBM's perception of the market. Also describes the features of this new GMS, based on these requirements and on some described design patterns – date types simpler than the existing one, but satisfying the proposed requirements. The conclusion is that in spite of the added requirements, the proposed SCF is not more complex for a developer.

Discussion:

Question (Lucent): is the management interface for the use, or the system?

Reply (IBM): two roles – the administrator can manage folders but not see them, and the user who can manipulate all in the mailbox.

Question (Lucent): is unified messaging a requirement or the interface? Believes this is the job of the application. Should SCFs decide on the fly what the transport would be?

Reply (IBM): two ways to do that, each has one deciding, but the interface should be flexible enough to allow both. The app should be able to specify a transport means, but the network should be able to overrule this decision.

Discussion postponed for later, when the SCF specification is discussed (now focus on choice of options).

Question (Eamon): are security aspects requirements of the SCF itself, or could they be supported by other means like policies?

Reply (IBM): security is something to be decided generically.

Question (Alcatel): the proposed design patterns, how can they be implemented in the current GMS, and how would this affect BC?

Reply (IBM):

- Lots of small complexities (like an union with only one choice – it's necessary to look in detail to find out there is only one choice).
- Messaging and mailbox are different, distinct operations, but they're mixed together in the current GMS. Also this requires knowing about the vendor implementation – all operations should be obvious, not implied.
- Some states are introduced where a state is not required (not the same as synchronous), for instance to work with a mailbox it is necessary to create objects, while most of the operations do not need these objects.
- Flexibility with synchronous and asynchronous models.
- Ability to isolate the message send interface, and also the admin interface – to fix a security problem by allowing use of different interfaces. Some discussion on a requirement allowing free send of messages and its implications in spam. This discussion relates to the requirements for the new functionality and therefore is postponed.

N5-041009 Discussion on Messaging SCF N5-041005 (Apium)

The second part of this contribution is not for this discussion and is postponed.

Presentation:

The following questions are raised by this contribution:

- Who will use this API, what kind of developers are the target audience?

- What is the benefit of this API compared with using IMAP together with a decent API kit, e.g. Javamail?
- What will be the cost of a gateway with this API? Even though the API offers benefits compared with an IMAP/API kit solution, is there a business case for an operator to invest in a messaging GW?
- The API design strives for OO simplicity, but the datatypes are still rather complex. Example for the new Messaging SCF: TpMsgHeaderField, TpBodyPartDescription, The Javamail API makes use of objects for folders, messages, mime parts etc, but is still relatively easy to use.

Discussion:

Comment (Ericsson): contribution 05 is based on the belief that there is a need to address the MMS requirement, but also still to support the mailbox functionality. The contribution assumes the mailbox is a sort of cache where the app can get messages at its own convenience, that's why there are no management interfaces.

Comment (IBM): today SPs have existing systems for several messaging capabilities, but not an API that allows them to present a single interface to customers, allowing them to access all these messaging systems, with folders but not messaging transparent. A gateway would present this integrating messaging storage and messaging interfaces (from the user's point of view).

Comment (Aepona): to the second question - there are other SCFs where the mapping is very lightweight, but the reasons why we use an API are others, like horizontal service abstraction, a broad-range and consistent set of APIs, the use of the Framework.

Comment (Apium): the reason to discuss the added value of this is to avoid spending time doing an API that nobody will use. Some customers want to buy for instance CC but not Mobility, and instead want to use directly IMAP – would they then use the Messaging part of the gateway?

Reply (Ericsson): proposes not to discuss business models here – our presence here shows our companies see a business case in this.

Reply (Alcatel): we have a requirement to support MMS. We can discuss

Question (Apium): for the first question, can we conclude we're addressing rather experienced developers?

Reply (Lucent): a developer that would just want to send a message would use something else. We seem to have moved the complexity discussion to developers like this.

Reply (Apium): then as a consequence we shouldn't be worried about issues like OO complexity. Agreed that rather than complexity we should be discussing that GMS is cumbersome – this is what we want to get rid of, and we shouldn't be afraid of introducing a folder object if we need one.

Question (Alcatel): does that mean we can evolve GMS in a BC way?

The meeting agrees that we cannot solve the cumbersomeness of GMS in a BC way.

2 Discussion of options based on arguments on the contributions

Agreement that we need a new SCF.

Question (Ericsson): what does deprecating the GMS mean?

Reply (IBM): preference would be to keep GMS but “mark” it as not BC, so we do not confuse people, and have a new SCF in a major release (with a new part). Parlay to decide what is the market message.

Adrian clarifies what this means for the ETSI spec: it means nothing, nobody touches them.

Agreement on this. Parlay to decide what is the market message. The major release would coincide with 3GPP Rel6.

Comment: Ericsson: one consideration is whether we want to continue improving GMS as a synchronous API, and introduce something new for those who want to use an asynch API.

IBM: sees no reason for a synch interface, it is not required. Anyway GMS cannot be improved even in a synch way and still be kept BC. As said before, patches will not fix GMS, changes we can do from the existing GMS are limited.

Agreement it is not possible to fix GMS is a BC way. Therefore the question is whether we want to have two new SCFs – and whether one of them should still be called Part 9. Agreement that the discussion of two SCFs or one should be postponed to the more detailed technical discussion.

Chair wrap-up: agreed to have a new SCF (or two). Further discussion (like one SCF or two) is technical. Any other discussion (what part it is etc) can be discussed in parallel (and with Ultan). Agreed that the meeting continues discussing technical proposals.

3 Drafting session aiming at the specifications to implement the option chosen, based on the relevant contributions

Contributions 09 and xx are withdrawn.

3.1 Requirements

Since we don't have a very detailed stage 1, and no stage 2 for MMS, we need more specific technical requirements. Discussion now focuses on the requirements, based on the contributions to this meeting.

Messaging types

Agreed to support the following types of messaging (precise functionality to be discussed):

- MMS
- SMS
- Email
- Voicemail
- Paging
- WAP Push
- IM

Other requirements (including design considerations)

- Multicasting
- Support of text messages, and text messages with attachments
- Isolation of interfaces depending on their function

- Support for asynchronous interfaces where suitable
- Support for content fidelity (the content sent is the same as the content received, alternative formatting).
- Selection for content structuring (whether the application is responsible for interpreting body content or whether interfaces are provided for this)
- Ability to notify an application of the arrival of a message
- A predictable organizational structure for message store
- Ability to store application created messages in a message store and define the destination (mailbox and folder, based on filters set by the user) for inbound messages.
- Provide end user security to isolate mailboxes from unauthorized use.
- Support for stateless approach where applicable.
- Support for a full set of operations for folders and messages.
- Support for a hierarchy of folders.
- Message identifiers will be valid for at most the duration of the application session (the duration of the application session will be dependent on the design chosen for the Messaging SCF).
- Support for correlation of message delivery notifications with the original message.
- Support for a flexible address scheme.

There was no support for a requirement for Management functionality or an administrative role. There was agreement that if provided, it should be a separate SCF, so work on the “core” Messaging functionality can continue and this discussion postponed.

This concludes the stage 1 discussions. Objective for Friday: to agree on a stage 2. Contributions for a resulting stage 3 in line with these stages 1 and 2 will be the objective for the Atlanta meeting.

3.2 Capabilities

Discussion of the capabilities proposed in the contributions to this meeting, based on the requirements agreed above.

N5-041004 Proposed structure of and capabilities in the Messaging SCF

Capabilities proposed:

- Ability to send a message
- Receive a message notification
- Set and remove a trigger for the “receive message notification” event
- Get a list of the messages in the mailbox
- Retrieve the entire message at once (i.e. both headers and entire body)
- Retrieve the headers of a message
- Retrieve the entire body of a message
- Get a list of the body parts contained in the message
- Retrieve one of the body parts contained in the message
- Delete a message from the mailbox
- Open and close the mailbox

Discussion:

Q (IBM): is “send message notification” optional?

A (Ericsson): yes.

Comment (IBM): for retrieval of parts of message, there is not always interest in retrieving the whole header, so it would be good to be able to identify which parts of the header are wanted. This is better done with an enumeration list, rather than with different methods for different retrievals.

Comment (IBM): would like to add the ability to operate on mailbox folder message, specifying the folder.

N5-041007 Generic Messaging Requirements and Capabilities

Capabilities proposed:

- Mailbox
 - createFolder – create a new folder in a mailbox
 - changeFolderName – change an existing folder name
 - deleteFolder – delete an existing folder
 - getFolderNames – get list of folder names in a mailbox
 - createMessage – create a message in a folder
 - getMessage – get a message from a folder
 - getMessageParts – get specified message parts from a folder
 - updateMessage – update an existing message
 - moveMessage – move a message to another folder in this mailbox
 - deleteMessage – delete a message
 - getMessageIds – get list of message ids within a folder
- MessageNotificationManager
 - createNotification – set up a new notification
 - start – start notifications
 - stop – stop notifications
- MessageNotification
 - messageReceived – notification that a message has been received
 - notificationError – notification that an error has occurred
- Message
 - sendSimpleMessage – send a simple text message
 - sendMessage – send a message with flexible options
 - sendAssociatedMessage – send a message and store in a mailbox folder
 - getMessageDeliveryStatus – get status of a message send action

Discussion:

Question (Lucent): the proposed “sendSimpleMessage” and “sendMessage” are two mechanisms the same thing are not desirable since we’re aiming at simplicity. Maybe the simple one could be for PX, the other for the regular APIs.

Answer (IBM): ok that the simple option is for PX.

Comment (Ericsson): proposes to write just “sendMessage” – anything else can be decided at stage 3.

Agreed.

Question (Apium): sendAssociatedMessage does not meet the requirements we have agreed. What is we want to store a draft message?

Answer (IBM): you can create a message that is a draft, and then later, when doing sendMessage, retrieve that draft and send it.

Comment (Lucent): what is the meaning of association?

Answer (IBM): since message storage and sending are split, this is the way to relate them. It would be possible to have a single sendMessage method, with a parameter for the store folder.

Comment (Apium): rename the method “sendAndStoreMessage”.

Agreed that any of the two proposed solutions are good, to be decided at stage 3.

N5-041009 Discussion on Messaging SCF N5-041005 Capabilities proposed:

- Folder support, including get the list of subfolders of a specific folder.
- Message flag support – reading and setting of the most common message flags: Answered, deleted, seen. Also, the draft, flagged, recent and user flags could be supported. Flags to be supported to be discussed later.
- Unread message count support.
- Header field handling - no data types for all the headers available, just a method asking to get a specific header (if we’re dealing with skilled developers, they can specify the header they want).

Discussion

Question (Lucent): the Flag support is very email inspired, how would it work for other messaging types?

Answer (Apium): for SMS there is no such notion. If a particular message store does not support flags there could be an exception.

Answer (IBM): some flags are relevant for SMS, MMS, voicemail, like “unread message count” (you have messages but haven’t read them). “Delete” is something we’ll have to find a semantic for, so the flag may or may not be relevant.

Comment (Lucent): we’re building a Messaging SCF by looking for common functionality from all messaging types, and also some specific. In the latter case we need to be very specific on how the behaviour would be in other message types.

Agreed.

Question (IBM): could be that a mailbox session ID is not necessary, though this is probably a stage 3 issue. In a stateless model it is not necessary to open mailbox before interacting with it (there is nothing to open).

Agreed that this should be discussed in stage 3.

Question (Lucent): how is authentication handled in a stateless model?

Answer (IBM): it is associated to each operation.

Question (Lucent): how does it work for a non-email type of operation?

Answer (IBM): today we don’t have a way to identify a sub-entity in the application – only the app authenticates with the gateway. We need to define how to establish this identify – in a session, or in a message when performing an operation.

Agreed that this should be discussed in stage 3.

N5-0410010 Direct Messaging API Discussion Fodder Proposes having different APIs for “push” (where the message is sent directly to a terminal where the end user can examine the message more or less directly) and “pull” (where the recipient of a message explicitly retrieves the message from a server) Messaging. Afraid that otherwise we wouldn’t have enough functionality supported for the “push” kind of message. An UML model is proposed for the “push” case. Focus if in the wish to have a standard API for specifying SMS service properties.

Discussion:

Question (IBM): DirectMessage information will be common to pull and push. It should be harmonizable no matter the message type.

Answer (Apium): agreed.

Question (BTEExact): could the pull message be the old part 9?

Answer (Apium): no.

Question (Lucent): is it the idea to have different APIs for different message types?

Answer (Apium): party; the idea is that there are at least some SMS specific methods, maybe WAP. The reasons is that these are things visible to the user, and otherwise we’d be limiting the application functionality.

Comment (Aepona): the API must enable maximum app flexibility.

Comment (IBM): this would be good to have in a 29.998 document, because they’re mappings to the individual message systems. An application developer could be interested in sending a message, but not necessarily in how it is done.

Answer (Apium): in order to have the mapping, we need to have the functionality in the API. True that all application developers may not be interested in these details, but then we just need to have good default options for the parameters. Having the functionality available does not mean everybody has to use it.

Question (Alcatel): but if we have the interfaces specific to the message type then this is visible to all developers.

Apium: this is just a proposal – the implementation details can be decided later.

Alcatel: but this is a stage 2 decision, we cannot postpone it.

Eriscsson: sees the benefits in giving the app the flexibility, but the proposal also creates lots of new complexity – new interfaces, and also for a remote API there will be all these objects, resulting in lots of interactions; also need to monitor the lifecycle of all these objects.

Apium: that could be solved having Direct Message as a data type.

Aepona: prefers that on the API we’re not over prescriptive at parameter level; true that we need to address this, but there should be other solutions.

Conclusion: the proposal to split in two APIs - one for push one for pull – is dropped. For the second proposal – the split in interfaces that are specific to the messaging type – it is left for further discussion.

Conclusion

The following has been agreed as functionality for the Messaging SCF(s):

- Mailbox
 - Mailbox security
 - Create Folder – create a new folder in a mailbox
 - Change Folder Name – change an existing folder name
 - Delete Folder – delete an existing folder

- Get Folder Names – get list of folder names in a mailbox
- Get Folder Info – folder info would be unread messages and total number of messages
- Create Message – create a message in a folder
- Get Message – get a message from a folder
- Get Message Parts – get specified message parts from a folder (parts include header parts, body parts and flags)
- Update Message – update an existing message
- Set Message Flags (may be part of update message)
- Move Message – move a message to another folder in this mailbox
- Delete Message – delete a message
- Get Message Information – get list of message Ids, or message header parameters within a folder
- Get Subfolders
- Support of sub-folders (may be an error code or a query)
- Open and close the mailbox (depending on the stage 3 approach)
- MessageNotificationManager
 - Create Notification – set up a new notification
 - Start – start notifications
 - Stop – stop notifications
- MessageNotification
 - Message Received – notification that a message has been received
 - Notification Error – notification that an error has occurred
- Notification for message delivered on a send message
 - Message Delivered – notification that a message has been delivered
 - Notification Error – notification that an error has occurred
- Message
 - Send Message – send a message with flexible options (including support for multicasting, attachments and delivery confirmation)
 - Send and Store Message – send a message and store in a mailbox folder
 - Get Message Delivery Status – get status of a message send action
 - Flexible address scheme

This is the functionality agreed, that together with the requirements above will be used as input to stage 3 contributions in Atlanta.

4 Participants list

Chairman

ABARCA Chelo	ALCATEL S.A.	FR
STRETCH Richard	BT Group Plc	GB

Chair's Guests (Parlay Members)

DYST Joergen	Appium Technologies	SE
SOBIRK Daniel	Appium Technologies	SE

Members

BROOK Richard	SAMSUNG Electronics	GB
HOOPER Mark	Lucent Technologies N. S. UK	GB
HU Yun-Chao	ERICSSON LM	SE
HUMPHREY Jane D	MARCONI COMMUNICATIONS	GB
MCINTYRE Joe	IBM EUROPE	DE
MURRAY Eamonn	AePONA LTD	GB
UNMEHOPA Musa	Lucent Technologies N. S. UK	GB
VAN RIJSSEN Erwin	ERICSSON LM	SE

Number of Attendees: 12

Name	Organization represented	Status, partner	Ctry	Phone	Email
Ms. Chelo Abarca	ALCATEL S.A.	3GPPMEMBER (ETSI)	FR	+33 1307 70469	chelo.abarca@alcatel.fr
Mr. Richard Brook	SAMSUNG Electronics	3GPPMEMBER (ETSI)	GB	+44 7776 181555	richardbrook39@aol.com
Mr. Joergen Dyst	Appium Technologies	Parlay	SE	+46 40 664 29 73	jorgen.dyst@appium.com
Mr. Mark Hooper	Lucent Technologies N. S. UK	3GPPMEMBER (ETSI)	GB	+1 919-844-4667	hooper@lucent.com
Mr. Yun-Chao Hu	ERICSSON LM	3GPPMEMBER (ETSI)	NL	+31 161 24 2148	yunchao.hu@ericsson.com
Ms. Jane D Humphrey	MARCONI COMMUNICATIONS	3GPPMEMBER (ETSI)	GB	+44 24 76564232	jane.humphrey@marconi.com
Mr. Joe McIntyre	IBM EUROPE	3GPPMEMBER (ETSI)	US	+1 512 823 1675	joe@us.ibm.com
Mr. Eamonn Murray	AePONA LTD	3GPPMEMBER (ETSI)	GB	+44 28 90269188	eamonn.murray@aepona.com
Mr. Daniel Sobirk	Appium Technologies	Parlay	SE	+46 46 286 26 80	daniel@hds.ideon.se
Mr. Richard Stretch	BT Group Plc	3GPPMEMBER (ETSI)	GB	+44 1473 607487	richard.stretch@bt.com
Mr. Musa Unmehopa	Lucent Technologies N. S. UK	3GPPMEMBER (ETSI)	NL	+31 35 687 1684	unmehopa@lucent.com
Mr. Erwin van Rijssen	ERICSSON LM	3GPPMEMBER (ETSI)	NL	+31161242320	Erwin.van.Rijssen@ericsson.com