	Commercial in Confidence GP2GP Supplementary Specification: Handling Large Messages			
	Programme	**Programmes & Service Delivery**	**DOCUMENT RECORD ID KEY**	
	Sub-Prog / Project	**GP2GP**	***NPFIT-PC-BLD-0170.03***	
	Prog. Director	**Kemi Adenubi**		
	Owner	**Jill Hepworth**	Version	**1.1**
	Author	**Jim Poole** **Will Nossiter** **Tim Tett** **Aled Greenhalgh**		
	Version Date	**13-Feb-2014**	Status	**Final**

GP2GP Supplementary Specification

Subject	GP2GP: Handling Large Messages
Reference	NPFIT-PC-BLOD-0170.03 v1.1
Supplier(s)	All
Summary	This supplementary specification provides the detailed requirements for the sending of GP2GP messages that are outside Spine size limitations.
Justification	Without the implementation of this specification, an increasing number of GP2GP transfers will fail because they exceed the size and attachment constraints imposed by the TMS. Implementation of this specification removes the remaining technical barrier to successful EHR transfer.
Rollout Dependencies	Conformance with GP2GP Compliance Specification V1.1a

Amendment History:

Issue	Version	Date	Amendment History
01	01	5-Aug-2010	This requirement was originally documented in the GP2GP Compliance Specification V2.2 specification published in October 2009 Release 6.1. Further analysis has changed the original specification and this document was created as part of the 2010 specification review and rewrite.
02	02	21-Sep-2010	Peer reviewed
03	02	28-Jan-2011	Draft for approval
1.0	02	28-Jan-2011	Approved
1.1	03	13-Feb-2014	Updated for review

Forecast Changes:

Anticipated Change	When
No material change anticipated.	

Reviewers:

The document must be reviewed by CfH / DoH parties listed below before sign off. If named individuals are not qualified to review the document, they should act as representatives of the interested party and delegate the review to others within the organisation.

The document is also circulated to GPSoC system suppliers and is an opportunity for them to comment before the specification is approved.

Name	Title / Responsibility	Date	Version
Will Nossiter	GP2GP Technical Architect	28-Jan-20111	1.1
Jill Hepworth	GP2GP Program Manager	21-Sep-2010	1.1
Pete Turnbull	GP2GP Integration and Clinical Validation Manager	21-Sep-2010	1.1
Dave McAvenue	GP2GP Integration and Clinical Validation Lead	21-Sep-2010	0.2
Dave Bagnall	GP2GP Compliance Test Manager	21-Sep-2010	1.1
Ramsay Baker	GP2GP Deployment Manager	21-Sep-2010	1.1
Damian Murphy	Senior Assurance Manager	27-Jan-2011	0.3
Kevin Sprague	Interoperability Team	21-Sep-2010	0.2
Stuart Davies David Stables	EMIS	21-Sep-2010	0.2
Ewan Jones	iSoft	21-Sep-2010	0.2
Steve Lewis	InPS	21-Sep-2010	0.2
Chris Bates (TPP) Richard Sutcliffe (CSC)	TPP / CSC	21-Sep-2010	0.2
Vince Gregori	Microtest	21-Sep-2010	0.2

Name	Title / Responsibility	Date	Version
Aled Greenhalgh	GP2GP Solutions Architect		1.1

Approvals:

This document requires the following approvals:

Name	Signature	Title / Responsibility	Date	Version
Kemi Adenubi		CfH GPIT Programme Director		0.3
Jill Hepworth		GP2GP Development Programme Manager		1.1
Mike Curtis		Tech Office		1.1

Distribution:

Reviewers and approvers plus:

Name	Title / Responsibility	Date	Version
Alasdair Thompson	GPSoC		0.3
Alan Hassey	GP2GP Project Board Member		1.1
Paul Cundy	GP2GP Project Board Member		1.1
	GPSoC Release Managers		0.3

Document Status:

This is a controlled document. This document version is only valid at the time it is retrieved from controlled file-store, after which a new approved version will replace it.

On receipt of a new issue, please destroy all previous issues (unless a specified earlier issue is base-lined for use throughout the programme).

Related Documents:

These documents will provide additional information.

Ref no	Doc Reference Number	Title
1	HSCIC-PC-BLD-0068	GP2GP R2.2 Requirements Specification
2	NPFIT-PC-BLD-0172.01	Use Case 1: Transfer electronic healthcare record
3	NPFIT-PC-BLD-0173.01	Use Case 2: Transfer and analyse management information
15	NPFIT-PC-BLD-0171.01	Supp Spec: Harvesting management information
18	NPFIT-PC-BLD-0083.08	GP2GP Response Codes
19	NPFIT-PC-BLD-0069.24	GP2GP Spine Technical Design
20	- See note below	TMS External Interface Specification
21	http://www.ietf.org/rfc/rfc2392.txt	NWG Memo: Content-ID and Message-ID Uniform Resource Locators
30		ebXML Message Service Specification v2.0

Important Note – EIS Versions

Each version of the External Interface Specification relates to a particular release of the 'Spine', e.g. EIS 11.6 relates to the 2008B release and EIS 12.x relates to 2009.

Later version of the EIS will be published from time to time. Suppliers are required to ensure that their systems support the latest applicable version of the EIS for the spine release that their systems will be operating under.

All EIS versions can be found in the EXT Infrastructure and/or EXT Common (earlier versions) folder within FileCM

Important Note – Structure of specification documents

Figure 1 of document reference #2 shows the documents that comprise the GP2GP: Handling Large Messages

Glossary of Terms:

List any new terms created in this document. Mail the NPO Quality Manager to have these included in the master glossary above [1].

Term	Acronym	Definition
Common Point to Point messaging	P2P	Point to point messaging service across TMS designed to forward unspecified messages utilising interaction COPC_IN000001UK01 from MIM 7
Common Content Large Messaging	CCLM	Generic approach to the deconstruction and sending of a Large Message (greater than 5MB or more than 100 ebXML attachments) in parts using common point to point messages to carry large payload and the existing message to carry core HL7 content and associated message IDs.
	ebXML	An XML based data interchange standard for business to business data. Successor of EDIfact and similar EDI standards.
External Interface Specification	EIS	Documents specifying GP2GP and other external interfaces to the Spine
Health Level 7	HL7	An XML based data interchange standard for heathcare information. Usually embedded in ebXML messages.
Large Message		A message is classified as large if it exceeds the number of attachments and / or total message size constraints of the Spine GP2GP domain.
Transaction Messaging Service	TMS	A subsystem of the Spine that provides the interfaces between Spine data, end-systems and services external to the Spine.
Electronic Healthcare Record	EHR	A record of a patient's primary care transferred between primary care organisations using the GP2GP solution.
EHR Extract	-	The extracted information from a patient's old GP practice electronic patient record that is to be sent to the patient's new GP practice.
EHR Request	-	The message sent by the Requesting system to the Sending system requesting the EHR Extract
EHR Response		Used synonymously with 'EHR Extract'
Electronic Patient Record	EPR	A patient's primary care record held electronically within a primary care system.

Message Implementation Manual	MIM	The reference that defines the message patterns, schemas and content of the GP2GP messages used in GP2GP.
	MIM 3	Specifically version 3.1.10 of the MIM that defines the messages used in GP2GP baseline 1.1a and 2.2a.
	MIM 7	Specifically version 7.2.02 of the MIM that defines the messages used in GP2GP baseline 2.2b and 2.2c. This also covers the Common Point 2 Point messages used in 2.2a for the Large Messaging requirements bundle.
Organisation Data Service	ODS	ODS codes (formerly NACS codes) provide a unique identifier for any organisational entity providing NHS services, whether a trust, PCT, a hospital, a ward within a hospital, a treatment centre or mobile unit.
Requesting System		The system that requests an EHR Extract, i.e. the system of the patient's new practice.
Safe Exchange Framework	SEF	Message filtering service that can inhibit messages between suppliers / software / versions. Allows central shut down of specific GP2GP interactions in the event of (clinical safety) problems.
Sending System		The system that sends an EHR Extract, i.e. the system of the patient's old practice.
Large Messaging		A specialisation of the Common Content Large Message (CCLM) solution to overcome the Spine TMS limitations on an EHR Extract message in the GP2GP domain.

The keywords MUST, SHOULD and MAY are to be interpreted as described in RFC2119:

- MUST: This word, or the terms "REQUIRED" or "***SHALL***", means that the definition is an absolute requirement of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications MUST be understood and carefully weighed before choosing a different course.
- MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One implementer may choose to include the item because a particular implementation requires it or because the implementer feels that it enhances the implementation while another implementer may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

Contents

1. Background

1.1. The need for a Large Message protocol

The NHS National Spine ("Spine") provides a proven set of transports for shipping messages between identified and accredited systems. Spine hosts various services itself, but also supports messaging between systems external to it – for example between two GP practice systems, or between a GP system and an Acute Trust system. These "point to point" delivery services are implemented using ebXML "Reliable Messaging", so offer de-duplication, retry and so on. The vast majority of messages handled by Spine are relatively small. Most messages consist of some header information and an HL7v3 XML payload, typically totalling less than 50KB and very often less than 10KB.

So it is reasonable that Spine be optimised for messages in this size range and, as a result, Spine enforces a maximum message size of 5MB. However in some cases messages have to be transmitted that are larger than this:

- A relatively small message payload, with a large number of attachments
- A very large message
- A message with large attachments

Business requirements dictate that these must be handled. Simply to increase Spine's maximum allowed message size is not an attractive option – if for no other reason than "larger than 5MB" has no upper bound.

1.2. Potential problems with EIS Large Messaging

The Spine External Interface Specification (EIS) [Ref: 20] presents a large messaging protocol that requires message handling system suppliers to implement code to compress and split a large message, and to orchestrate transmission and reassembly of the fragments. Each fragment is controlled separately, with individual timing and retry capabilities. It essentially replicates TCP at a level a little below the application.

Writing a handler for this is complex and consequently expensive. As individual suppliers (at least at the level of message handler vendors) are required to construct the large message adaptor, this expense is replicated, and scope is introduced for complex interoperability failures. Where message handler vendors have made similar size optimisations as Spine, how the large message adaptor is integrated into the existing MHS is unclear – especially in the case where the large message requires routing inside the end system rather than being consumed at the receiving MHS.

Whilst the EIS system provides for automated retry of message fragments, the loss or abandonment of a fragment will destroy the entire transmission due to the need to have the fragments intact in order to reconstitute the message on receipt.

The EIS system provides some element of service discovery with its "Transmission Parameters" request, it is unclear how it would adapt to improvements in message addressing.

The EIS large message protocol transmits as a series of "large message fragments" – it does not transmit as the actual message type in the way that smaller messages do. As such integration with Spine and end-system security and orchestration is again unclear and potentially a fruitful source of obscure interoperability and maintenance trouble.

None of these are necessarily "show-stoppers". With a need for rapid, trouble free implementation NHS CFH have identified the following solution.

2. Large Messaging Definition

2.1. Features of "Common Content Large Messaging" in GP2GP

Any alternative large messaging protocol for GP2GP must have the following features, to address the issues raised against the EIS mechanism:

- Use conventional Spine messaging techniques. This eases implementation by separating any specific large-message processing from transmission. It also means that existing security and message handling processes and procedures are re-used, and improvements in addressing and general discovery services are immediately usable with no change to the LM handler.
- Use a mechanism to make the sender aware that the recipient is Large Message aware. This must be backward compatible with non-Large Message aware systems still operating.
- Ensure that the business meaning of the final acknowledgement of the EHR Request Completed message is maintained. This may be a trigger to subsequent actions.
- Send the primary payload under the same identifier as it would for "non-LM" messaging. This is important for integration with existing message handling and access control.
- Support "survivability" in the case where part of the transmission is lost. Losing some parts of the transmission will always be fatal. But not all parts need be equally important – in some cases parts may fail but processing can still continue. This makes physical loss of a fragment similar to "logical loss" where it is delivered either corrupt, or in a format unreadable by the receiver.
- Support "incremental" receipt (generic functionality). Not all applications will require this or find it suitable. However in the case where it is, receivers should be able to start processing as data is received, without necessarily waiting on complete delivery. The decision on whether to do so should be made based on application guidance, and contextual information available to the receiving system.
- Following from the last two features, a protocol should support "incremental failure", and to be able to signal individual failures but for the receiver to carry on processing if it decides it is appropriate to do so.
- Support the case where message receipt is actually performed by some internal system, which receives content after routing from the Spine-connected MHS.

In addition to this, "basic" capabilities are:

- Handle "large" messages made up of many attachments
- Handle large single attachments beyond the size of Spine limitations
- Handle large HL7 payloads.
- Handle transmission of Spine unsupported MIME file types
- Handle any combination of the above.
- Support use of ConversationID
- Have configurable settings to cope with changes in current Spine limitations on attachment size, message size and number of attachments.
- Be independent of the MIM version of the GP2GP EHR Request, Extract and Application Acknowledgment messages.

2.2. Large Messaging Approach

The CCLM approach is based on splitting a Large Message up and transferring it as standalone chunks. The "primary" message (the HL7v3 XML payload – EHR Extract) is separated and sent first with as many attachments as will fit within Spine limitations (currently up to 99 files attached and a maximum size of 5MB). Additional files are sent as attachments to discrete Common Point to Point messages: *urn:nhs:names:services:gp2gp:COPC_IN000001UK01*. The message flows depicted in Figure 2 show the sequence and types of messages used in the non-Large Message process. Figure 2 illustrates the additions for the Large Message process. In a successful scenario the following steps occur:

1. The Requesting system sends the EHR Request to the Sending system. The EHR Request message will use the highest common MIM version. This can be determined by querying SDS – consult the SDS section of the Spine Technical Design [Ref: 19].
2. The Sending system performs the required checks on the patient and their record. If the EPR is available, the EHR Extract is constructed and further checks on the size, number of attachments etc are made to determine if sending the EHR Extract requires the use of Large Messaging.
    a. If it does not require Large Messaging, the EHR Extract is sent and, if MIM 3 messaging is being used, a positive Application Acknowledgement to the request is also sent (See Note below).
    b. If Large Messaging is required, the Sending system checks whether the patient's new practice (the Requesting system) supports the Large Messaging Protocol by querying the Spine Directory Service – consult the SDS section of the Spine Technical Design [Ref: 19].
        i. If the Requesting system does not support the Large Message interactions, a negative Application Acknowledgement with Response code 14 is sent and the GP2GP transfer process ends.
        ii. If the Requesting system does support the Large Message interactions, the EHR Extract is broken up into additional Common Point to Point messages to ensure the size and attachment limits are conformed with. The EHR Extract message is sent containing references to external Message IDs of the Common Point to Point messages used to send the remainder of the Extract.

    (Note: the positive Application Acknowledgement has been removed for DMS 1/MIM7 EHR Requests in Baseline 2.2 of the GP2GP Compliance requirements but must continue to be sent in response to MIM3 EHR Requests to support backward compatibility).

3. The Requesting system receives the EHR Extract with or without references to external Message IDs of the Common Point to Point messages.
    a. If there are no external references, the process continues as normal to alert the General Practice to integrate or reject the patient's EHR Extract into their local record.
    b. If there are external references, the Requesting system instructs the Sending system to send the additional Extract data in the other messages by sending a "CONTINUE" message in the form of a Common Point to Point message. This message, as with all the messages, contains the Conversation ID used in the original EHR Request.
4. The Sending system receives the "CONTINUE" message and knows this is an instruction to continue with the transmission of the remainder of the Extract.
5. The Sending system splits the remainder of the Extract up into one or more Common Point to Point messages, each conforming to the TMS limitations above on size and number of attachments, and sends them. It is not necessary to wait for the previously sent Common Point to Point message to be acknowledged before sending the next. Resend must be supported as necessary in line with the message's contract properties. The Sending system will log any message not sent within 24 hours.
6. The Requesting system receives the Common Point to Point messages and rebuilds them from constituent parts where they have been split up. Positive or Negative Application Acknowledgements are sent in response to each of the individual messages. Consult the GP2GP Response Codes document for the list of Large Message related negative Application Acknowledgement codes.
7. When all the expected messages (listed in the EHR Extract headers) have been received or the overall timeout of the transfer has been reached (see later), the system prompts the General Practice to integrate or reject the re-constituted EHR Extract. It is not clinically safe to integrate a partially received EHR Extract and therefore it is prevented by the Requesting system. If the overall timeout of the transfer has been reached the Common Point to Point message will be rejected with Response code 25.

8. When a user integrates the EHR Extract or rejects it, the Requesting system sends a final Application Acknowledgement to the Sending system. This acknowledgement has the same business meaning as for a non-large message transfer. The only option for integration is full integration of all elements that were sent after successful receipt and re-constitution. The system must only return a positive Application Acknowledgement if all sent messages have been successfully received and all of the reconstituted EHR Extract has been integrated. If some Large Messages were NOT successfully received or re-constituted or a time-out occurred the system must return a negative Application Acknowledgement with Response Code 31.

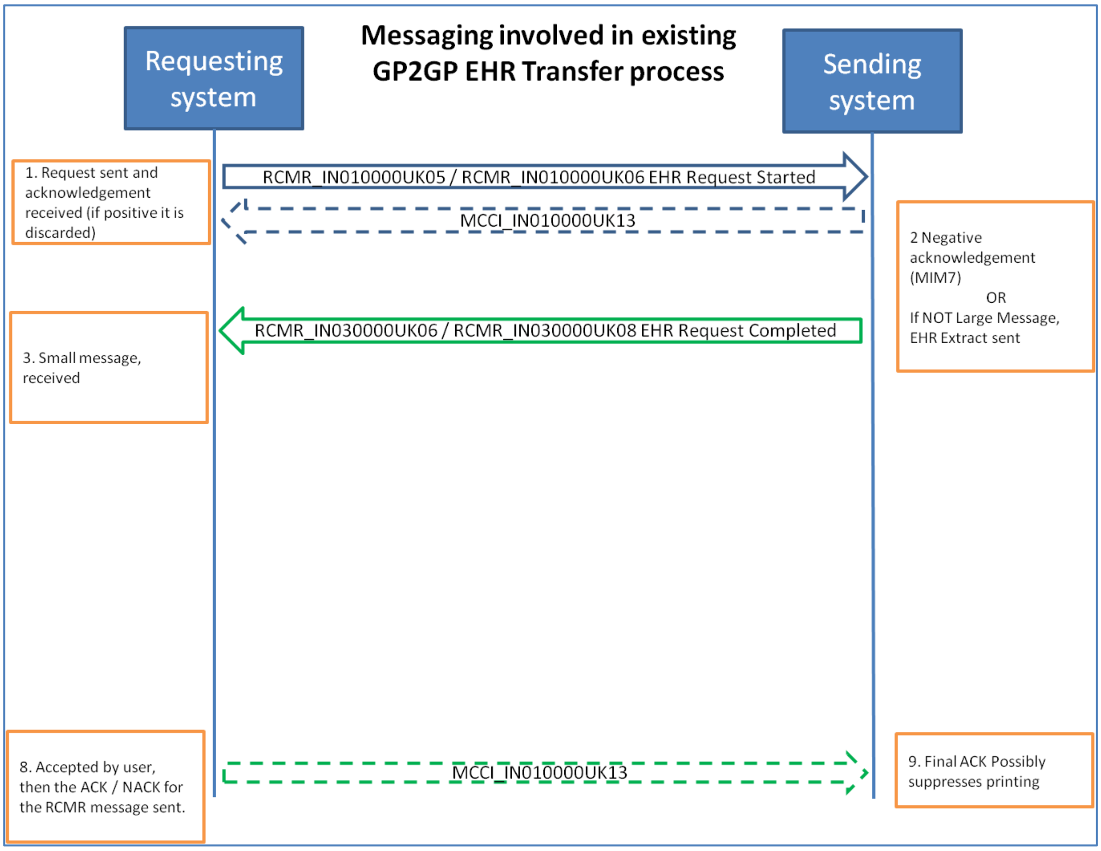
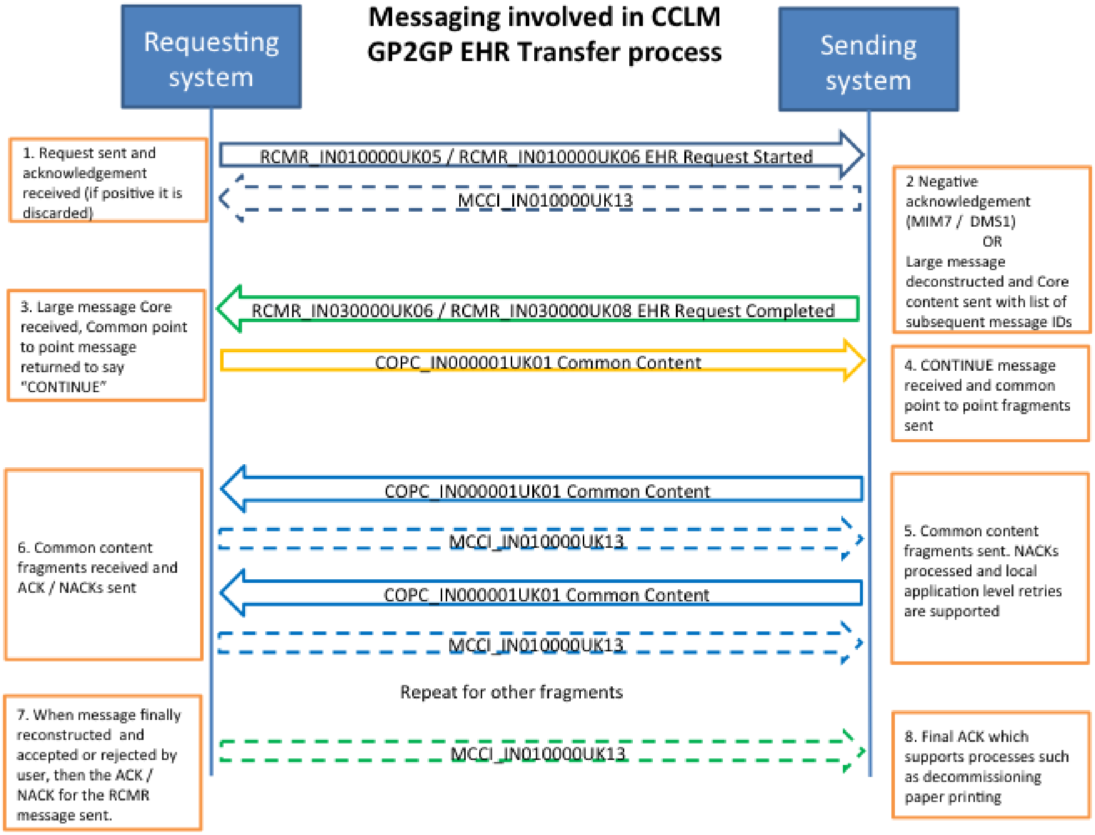

Requesting system
Messaging involved in existing GP2GP EHR Transfer process
Sending system
1. Request sent and acknowledgement received (if positive it is discarded)
RCMR_IN010000UK05 / RCMR_IN010000UK06 EHR Request Started
MCCI_IN010000UK13
2 Negative acknowledgement (MIM7) OR If NOT Large Message, EHR Extract sent
RCMR_IN030000UK06 / RCMR_IN030000UK08 EHR Request Completed
3. Small message, received
8. Accepted by user, then the ACK / NACK for the RCMR message sent.
MCCI_IN010000UK13
9. Final ACK Possibly suppresses printing

Figure 1 - Schematic view of GP2GP Non-Large Message transactions


Requesting system
Messaging involved in CCLM GP2GP EHR Transfer process
Sending system
RCMR_IN010000UK05 / RCMR_IN010000UK06 EHR Request Started
MCCI_IN010000UK13
RCMR_IN030000UK06 / RCMR_IN030000UK08 EHR Request Completed
COPC_IN000001UK01 Common Content
COPC_IN000001UK01 Common Content
MCCI_IN010000UK13
COPC_IN000001UK01 Common Content
MCCI_IN010000UK13
Repeat for other fragments
MCCI_IN010000UK13

1. Request sent and acknowledgement received (if positive it is discarded)

2 Negative acknowledgement (MIM7 / DMS1) OR Large message deconstructed and Core content sent with list of subsequent message IDs

3. Large message Core received, Common point to point message returned to say "CONTINUE"

4. CONTINUE message received and common point to point fragments sent

5. Common content fragments sent. NACKs processed and local application level retries are supported

6. Common content fragments received and ACK / NACKs sent

7. When message finally reconstructed and accepted or rejected by user, then the ACK / NACK for the RCMR message sent.

8. Final ACK which supports processes such as decommissioning paper printing

Figure 2 - Schematic view of GP2GP Large Message transactions

2.3. Linking up the Messages

The Large Message approach requires that a mechanism exists to correlate all the discrete Common Point to Point messages, and the "primary" HL7 message so that the receiver knows what to expect, and can reassemble what it receives.

Spine asynchronous messages are shipped using ebXML, where a message consists of a SOAP header with ebXML transmission management information, and a SOAP body with a manifest that carries information on the content. The manifest references content via a URI. This URI may refer to an attachment in the message (referring to the "content-id" of the attachment), and this is how HL7 content is referenced. It is also how attachments sent "in-line" within the EHR Response message are referenced. The ebXML[1] specification (section 3.2, p22 et seq) explicitly allows for the URI to point to a resource outside the transmission (message) that carries the manifest. An error occurs only where the receiver is unable to resolve the resource referenced by the URI in the manifest.

When Spine's messaging transport encounters manifest items other than the main HL7 message, it disregards them. So a Spine message sent with a manifest that in addition to the HL7 part also references "external" data passes successfully through TMS. Figure 3 to Figure 5 illustrate this.

```
<eb:Manifest SOAP:mustUnderstand="1" eb:version="2.0">

<eb:Referencexlink:href="cid:847eb954-5749-11de-9ce0-
a54f114d2be6@spine.nhs.uk">

      <eb:Schema eb:location="http://www.nhsia.nhs.uk/schemas/HL7-
      Message.xsd" eb:version="1.0"/>
      <eb:Description xml:lang="en">HL7 payload</eb:Description>
      <hl7ebxml:Payload style="HL7" encoding="XML" version="3.0"/>
```

[1] See *ebXML Message Service Specification* v2.0, p22 section 3.2 et seq [Ref: 30].

```
    </eb:Reference>
    <eb:Reference eb:id="_0F6A1B08-9CE7-455D-B2FD-4EF8DE4D9C49"
    xlink:href="cid:84812a55-5749-11de-9ce0-a54f114d2be6">
      <eb:Description xml:lang="en">Attachment</eb:Description>
    </eb:Reference>
</eb:Manifest>
```

Figure 3 - Normal HL7 plus “in-line” attachment

Inspection of the contents of Figure 3 to Figure 5, show a manifest “as sent” and “as received” and shows that Spine TMS (in the NIS1 test environment in this case), forwards the message intact.

Note that for the HL7 and the “in-line” attachments, the manifest reference uses a “cid:” or content id URI scheme. The attachments are sent as MIME parts, and the content id scheme says that the following data is the content id of a MIME part. The external examples show a similar structure, but use a “mid:” URI scheme[2]. This is a message id – and on receipt the large message is reassembled by following these message ids. In CCLM the message id used, is the HL7 message id of the Common Point to Point COPC_IN000001UK01 message.

```
<eb:Manifest SOAP:mustUnderstand="1" eb:version="2.0">
    <eb:Reference xlink:href="cid:40fc95c5-5757-11de-9a31-
    eb35c6cf0e0a@spine.nhs.uk">
      <eb:Schema eb:location="http://www.nhsia.nhs.uk/schemas/HL7-
      Message.xsd" eb:version="1.0"/>
      <eb:Description xml:lang="en">HL7 payload</eb:Description>
      <hl7ebxml:Payload style="HL7" encoding="XML" version="3.0"/>
    </eb:Reference>

    <eb:Reference eb:id="_3C64633D-291E-4077-AA47-886FD7FC64E3"
    xlink:href="mid:1A60E99A-C40E-44E6-9CDA-296D36423267">
      <eb:Description xml:lang="en">Filename="EEB93039-4285-4937-AEDB-
      18844C14DC9inps.co.ukVision3.gzip" ContentType=text/xml Compressed=No
      LargeAttachment=No OriginalBase64=No Length=302448 DomainData="X-
      GP2GP-Skeleton: Yes"</eb:Description>
    </eb:Reference>

    <eb:Reference eb:id="_0F6A1B08-9CE7-455D-B2FD-4EF8DE4D9C49"
    xlink:href="mid:A64A1C94-F913-455C-897A-740060CEE67A">
      <eb:Description xml:lang="en">Filename="DB8E2708-7938-427F-9D57-
      E2174ACD4C4A_009V0000.pdf" ContentType=application/pdf Compressed=No
      LargeAttachment=No OriginalBase64=Yes Length=234876</eb:Description>
    </eb:Reference>
</eb:Manifest>
```

Figure 4 - HL7 plus external data as sent

```
<eb:Manifest SOAP:mustUnderstand="1" eb:version="2.0">
    <eb:Reference xlink:href="cid:40fc95c5-5757-11de-9a31-
    eb35c6cf0e0a@spine.nhs.uk">
      <eb:Schema eb:location="http://www.nhsia.nhs.uk/schemas/HL7-
      Message.xsd" eb:version="1.0"/>
      <eb:Description xml:lang="en">HL7 payload</eb:Description>
      <hl7ebxml:Payload style="HL7" encoding="XML" version="3.0"/>
```

[2] See NWG Memo: *Content-ID and Message-ID Uniform Resource Locators* [Ref: 21]

```
    </eb:Reference>

    <eb:Reference eb:id="_3C64633D-291E-4077-AA47-886FD7FC64E3"
    xlink:href="mid:1A60E99A-C40E-44E6-9CDA-296D36423267">
      <eb:Description xml:lang="en">Filename="EEB93039-4285-4937-
      AEDB-18844C14DC9inps.co.ukVision3.gzip" ContentType=text/xml
      Compressed=No LargeAttachment=No OriginalBase64=No
      Length=302448 DomainData="X-GP2GP-Skeleton:
      Yes"</eb:Description>
    </eb:Reference>

    <eb:Reference eb:id="_0F6A1B08-9CE7-455D-B2FD-4EF8DE4D9C49"
    xlink:href="mid:A64A1C94-F913-455C-897A-740060CEE67A">
      <eb:Description xml:lang="en">Filename="DB8E2708-7938-427F-
      9D57-E2174ACD4C4A_009V0000.pdf" ContentType=application/pdf
      Compressed=No LargeAttachment=No OriginalBase64=Yes
      Length=234876</eb:Description>
    </eb:Reference>
</eb:Manifest>
```

Figure 5 - HL7 plus external data as received

Figure 6 shows the complete disaggregation in a diagram for relationships between messages and message fragments which holds true for all scenarios of CCLM.

In the situation where the Core HL7 exceeds 5MB, which is thought to be a very small probability, the ebXML manifest will point to either:

- Two (or more) message ids containing the core HL7

In this case, an EHR Extract 'Skeleton' message (see later) is sent containing no clinical information but with a Manifest section listing all other messages and attachments comprising the EHR Extract.

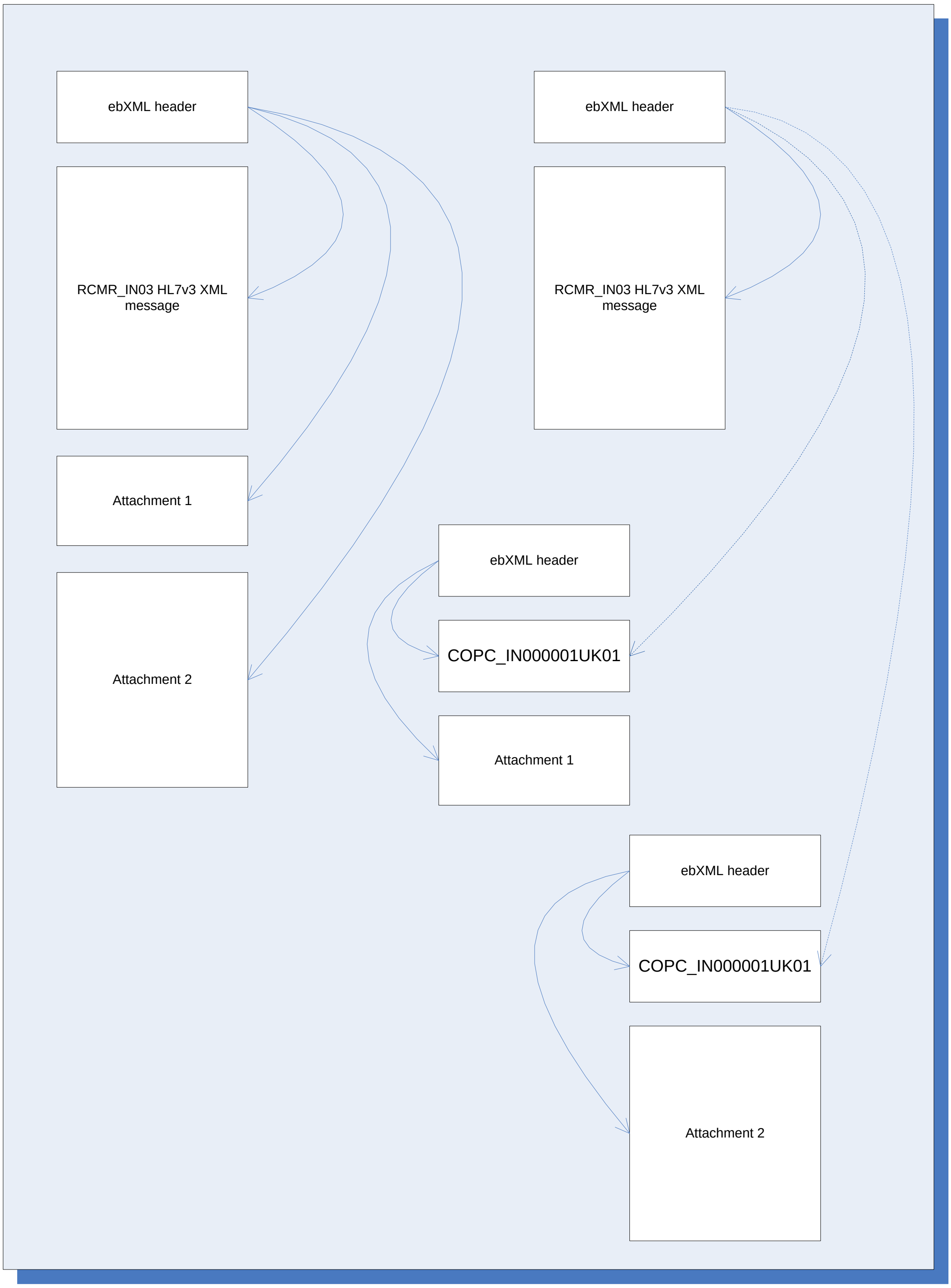


Figure 6 - CCLM disaggregation of attachments

2.3.1. Transmission and Attachment information

Applications which use attachments, such as GP2GP and the Choose and Book "referral" message, typically include the original filename. During transmission, other information is also required to ensure that the receiving system can reassemble the Large Message correctly.

This is carried in the manifest references' <eb:Description> elements, which are part of the ebXML specification. Figure 4 and Figure 5 give examples of how this is done, as a series of case-sensitive key=value pairs:

Key	Description
Filename	The original filename for any attachment. Typically this will be the name by which structured information in the HL7 message refers to the attachment.
ContentType	The original MIME type of the attachment data.
Compressed	Whether the large message pre-processing compressed the attachment. Values Yes or No
LargeAttachment	If the attachment itself is "large". Values Yes or No.
OriginalBase64	The CCLM process sends attachments as base64 encoded streams. If the "original" for the attachment was already base64 this is not done, but the "OriginalBase64" value is Yes. Otherwise No.
Length[3]	Size in bytes of the original attachment. This should be set after any compression and base64 encoding (i.e. the contribution that the attachment would make to the content length of a message where it was sent in-line).
DomainData	Only included when the core HL7 Extract is over 5MB.

2.3.2. Understanding a CCLM reference

The extract in Figure 7 contains a reference which we can split up and look at in detail:

```
<eb:Reference eb:id="_3C64633D-291E-4077-AA47-886FD7FC64E3"
xlink:href="mid:A64A1C94-F913-455C-897A-740060CEE67A">
    <eb:Description xml:lang="en">Filename="DB8E2708-7938-427F-9D57-
    E2174ACD4C4A_009V0000.pdf" ContentType=application/pdf
    Compressed=No LargeAttachment=No OriginalBase64=Yes
    Length=234876</eb:Description>
</eb:Reference>
```

Figure 7 - Detail of <eb:Reference> element in CCLM with small attachment

The extract was taken from a GP2GP EhrExtract RCMR_IN030000UK06 message. In a "non-large messaging" example the PDF file would have been attached in-line, and the reference would have used a "cid:" URI scheme to point to the content-id of the attachment containing that file.

Term	Description
xlink:href="mid:A64A1C94-F913-455C-897A-740060CEE67A"	This attachment is sent separately, as an in-line attachment to a COPC_IN000001UK01 message with HL7 message id A64A1C94-

[3] The "length" attribute added on the suggestion of Manuel Reyes, from EMIS.

Term	Description
	F913-455C-897A-740060CEE67A
Filename="DB8E2708-7938-427F-9D57-E2174ACD4C4A_009V0000.pdf"	The original filename was DB8E2708-7938-427F-9D57-E2174ACD4C4A_009V0000.pdf – note that the Filename will be quoted to support filenames that contain spaces.
ContentType=application/pdf	The MIME type of the original attachment was application/pdf
Compressed=No	The CCLM pre-processor has not compressed this attachment, so when the receiver reassembles the large message, it should not try to decompress this file.
LargeAttachment=No	The PDF file is not "large" and is transmitted as a single in-line attachment in one Common Point to Point message.
OriginalBase64=Yes	The original source of the PDF attachment, had already base64 encoded it. *Note: The aim is to reconstruct the input GP2GP message, in which case the reassembler needs to know whether it had to base64 encode an attachment. It is expected expect that most binary content is base64 encoded - and so to have this value set to "Yes"*
Length=234876	The base64 encoded PDF file is 234,876 bytes long.
DomainData	There is no domain-specific information in this example.

Note that "OriginalBase64" is mainly of use in the case where the CCLM process consumes a conventionally-structured message which has been assembled regardless of its being too large to send over Spine. It is intended to allow a receiver to re-create the large message as-sent. CCLM allows the message in parts to be checked before assembly.

The CCLM process records the length of an attachment, but does not use it itself. "Length" is included as information for a receiver.

2.3.3. Large Single Attachments

So far the principle of sending attachments separately is straightforward. However it fails in the case where a single attachment is "large" – over the Spine 5MB limit. CCLM handles this by splitting the attachment itself and sending each of the fragments as a Common Point to Point message attachment. In this case, the first fragment is referenced from the main manifest (Figure 8) of the EHR Extract.

This is where the "LargeAttachment" information is used:

```
<eb:Reference eb:id="_EDED40DA-677F-4A6B-8EAE-5AFD9ADD97F3"
xlink:href="mid:D3A7EDD9-4CA4-4ED6-99B9-7E305C248ED0">

    <eb:Description xml:lang="en">Filename="6BF2B25D-86F1-4781-9FC8-
    50B710B93C05_009H5000.mpg" ContentType=video/mpeg Compressed=No
    LargeAttachment=Yes OriginalBase64=Yes
    Length=7265112</eb:Description>
```

```
</eb:Reference>
```

Figure 8 - Detail of <eb:Reference> element in CCLM with large attachment

The Common Point to Point message with the HL7 message id of D3A7EDD9-4CA4-4ED6-99B9-7E305C248ED0 carries the first fragment of the large attachment. The ebXML manifest of that message references the separate Common Point to Point messages that carry the rest of the fragments. The "Length" shows the size of the intact attachment, after the base64 encoding.

2.3.4. Use of Compression

The CCLM protocol allows attachments to be compressed for transmission but it does not mandate it. Whether to compress a given attachment or not is for the message sender (or the Large Message pre-processor) to decide. If it does decide to compress, then this will be indicated by the manifest description containing "Compressed=Yes" – otherwise it will contain "Compressed=No".

Looking at the example from above, again:

```
<eb:Reference eb:id="_EDED40DA-677F-4A6B-8EAE-5AFD9ADD97F3"
xlink:href="mid:D3A7EDD9-4CA4-4ED6-99B9-7E305C248ED0">

    <eb:Description xml:lang="en">Filename="6BF2B25D-86F1-4781-9FC8-
    50B710B93C05_009H5000.mpg" ContentType=video/mpeg Compressed=No
    LargeAttachment=Yes OriginalBase64=Yes
    Length=7265112</eb:Description>
</eb:Reference>
```

Figure 9 - Detail of <eb:Reference> element in CCLM and use of compression

In this case the sender elected not to compress the attachment. Why did it decide so? In the general case we do not know. We might guess here that compressing an MPEG video file would be pointless (it is already compressed) or might even make it larger[4]. In the case of a large XML document, the decision might well be different. But we have no reason to care. CCLM receivers just read the "Compressed" value and act accordingly.

Compression where it is performed MUST be done to the GZIP format.

2.3.5. Acknowledgements

The "continue" message sent by the Requester on receipt of the EHR Extract core message is a Common Point to Point message containing an HL7 positive acknowledgement to the EHR core message. The Sending system must expect this response and be aware of the difference between this COPC Common Point to Point message and the MCCI Application Acknowledgement message. An example of the COPC Common Point to Point "continue" message is provided in Appendix 1.

At the time of writing, the message definition in EIS [Ref: 20] for Common Point to Point (COPC_IN000001UK01) states that there is an asynchronous response of urn:nhs:names:services:cc:MCCI_IN010000UK13. This must not be used and the equivalent on the gp2gp service must be used instead - urn:nhs:names:services:gp2gp:MCCI_IN010000UK13. The Requesting system will return these to the sender for each Common Point to Point message they receive. The Sending system will not return an Acknowledgement to the "Continue" message.

2.3.6. EHR Transfer Timeouts

Spine asynchronous messages are sent using the ebXML "reliable messaging" specification, in which the effective "time out" for a message transmission is given by the persistDuration contract property. This is the minimum time a reliable message is persisted by the transport – after which the transmission attempt may be

[4] This is also likely to be the case with files from the latest versions of the Microsoft Office suite – these are zip-compressed XML files and are likely to grow if a further attempt is made to compress them. Attempting to compress a strongly-encrypted file is similarly pointless.

considered to have failed. Spine TMS applies persistDuration to each message independently, from the time carried in the message's own "Timestamp" element in the ebXML header.

For a CCLM transmission with many parts, or being sent from a busy message handler, the time stamps of the primary HL7 and various associated Common Point to Point messages may differ widely and for this reason the application timeout for the overall transfer of the EHR Extract using Large Messaging is to be dynamically calculated. The calculation to be used is:

```
Timeout [secs]= (A x persistDuration contract property of EHR Response
[secs]) + (B x Number of COPC Common Point to Point EHR messages x
persistDuration contract property of COPC Common Point to Point messages
[secs])
```

The timeout for Transfers which do not use Large Messaging will just be the persistDuration value for the EHR Response message.

A & B are weighting factors associated with general message transmission delays and volume based throughput times to allow adjustment if required outside of the ebXML contract properties.

The RCMR EHR Request Completed (EHR Extract) message's own "Timestamp" element in the ebXML header will be used as the basis of comparison to see if the timeout has been reached.

The configuration items A and B must be only maintainable by supplier support staff. Values of these parameters will be defined during testing and implementation but should be assumed to be 1 if not provided.

Timeouts for the individual COPC Common Point to Point messages carrying the EHR messages will be determined by the contract properties of message and domain.

The calculated ebXML timeout is final and the dynamic timeout duration calculations are to account for timeouts of the overall GP2GP EHR Transfer. If the Requesting system calculates that the overall transfer has timed out, it negatively acknowledges (Response code 25) any further COPC Common Point to Point message parts arriving after the timeout is reached.

2.3.7. Retries

Failure of a Common Point to Point EHR fragment will be detected by receipt of a negative Application Acknowledgement (NACK) to a message. The Sending system must NOT support retries at the application level as the ebXML contract properties will already have provided the message level retry functionality.

The systems will record the Response code for each Common Point to Point message to support the GP2GP Paper Transfer Decommissioning requirements.

2.3.8. Support for ConversationID

All messages within the CCLM dialogue must support the use of <eb:ConversationID> element in the ebXML header to allow monitoring and fault resolution. All messages in the CCLM dialogue are to utilise the conversation ID supplied in the originating EHR Request. Conversation ID is to be propagated to all acknowledgements and resends.

3. System Requirements

3.1. System Configuration

Req ID	Requirement Text	Priority
LM01	The Sending system ***shall*** maintain the following configuration settings for each General Practice deployment instance: The TMS Maximum Message size (currently 5MB) The TMS Maximum Number of attachments limit (currently 99) GP2GP Maximum Attachment size (currently 4.5MB) (Note: These are required to future proof the Large Messaging Protocol if there is a change in the Spine transport contract properties in future.).	MUST
LM02	The configuration settings above ***shall*** only be changed by supplier support staff and ***shall*** only be changed at the request of the Authority.	MUST
LM03	The system ***shall*** have 2 configurable Timeout modification settings to control the timeout period before an overall GP2GP EHR Transfer is considered to have expired, i.e. it is an application timeout setting. These settings, known as A and B, ***shall*** be used in addition to the contract properties of the EHR Response and Common Point to Point interactions. A is a decimal number used to modify the Timeout period for the EHR Response message and ***shall*** be set to 1.0 initially. B is a decimal number used to modify the Timeout period for the set of Common Point to Point messages used in a Large Message transfer and ***shall*** be set to 1.0 initially. The systems ***shall not*** use A and B to modify the persistDuration of any single EHR Response or Common Point to Point message.	MUST
LM04	The Requesting and Sending systems ***shall*** both calculate the 'EHR Transfer Timeout' period as defined below: Timeout = (A x EHR timeout) + (B x No. of COPCs x COPC Timeout) - "EHR timeout" is the persistDuration on the EHR Extract's contract properties - "No. of COPCs" is the total number of Common Point to Point messages listed in the EHR Extract headers and inside any of the Common Point to Point messages where these have been subdivided - "COPC Timeout" is the persistDuration on the Common Point to Point's contract properties The systems ***shall*** calculate the timeout in seconds. (Note that the persistDuration setting on SDS can be in seconds, minutes or hours) and ***shall*** compare this to the Timestamp element in the ebXML header of the EHR Extract. The date and time to compare against ***shall*** be the RCMR EHR Request Completed (EHR Response) message's own "Timestamp" element in the ebXML header.	MUST

Req ID	Requirement Text	Priority
LM05	Access to the system configuration settings ***shall*** be restricted to supplier support staff. The values of these settings ***shall*** only be changed at the request of the Authority.	MUST

3.2. Sending System Requirements

3.2.1. When to use Large Messaging

The following requirements apply to the GP2GP Large Messaging solution:

Req ID	Requirement Text	Priority
LM06	The Sending system ***shall*** determine if an EHR Extract requires the use of Large Messaging before sending the EHR Response message. Large Messaging is required when an EHR Extract has: A. An EHR Extract (HL7 payload) and attachments size that is greater than the TMS maximum message size (currently 5MB). B. Any individual attachment with a size greater than the TMS maximum message size. C. More attachments than the TMS maximum attachments limit (currently 99). D. An EHR Extract (HL7 payload) size that is greater than the TMS maximum message size. E. One or more attachments that uses a Spine unsupported MIME type. Consult the Spine External Interface Specification [Ref: 20]) for list of supported MIME types. The Sending system ***shall*** handle any combination of the above situations. NB If options B or D are true, then option A must also be true.	MUST
LM07	Where the Sending system determines the EHR Extract requires the use of Large Messaging it ***shall*** determine if the Requesting system supports the Large Messaging Protocol, as defined in this document, before attempting to send the EHR Response or related Common Point to Point messages. The Sending system ***shall*** use SDS to determine support for the Large Messaging Protocol in the Requesting system. This is documented in the Spine Technical Design [Ref: 19].	MUST
LM08	If the Large Messaging Protocol is NOT supported by the Requesting system, the Sending system ***shall*** return an Application Acknowledgement with the Response code 14 – see Response Codes [Ref: 18]. The GP2GP EHR Transfer process ends.	MUST
LM09	The implementation of the Large Messaging solution ***shall*** be independent of the MIM version of the GP2GP EHR Request, Extract and Application Acknowledgment messages supported, i.e. Large Messaging can be used whether the EHR Request is either a MIM 3 or DMS 1 / MIM 7 message.	MUST

3.2.2. Message Orchestration

The Sending system needs to determine the way in which the EHR Extract is going to be sent, i.e. what combination of EHR Response and Common Point to Point (P2P) messages will be used before sending anything. The EHR Response message must always be sent first and the Manifest section must include HL7 message ID references for all other parts of the EHR Extract, i.e. as a ‘cid’ reference if included within the EHR Response message or a ‘mid’ reference if included in a P2P message.

Note that the ebXML message, which has its own ID and timestamp, must not be created at this time as they will be queued and sent in due course and it is therefore possible that they could be sent after their timeout period (=timestamp+persistduration).

Req ID	Requirement Text	Priority
LM10	The Sending system ***shall*** use the following algorithm to determine how to send the EHR Extract in the most efficient manner: Set $max_message_size to current maximum Spine message size limit Set $max_message_attachments to current maximum Spine message attachment number limit IF the core HL7 payload is < $max_message_size THEN BEGIN Send the HL7 payload in the EHR Response message AND Include as many attachments in the EHR Response message until $max_message_size or $max_message_attachments have been included (whichever is reached first) and send any remaining attachments in individual P2P messages (see Large Attachments requirement below) END ELSE { the core HL7 payload is > $max_message_size } BEGIN Send a Skeleton EHR Extract in EHR Response message AND Compress the HL7 payload using GZIP and send via P2P message(s) AND Include as many attachments in the EHR Response message until $max_message_size or $max_message_attachments have been included (whichever is reached first) and send any remaining attachments in individual P2P messages (see Large Attachments requirement below) END. This will ensure that the minimum number of P2P messages is used. Note that it is not possible to compress multiple attachments into a single file as the EHR Response Manifest reference does not support multiple attachments in a single attached file.	MUST
LM11	Large Attachments: If any single attachment exceeds the Spine maximum message size limit the Sending system ***shall*** compress it using GZIP if it is in a format that will significantly reduce the size of the file once compressed or if not in a compressible format, the Sending system ***shall*** send it in chunks using multiple P2P messages. If it is still greater than the Spine maximum message size limit once compressed, the Sending system ***shall*** send it in chunks using multiple P2P messages. The Sending system ***shall*** create chunks by splitting the attachment in a bitwise fashion.	MUST

Req ID	Requirement Text	Priority
LM12	If the HL7 payload is < the Spine maximum message size limit the Sending system ***shall*** send the core HL7 payload under the same interaction as a non-Large Messaging situation. NB this ***shall*** be either MIM 3 or DMS 1/MIM 7 depending on the MIM version of the received EHR Request.	MUST
LM13	The Sending system ***shall*** send the EHR Response before any P2P messages and ***shall*** wait for a "CONTINUE" message from the Requesting system before sending any Common Point to Point messages. This ***shall*** not affect normal retry behaviour.	MUST
LM14	The Sending system ***shall*** not send an Application Acknowledgement in response to the "CONTINUE" message.	MUST
LM15	The Sending system ***shall*** ONLY include ONE attachment (or file) in each Common Point to Point message.	MUST

3.2.3. Message Referencing

All references from the manifest section of the EHR Response message must use HL7 messages IDs.

Req ID	Requirement Text	Priority
LM16	All P2P messages sent as part of the EHR Extract ***shall*** have corresponding HL7 message ID references included in the Manifest section of the EHR Response message (see earlier section on encoding such references).	MUST
LM17	If a file is split into chunks and sent using multiple P2P messages the Manifest section of the message containing the first chunk ***shall*** include HL7 references to the remaining messages containing the other chunks.	MUST

3.2.4. Message Tracking

Req ID	Requirement Text	Priority
LM18	The Sending system ***shall*** treat each Common Point to Point message as a separate TMS message and maintain individual contract properties (e.g. retry behaviours, timeouts etc) for each message. Any errors reported for a message ***shall*** be logged and ***shall not*** halt the transmission of any unsent P2P messages.	MUST
LM19	The Sending system ***shall*** use the status of the final Application Acknowledgement to the EHR Response to record the integration success on the Requesting system. (See GP2GP Paper Transfer Decommissioning section within the GP2GP R2.2 Requirements Specification).	MUST
LM20	The Sending system ***shall*** record the positive or negative Application Acknowledgement along with the response code for each Common Point to Point message for use in reporting and the GP2GP Paper Transfer Decommissioning requirements.	MUST
LM21	The Sending system ***shall*** calculate the EHR Transfer Timeout when its MHS creates the EHR Response message for sending using the timestamp of the ebXML message as the base time.	MUST

Req ID	Requirement Text	Priority
LM22	If the EHR Transfer Timeout period is reached before all of the EHR Extract (including all Common Point to Point messages) have been sent and no errors have been received, the Sending system ***shall*** update the status of the EHR Transfer to 'EHR Extract Send Failure' and ***shall*** update the status of the EHR Transfer on the patient record.	MUST

3.2.5. Attachments

Req ID	Requirement Text	Priority
LM23	The Sending system ***shall*** include the original properties of each attachment within the EHR Extract message or the Common Point to Point message as appropriate. The properties ***shall*** be added to the manifest inside the eb:Description element. The properties and their formats ***shall*** be those defined in the examples in this document. E.g. Filename, Compressed etc.	MUST
LM24	The Sending system ***shall*** include these key-value pairs in the eb:Description tag for each Common Point to Point message: - Filename - Original ContentType - Whether the file is compressed - Whether the attachment is large - The original length of the file - DomainData DomainData ***shall*** only be included when the HL7 is itself larger than Spine maximum message size (LM01 D). In this case the entry will be DomainData="X-GP2GP-Skeleton: Yes".	MUST
LM25	Removed	Removed
LM26	If the Sending system compresses an attachment for transmission it ***shall*** set the 'Compressed' key value pair to Yes, if not it ***shall*** set it to No.	MUST
LM27	The Sending system ***shall not*** send any attachment compressed where the compressed file is larger than the uncompressed file, including files that have already been compressed as compression again will probably lead to a larger file. The Sending system ***shall not*** compress any attachment where the file is encrypted. Common compressed file types include MP3, MPEG, PNG, GIF, JPEG.	MUST
LM28	The Sending system ***shall*** use the GZIP format when compression is used on an attachment or the EHR Extract.	MUST
LM29	The Sending system ***shall*** send Spine unsupported MIME type attachments as application/octet-stream.	MUST
LM29.1	Sending/requesting systems ***shall*** split/reassemble large attachments in the order in which they are listed in the manifest.	MUST

Req ID	Requirement Text	Priority
LM29.2	Requesting systems ***shall*** handle attachment fragments being received outside the order specified in the manifest as the Spine Forward Reliable channel provides no guarantee of delivery order.	MUST

3.2.6. Error Handling

Req ID	Requirement Text	Priority
LM30	The Sending system ***shall*** not retry to send a Common Point to Point message on receipt of a negative Application Acknowledgement.	MUST

3.3. Requesting System Requirements

3.3.1. Responding to the EHR Reponse

Req ID	Requirement Text	Priority
LM31	The implementation of the Large Messaging solution ***shall*** be independent of the MIM version of the GP2GP EHR Request sent, i.e. Large Messaging can be used whether the EHR Request is either a MIM 3 or DMS 1/MIM 7 message.	MUST
LM32	The Requesting system ***shall*** on receipt of an EHR Response determine if Large Messaging is being used to transfer the EHR Extract by checking the manifest section for external references, i.e. URIs containing “mid:” which indicates associated Large Messages (i.e. Common Point to Point messages) to follow.	MUST
LM33	When Large Messaging is being used, i.e. there are Common Point to Point messages to follow, the Requesting system ***shall*** send a “CONTINUE” message to the Sending system. The “CONTINUE” message ***shall*** contain an Application Acknowledgement as demonstrated in Appendix 1. If a fatal error occurs, the Application Acknowledgement ***shall*** be returned with the appropriate error code – see Response Codes [Ref: 18].	MUST
LM34	The Requesting system ***shall*** treat each Common Point to Point message as an individual message and respond with individual positive or negative Application Acknowledgements as required.	MUST
LM35	If the system fails to receive a Common Point to Point Message for whatever reason or fails to reconstruct a chunked file the system ***shall*** return a negative Application Acknowledgement with Response code 31 to the EHR Response message (See Response Codes [Ref: 18]). The negative Application Acknowledgement ***should*** be returned after the receipt of all expected Common Point to Point messages.	MUST
LM36	The Requesting system ***shall*** calculate the EHR Transfer Timeout when an EHR Response message is received using the timestamp of the ebXML message as the base time.	MUST

Req ID	Requirement Text	Priority
LM37	If the EHR Transfer Timeout period is reached before all of the EHR Extract (including all Common Point to Point messages) have been received, the Requesting system ***shall*** send a negative Application Acknowledgement in response to the EHR Response message with Response code 25 (see Response Codes [Ref: 18] and ***shall*** update the status of the EHR Transfer on the patient record.	MUST
LM38	The Requesting system ***shall*** utilise the conversation ID supplied in the originating EHR Request message in all messages in the dialogue for the transmission of the EHR Extract including the Common Point to Point messages.	MUST

3.3.2. EHR Extract Integration

Req ID	Requirement Text	Priority
LM39	The Requesting system ***shall*** utilise the original properties of the attachments contained within the manifests of the EHR Extract message or the Common Point to Point message as appropriate in order to re-instate the attachments. This is especially important to attachments that have been transferred as application/octet-stream or another format where the original MIME type is unsupported by the SPINE.	MUST
LM40	The Requesting system ***shall*** re-construct attachments from the key-value pairs in the eb:Description tag for each Common Point to Point message: - Filename - Original ContentType - Whether the file is compressed - Whether the attachment is large - The original length of the file.	MUST
LM41	The Requesting system ***shall*** process the GZIP format when compression has been used on an attachment or the EHR Extract, i.e. uncompress it to restore the original uncompressed file together with its original filename.	MUST
LM42	The Requesting system ***shall*** prevent the user from integrating the EHR Extract until all the elements including Common Point to Point messages have been successfully received.	MUST
LM43	The Requesting system ***shall*** keep the GP2GP Transfer Status on the patient's record up-to-date as its status changes and provide information to the user as described in the 'GP2GP Transfer Status Recording' section of the GP2GP R2.2 Requirements Specification.	MUST

3.4. Requesting and Sending System Requirements

Req ID	Requirement Text	Priority

Req ID	Requirement Text	Priority
LM44	After compliance has been awarded, the supplier ***shall*** add the MIM 7 Common Point to Point message entries under the 'gp2gp' service to the ASID and MHS entries on SDS for each system configured to support Large Messaging. The entries ***shall*** reflect both the sending and receipt of these messages.	MUST
LM45	The system ***shall*** use the conversation ID in the originating EHR Request message in all subsequent messages, including all Common Point to Point messages, relating to the patient's EHR Transfer.	MUST
LM45.1	Systems sending Common Point to Point Messages shall ensure these messages match the schema and Schematron rules provided by the Authority.	MUST

3.5. Management Information Requirements

Req ID	Requirement Text	Priority
LM46	The Requesting and Sending systems ***shall*** record audit and Management Information for the actions in this specification as required by the Harvesting Management Information supplementary specification [Ref: 15] and Use Case 2 Transfer and analyse management information [Ref: 3].	MUST

Appendix 1: COPC Common Point to Point “continue” message

The following message provides an example of the HL7 attachment for the COPC Common Point to Point message sent by the Requesting system, in acknowledgement of the RCMR EHR Transfer Completed (EHR Extract) message, indicating that the Common Point to Point EHR fragments can follow.

Note the sections highlighted below which identify the COPC as a “continue” message and provide the identity of the message acknowledged (as a “continue”).

```
<hl7:PayloadInformation xmlns:npfitct="template:NPFIT:content"
xmlns:gp="urn:nhs:names:services:gp2gp"
xmlns:npfitlc="NPFIT:HL7:Localisation"
xmlns="urn:hl7-org:v3"
xmlns:hl7="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
classCode="OBS" moodCode="EVN"
xsi:schemaLocation="urn:hl7-org:v3 ../../Schemas/COPC_MT000001UK01.xsd
urn:nhs:names:services:gp2gp ../../Schemas/GP2GP_LM.xsd">
    <code code="GP2GP_PI" codeSystem="2.16.840.1.113883.2.1.3.2.4.17.202"
displayName="GP2GP Payload Information"/>
    <id root="FA039330-7E63-446A-8CA4-E9E0D00DA6E8"/>
    <npfitlc:messageType root="2.16.840.1.113883.2.1.3.2.4.18.17"
extension="COPC_MT000001UK01"/>
    <value>
        <gp:Gp2gpfragment>
            <gp:Version>01</gp:Version>
            <gp:Recipients>
                <gp:Recipient>B83002</gp:Recipient>
            </gp:Recipients>
            <gp:From>C81007</gp:From>
            <gp:subject>Continue Acknowledgement</gp:subject>
            <gp:message-id>FA039330-7E63-446A-8CA4-E9E0D00DA6E8</gp:message-
id>
        </gp:Gp2gpfragment>
    </value>
    <pertinentInformation typeCode="PERT">
        <sequenceNumber value="1"/>
        <pertinentPayloadBody moodCode="EVN" classCode="OBS">
            <code code="GP2GP_PB"
codeSystem="2.16.840.1.113883.2.1.3.2.4.17.202" displayName="GP2GP Payload
Body"/>
            <id root="FA039330-7E63-446A-8CA4-E9E0D00DA6E8"/>
            <value>
                <gp:Gp2gpfragment>
                    <Message xmlns="urn:hl7-org:v3" type="Message">
                        <id root="FA039330-7E63-446A-8CA4-E9E0D00DA6E8"/>
                        <code code="0"
codeSystem="2.16.840.1.113883.2.1.3.2.4.17.101" displayName="Continue"/>
                        <creationTime value="201009201130"/>
                        <versionCode code="V3NPfIT3.1.09"/>
                        <interactionId root="2.16.840.1.113883.2.1.3.2.4.12"
extension="MCCI_IN010000UK13"/>
                        <processingCode code="P"/>
                        <processingModeCode code="T"/>
                        <acceptAckCode code="NE"/>
                        <acknowledgement typeCode="AA"><acknowledgementDetail
typeCode="IF"><code code="0"
codeSystem="2.16.840.1.113883.2.1.3.2.4.17.101"
displayName="Continue"/></acknowledgementDetail>s
                                             <messageRef><id root="6E242658-
3D8E-11E3-A7DC-172BDA00FA67"/></messageRef></acknowledgement>
                        <communicationFunctionRcv>
                            <device>
                                <id root="1.2.826.0.1285.0.2.0.107"
extension="715373337545"/>
                            </device>
                        </communicationFunctionRcv>
                        <communicationFunctionSnd>
                            <device>
                                <id root="1.2.826.0.1285.0.2.0.107"
extension="276827251543"/>
                            </device>
                        </communicationFunctionSnd>
                    </Message>
                    <gp:acknowledgedMessage>
                        <gp:id root="6E242658-3D8E-11E3-A7DC-172BDA00FA67"/>
                    </gp:acknowledgedMessage>
                </gp:Gp2gpfragment>
            </value>
        </pertinentPayloadBody>
    </pertinentInformation>
</hl7:PayloadInformation>
```

Figure 10 – Example of COPC Continue message in a Large EHR Extract scenario

Appendix 2: A CCLM Transmission Walk-Through

A reference system for CCLM has been constructed by the CfH National Integration Centre (NIC) and is available for inspection, demonstrations, and test use. This "walk-through" is based on that reference system. Its structure is shown in Figure 11 – Diagrammatic view CCLM reference system.

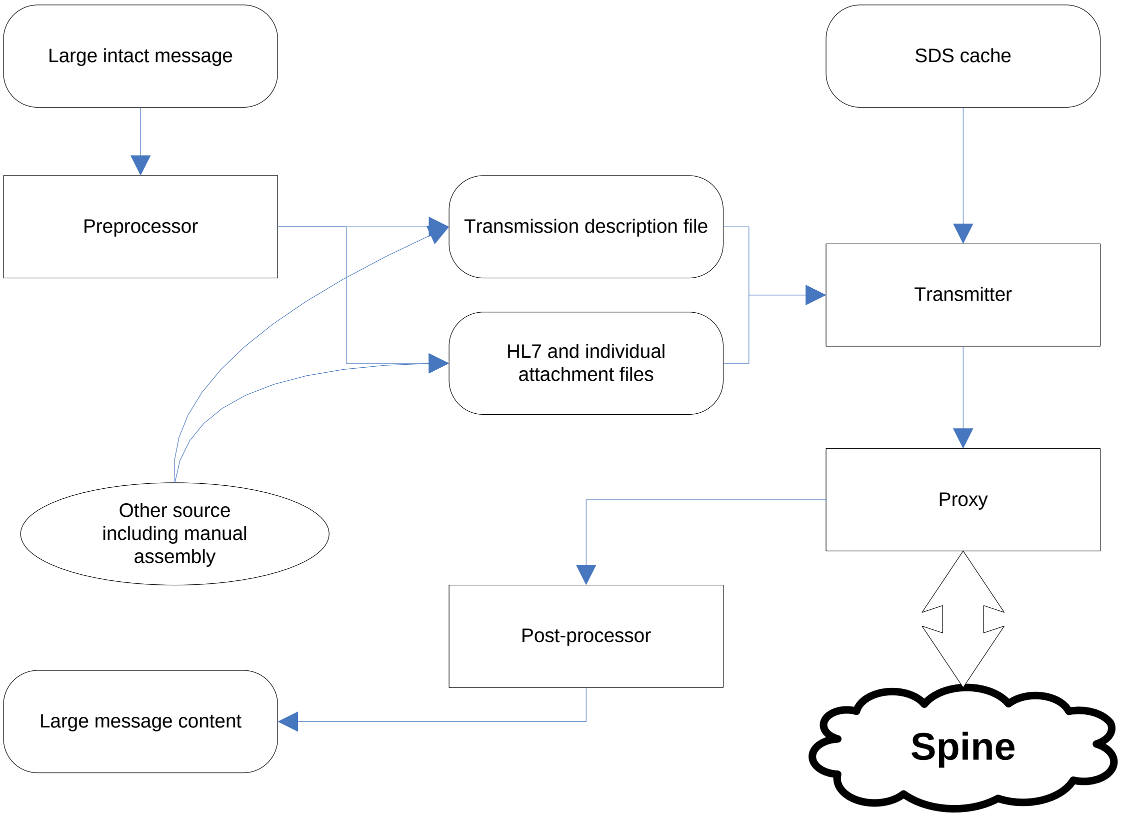

Large intact message
SDS cache
Preprocessor
Transmission description file
Transmitter
HL7 and individual attachment files
Proxy
Other source including manual assembly
Post-processor
Large message content
Spine

Figure 11 – Diagrammatic view CCLM reference system

AP2.1 Content

The content starts as an EHR Extract payload including zero[5] or more attachment files any number of which, including the HL7 EhrExtract message (the core HL7 payload) may exceed the 5MB size limit of TMS. In the case of a pre-prepared large message, a pre-processor exists to split the message into its attachments. The message transmission system uses a "description file" to hold details of the content to be sent. For the large message input case, the pre-processor writes both the description file, and the individual attachment files.

AP2.2 Transmission

Message addressing uses conventional Spine mechanisms and is based on the message type to be sent, and the "receiver ASID" in the primary HL7 message. Data from an SDS cache is used to resolve destination party id, other contract property details, and the URL to which the messages will be sent. The process is summarised in Figure 12.

The message transmitter reads the description file and identifies the HL7 part of the message. It does the SDS lookups and makes the SOAP/ebXML header for the HL7 message transmission. As it does so it inspects the details in the description file for each of the attachments, and constructs a "Common Point to Point" message, with the file as an attachment, based on those details. It records the description file details, and the HL7 message id of the Common Point to Point message. It writes the primary HL7 message' ebXML manifest entries based on those details.

[5] Where the HL7 XML itself is large (i.e. larger than current Spine message size limit), a mechanism for packaging it for transmission using the CCLM protocol is presented later in this document.

Assembly of the HL7 message and its ebXML wrapper complete, it opens a connection to the proxy, which forwards it to the message handler URL – typically this is the “forward reliable” Spine URL and GP2GP will use “forward reliable”.

Each of the Common Point to Point messages, with its attachment, is then sent again through the proxy.

In the case where an attachment is indicated to be “large”, it splits that attachment and makes a separate Common Point to Point message for each fragment. It is the HL7 message id of the first of these fragments that is recorded in the manifest of the primary HL7 message transmission.

All these Common Point to Point messages are again, sent through the proxy.

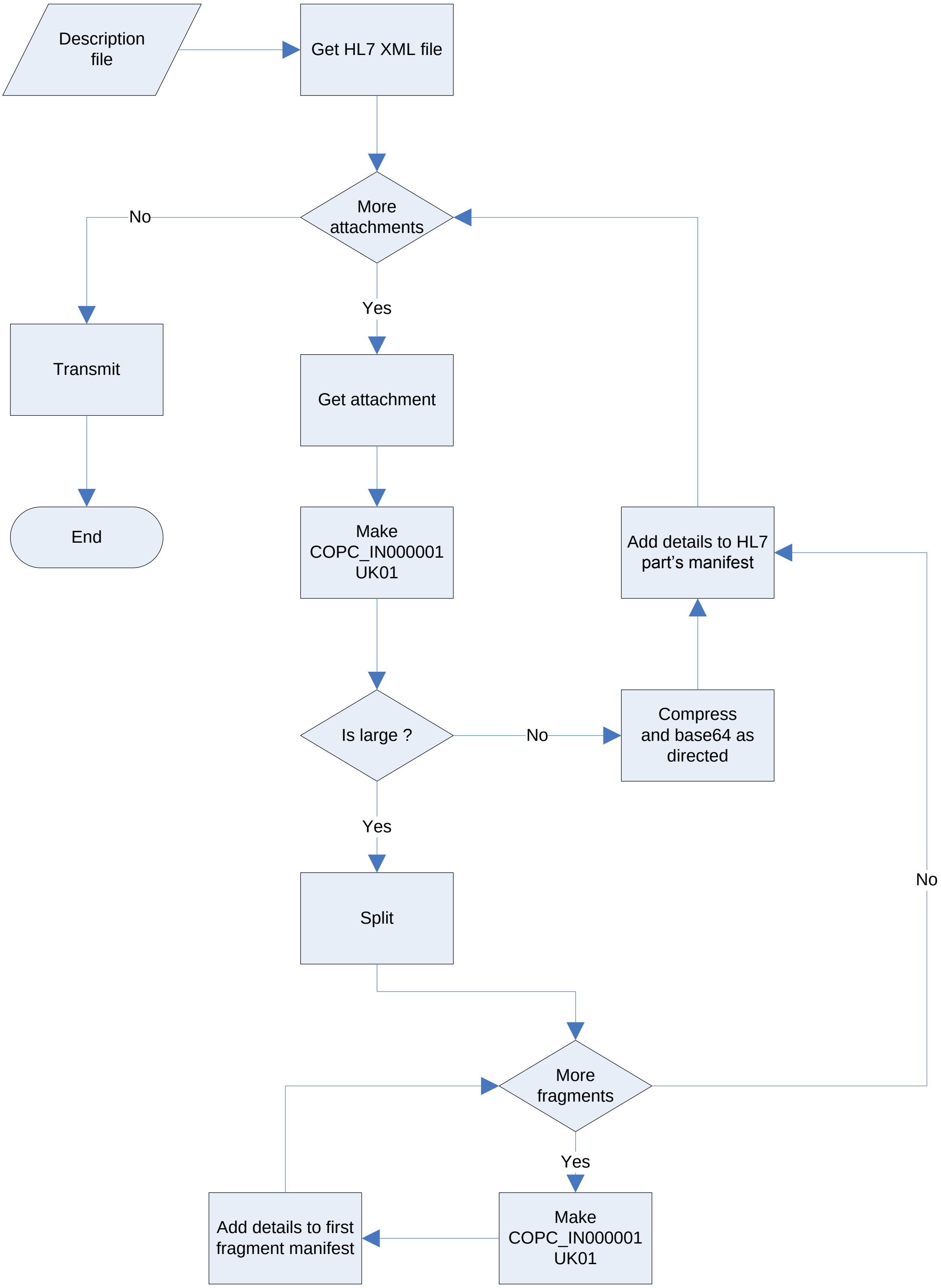


Figure 12 - Reference system transmission process

AP2.3 Receipt

The NIC messaging proxy handles both in- and out-bound messages. A different instance is used for receipt in our example. On receipt of the primary HL7 message, the inbound data is written to a log file. On receiving the Common Point to Point messages, these are also written to a log file but an MCCI_IN010000UK13 is returned to the sender, acknowledging receipt of the Common Point to Point message. The process is summarised in Figure 13.

Once everything has been received, a post-processor extracts the core HL7 payload from the EHR Response message. The post-processor then works through the references from the message' ebXML manifest. For each reference with a "mid:" URI scheme, it identifies the associated Common Point to Point message. Where the reference description indicates that the Common Point to Point attachment is compressed or base64 encoded, it decompresses and/or decodes the attachment first. Where the reference description indicates that the Common Point to Point attachment is the first of a sequence of fragments of a large attachment, it processes the ebXML manifest of that Common Point to Point message in the same way, and reassembles the parts to make the original attachment.

In all cases, attachment files are extracted and written to the original file name, as given in the "Filename" data from the reference. A report is written to drive the primary HL7 acknowledgment content.

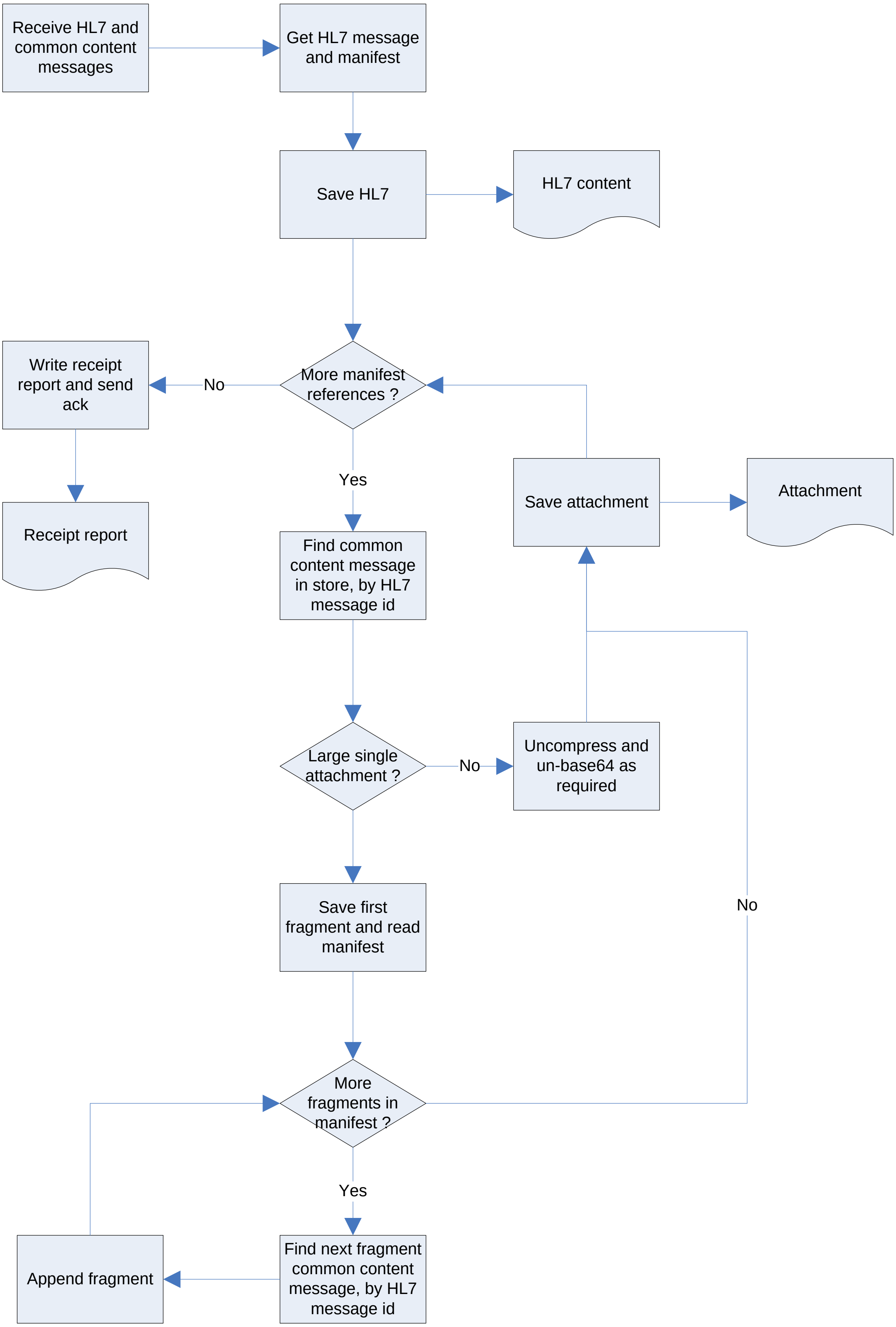

Receive HL7 and common content messages
Get HL7 message and manifest
Save HL7
HL7 content
Write receipt report and send ack
No
More manifest references ?
Yes
Save attachment
Attachment
Receipt report
Find common content message in store, by HL7 message id
Large single attachment ?
No
Uncompress and un-base64 as required
Save first fragment and read manifest
No
More fragments in manifest ?
Yes
Append fragment
Find next fragment common content message, by HL7 message id

Figure 13 - Reference system receiver process

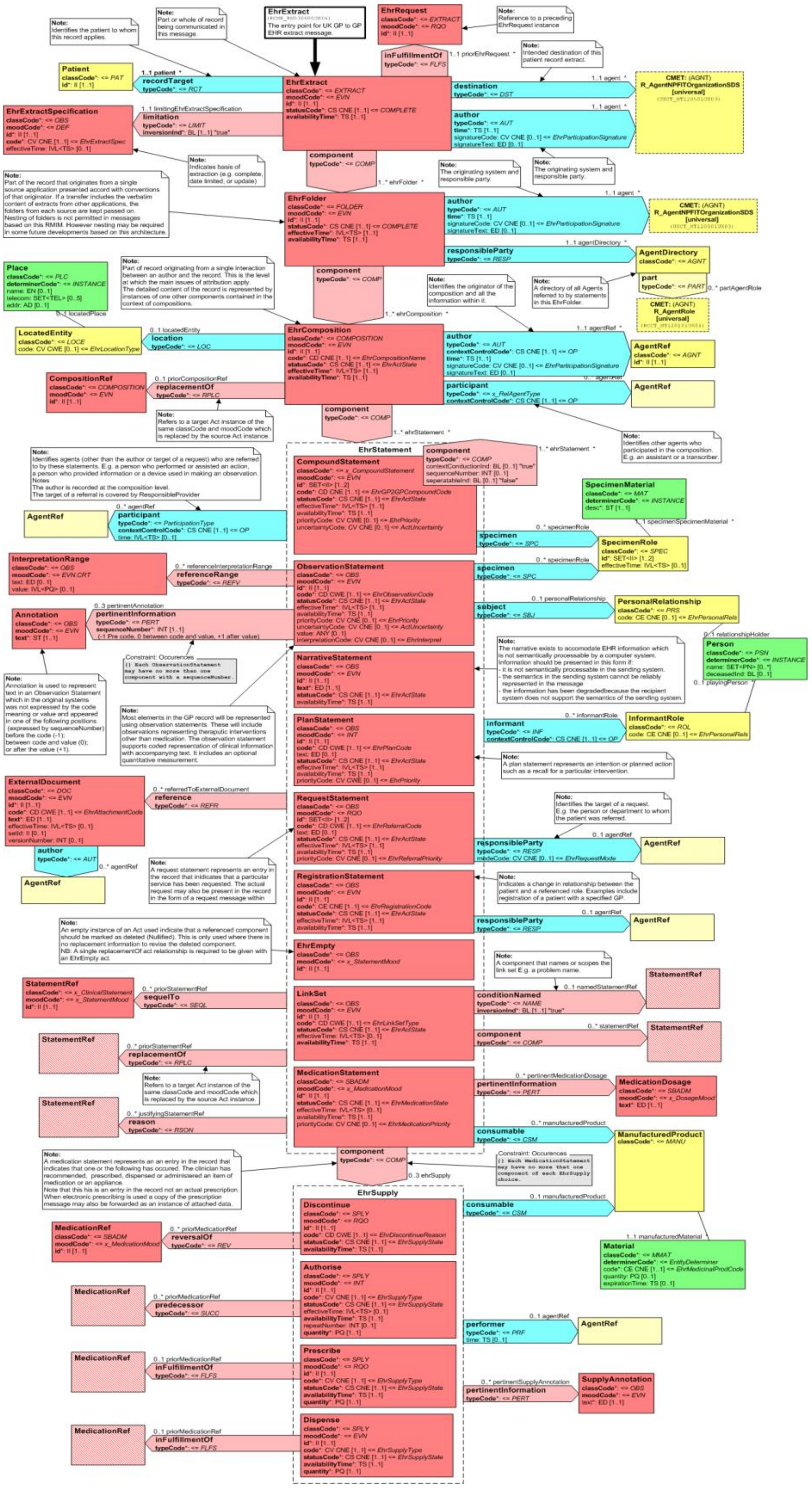
AP2.4 Large HL7 payloads

So far the CCLM protocol has covered the sending of EHR Extracts which are large due to having a number of attachments, and how to send attachments which are “large” of themselves. Because CCLM is designed to work with currently-implemented Spine messages, and transmission mechanisms it relies on a “primary HL7 part” to be the core transmission, and to carry the information necessary for reassembly by the receiver.

But what happens if the HL7 part itself is “large”? Whilst most examples of Spine messages are under 50K, XML tends to the verbose. And by their nature, HL7v3 XML documents of the sort defined for NPfIT are more verbose than most, because of their extreme degree of nested structure. So it is not unreasonable that a single HL7 document in the clinical messaging or laboratory reporting spaces (GP2GP, PSIS, Pathology and possibly even Choose & Book referrals) may be “large”.

Other than ease of implementation, the main reason CCLM uses a “primary HL7 part” is that to do so allows receiving systems to get a similar trigger event as they would with a non-CCLM message. To preserve this, anything that is done to the HL7 payload MUST still look like the real message to the extent of being identified in the same way, and subject to the same validation constraints.

How to achieve this is dependent on the HL7 messages to be exchanged. In the GP2GP domain, the mechanism relies on the general structure of the EhrExtract message, being a set of an arbitrary number of clinical statements. A clinical statement may have further statements nested within it, and/or be related to others in the extract. A statement may also refer to an external document – which is how the HL7 message references message attachments. The clinical statement set is surrounded by authorship information, and a structure which describes the patient’s care history. To handle the “large EhrExtract” case, the extract is compressed and added as an attachment. The authorship information is retained for a “skeleton extract”, but the clinical content is replaced by a single statement that refers to the compressed attachment containing the real message. Figure 14 and Figure 15 show the RCMR_IN03 message from MIM 3.1.10 to illustrate the difference between the real and the “skeleton” forms.



Note:
Identifies the patient to whom this record applies.
Note:
Part or whole of record being communicated in this message.
EhrExtract
The entry point for UK GP to GP EHR extract message.
EhrRequest
classCode*: <= EXTRACT
moodCode*: <= RQO
id*: II [1..1]
Note:
Reference to a preceding EhrRequest instance
inFulfillmentOf
typeCode*: <= FLFS
1..1 priorEhrRequest *
Note:
Intended destination of this patient record extract.
Patient
classCode*: <= PAT
id*: II [1..1]
1..1 patient *
recordTarget
typeCode*: <= RCT
EhrExtract
classCode*: <= EXTRACT
moodCode*: <= EVN
id*: II [1..1]
statusCode*: CS CNE [1..1] <= COMPLETE
availabilityTime*: TS [1..1]
destination
typeCode*: <= DST
1..1 agent *
CMET: (AGNT)
R_AgentNPFITOrganizationSDS
[universal]
EhrExtractSpecification
classCode*: <= OBS
moodCode*: <= DEF
id*: II [1..1]
code*: CV CNE [1..1] <= EhrExtractSpec
effectiveTime: IVL<TS> [0..1]
1..1 limitingEhrExtractSpecification
limitation
typeCode*: <= LIMIT
inversionInd*: BL [1..1] "true"
author
typeCode*: <= AUT
time*: TS [1..1]
signatureCode: CV CNE [0..1] <= EhrParticipationSignature
signatureText: ED [0..1]
1..1 agent *
Note:
Indicates basis of extraction (e.g. complete, date limited, or update)
component
typeCode*: <= COMP
1..* ehrFolder *
Note:
The originating system and responsible party.
Note:
The originating system and responsible party.
Note:
Part of the record that originates from a single source application presented accord with conventions of that originator. If a transfer includes the verbatim content of extracts from other applications, the folders from each source are kept passed on. Nesting of folders is not permitted in messages based on this RMIM. However nesting may be required in some future developments based on this architecture.
EhrFolder
classCode*: <= FOLDER
moodCode*: <= EVN
id*: II [1..1]
statusCode*: CS CNE [1..1] <= COMPLETE
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
author
typeCode*: <= AUT
time*: TS [1..1]
signatureCode: CV CNE [0..1] <= EhrParticipationSignature
signatureText: ED [0..1]
1..1 agent *
CMET: (AGNT)
R_AgentNPFITOrganizationSDS
[universal]
responsibleParty
typeCode*: <= RESP
1..1 agentDirectory *
AgentDirectory
classCode*: <= AGNT
part
typeCode*: <= PART
0..* partAgentRole
CMET: (AGNT)
R_AgentRole
[universal]
Place
classCode*: <= PLC
determinerCode*: <= INSTANCE
name: EN [0..1]
telecom: SET<TEL> [0..5]
addr: AD [0..1]
0..1 locatedPlace
Note:
Part of record originating from a single interaction between an author and the record. This is the level at which the main issues of attribution apply.
The detailed content of the record is represented by instances of one other components contained in the context of compositions.
component
typeCode*: <= COMP
1..* ehrComposition *
Note:
Identifies the originator of the composition and all the information within it.
Note:
A directory of all Agents referred to by statements in this EhrFolder.
LocatedEntity
classCode*: <= LOCE
code: CV CWE [0..1] <= EhrLocationType
0..1 locatedEntity
location
typeCode*: <= LOC
EhrComposition
classCode*: <= COMPOSITION
moodCode*: <= EVN
id*: II [1..1]
code*: CD CNE [1..1] <= EhrCompositionName
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
author
typeCode*: <= AUT
contextControlCode*: CS CNE [1..1] <= OP
time*: TS [1..1]
signatureCode: CV CNE [0..1] <= EhrParticipationSignature
signatureText: ED [0..1]
1..1 agentRef *
AgentRef
classCode*: <= AGNT
id*: II [1..1]
0..* agentRef
participant
typeCode*: <= x_RelAgentType
contextControlCode*: CS CNE [1..1] <= OP
AgentRef
CompositionRef
classCode*: <= COMPOSITION
moodCode*: <= EVN
id*: II [1..1]
0..1 priorCompositionRef
replacementOf
typeCode*: <= RPLC
Note:
Refers to a target Act instance of the same classCode and moodCode which is replaced by the source Act instance.
component
typeCode*: <= COMP
1..* ehrStatement *
Note:
Identifies other agents who participated in the composition. E.g. an assistant or a transcriber.
EhrStatement
component
typeCode*: <= COMP
contextConductionInd: BL [0..1] "true"
sequenceNumber: INT [0..1]
seperatableInd: BL [0..1] "false"
1..* ehrStatement *
Note:
Identifies agents (other than the author or target of a request) who are referred to by these statements. E.g. a person who performed or assisted an action, a person who provided information or a device used in making an observation.
Notes
The author is recorded at the composition level.
The target of a referral is covered by ResponsibleProvider
CompoundStatement
classCode*: <= x_CompoundStatement
moodCode*: <= EVN
id*: SET<II> [1..2]
code*: CD CNE [1..1] <= EhrGP2GPCompoundCode
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
priorityCode: CV CWE [0..1] <= EhrPriority
uncertaintyCode: CV CNE [0..1] <= ActUncertainty
SpecimenMaterial
classCode*: <= MAT
determinerCode*: <= INSTANCE
desc*: ST [1..1]
1..1 specimenSpecimenMaterial *
AgentRef
0..* agentRef
participant
typeCode*: <= ParticipationType
contextControlCode*: CS CNE [1..1] <= OP
time: IVL<TS> [0..1]
0..* specimenRole
specimen
typeCode*: <= SPC
SpecimenRole
classCode*: <= SPEC
id*: SET<II> [1..2]
effectiveTime: IVL<TS> [0..1]
InterpretationRange
classCode*: <= OBS
moodCode*: <= EVN.CRT
text: ED [0..1]
value: IVL<PQ> [0..1]
0..* referenceInterpretationRange
referenceRange
typeCode*: <= REFV
ObservationStatement
classCode*: <= OBS
moodCode*: <= EVN
id*: II [1..1]
code*: CD CWE [1..1] <= EhrObservationCode
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
priorityCode: CV CNE [0..1] <= EhrPriority
uncertaintyCode: CV CNE [0..1] <= ActUncertainty
value: ANY [0..1]
interpretationCode: CV CNE [0..1] <= EhrInterpret
0..* specimenRole
specimen
typeCode*: <= SPC
0..1 personalRelationship
subject
typeCode*: <= SBJ
PersonalRelationship
classCode*: <= PRS
code: CE CNE [0..1] <= EhrPersonalRels
0..1 relationshipHolder
Annotation
classCode*: <= OBS
moodCode*: <= EVN
text*: ST [1..1]
0..3 pertinentAnnotation
pertinentInformation
typeCode*: <= PERT
sequenceNumber*: INT [1..1]
(-1 Pre code, 0 between code and value, +1 after value)
Constraint: Occurences
() Each ObservationStatement may have no more than one component with a sequenceNumber.
NarrativeStatement
classCode*: <= OBS
moodCode*: <= EVN
id*: II [1..1]
text*: ED [1..1]
statusCode*: CS CNE [1..1] <= EhrActState
availabilityTime*: TS [1..1]
Note:
The narrative exists to accomodate EHR information which is not semantically processable by a computer system.
Information should be presented in this form if:
- it is not semantically processable in the sending system.
- the semantics in the sending system cannot be reliably represented in the message
- the information has been degradedbecause the recipient system does not support the semantics of the sending system.
Person
classCode*: <= PSN
determinerCode*: <= INSTANCE
name: SET<PN> [0..*]
deceasedInd: BL [0..1]
0..1 playingPerson *
Note:
Annotation is used to represent text in an Observation Statement which in the original systems was not expressed by the code meaning or value and appeared in one of the following positions (expressed by sequenceNumber) before the code (-1); between code and value (0); or after the value (+1).
Note:
Most elements in the GP record will be represented using observation statements. These will include observations representing theraputic interventions other than medication. The observation statement supports coded representation of clinical information with accompanying text. It includes an optional quantitative measurement.
PlanStatement
classCode*: <= OBS
moodCode*: <= INT
id*: II [1..1]
code*: CD CWE [1..1] <= EhrPlanCode
text: ED [0..1]
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
priorityCode: CV CWE [0..1] <= EhrPriority
0..* informantRole
informant
typeCode*: <= INF
contextControlCode*: CS CNE [1..1] <= OP
InformantRole
classCode*: <= ROL
code: CE CNE [0..1] <= EhrPersonalRels
Note:
A plan statement represents an intention or planned action such as a recall for a particular intervention.
ExternalDocument
classCode*: <= DOC
moodCode*: <= EVN
id*: II [1..1]
code*: CD CWE [1..1] <= EhrAttachmentCode
text*: ED [1..1]
effectiveTime: IVL<TS> [0..1]
setId: II [0..1]
versionNumber: INT [0..1]
0..* referredToExternalDocument
reference
typeCode*: <= REFR
RequestStatement
classCode*: <= OBS
moodCode*: <= RQO
id*: SET<II> [1..2]
code*: CD CWE [1..1] <= EhrReferralCode
text: ED [0..1]
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
priorityCode: CV CNE [0..1] <= EhrReferralPriority
Note:
Identifies the target of a request.
E.g. the person or department to whom the patient was referred.
0..1 agentRef
responsibleParty
typeCode*: <= RESP
modeCode: CV CNE [0..1] <= EhrRequestMode
AgentRef
author
typeCode*: <= AUT
0..* agentRef
AgentRef
Note:
A request statement represents an entry in the record that inidicates that a particular service has been requested. The actual request may also be present in the record in the form of a request message within
RegistrationStatement
classCode*: <= OBS
moodCode*: <= EVN
id*: II [1..1]
code*: CE CNE [1..1] <= EhrRegistrationCode
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime*: IVL<TS> [1..1]
availabilityTime*: TS [1..1]
Note:
Indicates a change in relationship between the patient and a referenced role. Examples include registration of a patient with a specified GP.
0..1 agentRef
responsibleParty
typeCode*: <= RESP
AgentRef
Note:
An empty instance of an Act used indicate that a referenced component should be marked as deleted (Nullified). This is only used where there is no replacement information to revise the deleted component.
NB: A single replacementOf act relationship is required to be given with an EhrEmpty act.
EhrEmpty
classCode*: <= OBS
moodCode*: <= x_StatementMood
id*: II [1..1]
Note:
A component that names or scopes the link set E.g. a problem name.
StatementRef
classCode*: <= x_ClinicalStatement
moodCode*: <= x_StatementMood
id*: II [1..1]
0..* priorStatementRef
sequelTo
typeCode*: <= SEQL
LinkSet
classCode*: <= OBS
moodCode*: <= EVN
id*: II [1..1]
code*: CD CWE [1..1] <= EhrLinkSetType
statusCode*: CS CNE [1..1] <= EhrActState
effectiveTime: IVL<TS> [0..1]
availabilityTime*: TS [1..1]
0..1 namedStatementRef
conditionNamed
typeCode*: <= NAME
inversionInd*: BL [1..1] "true"
StatementRef
0..* statementRef
component
typeCode*: <= COMP
StatementRef
StatementRef
0..* priorStatementRef
replacementOf
typeCode*: <= RPLC
Note:
Refers to a target Act instance of the same classCode and moodCode which is replaced by the source Act instance.
StatementRef
0..* justifyingStatementRef
reason
typeCode*: <= RSON
MedicationStatement
classCode*: <= SBADM
moodCode*: <= x_MedicationMood
id*: II [1..1]
statusCode*: CS CNE [1..1] <= EhrMedicationState
effectiveTime: IVL<TS> [0..1]
availabilityTime*: TS [1..1]
priorityCode: CV CNE [0..1] <= EhrMedicationPriority
0..* pertinentMedicationDosage
pertinentInformation
typeCode*: <= PERT
MedicationDosage
classCode*: <= SBADM
moodCode*: <= x_DosageMood
text*: ED [1..1]
0..* manufacturedProduct
consumable
typeCode*: <= CSM
ManufacturedProduct
classCode*: <= MANU
Note:
A medication statement represents an an entry in the record that indicates that one or the following has occured. The clinician has recommended, prescribed, dispensed or administered an item of medication or an appliance.
Note that this his is an entry in the record not an actual prescription.
When electronic prescribing is used a copy of the prescription message may also be forwarded as an instance of attached data.
component
typeCode*: <= COMP
0..3 ehrSupply
Constraint: Occurences
() Each MedicationStatement may have no more that one component of each EhrSupply choice.
EhrSupply
0..1 manufacturedProduct
consumable
typeCode*: <= CSM
Discontinue
classCode*: <= SPLY
moodCode*: <= RQO
id*: II [1..1]
code*: CD CWE [1..1] <= EhrDiscontinueReason
statusCode*: CS CNE [1..1] <= EhrSupplyState
availabilityTime*: TS [1..1]
MedicationRef
classCode*: <= SBADM
moodCode*: <= x_MedicationMood
id*: II [1..1]
0..* priorMedicationRef
reversalOf
typeCode*: <= REV
1..1 manufacturedMaterial *
Material
classCode*: <= MMAT
determinerCode*: <= EntityDeterminer
code*: CE CNE [1..1] <= EhrMedicinalProdCode
quantity: PQ [0..1]
expirationTime: TS [0..1]
Authorise
classCode*: <= SPLY
moodCode*: <= INT
id*: II [1..1]
code*: CV CNE [1..1] <= EhrSupplyType
statusCode*: CS CNE [1..1] <= EhrSupplyState
effectiveTime: IVL<TS> [0..1]
availabilityTime*: TS [1..1]
repeatNumber: INT [0..1]
quantity*: PQ [1..1]
MedicationRef
0..* priorMedicationRef
predecessor
typeCode*: <= SUCC
0..1 agentRef
performer
typeCode*: <= PRF
time: TS [0..1]
AgentRef
Prescribe
classCode*: <= SPLY
moodCode*: <= RQO
id*: II [1..1]
code*: CV CNE [1..1] <= EhrSupplyType
statusCode*: CS CNE [1..1] <= EhrSupplyState
availabilityTime*: TS [1..1]
quantity*: PQ [1..1]
MedicationRef
0..1 priorMedicationRef
inFulfillmentOf
typeCode*: <= FLFS
0..* pertinentSupplyAnnotation
pertinentInformation
typeCode*: <= PERT
SupplyAnnotation
classCode*: <= OBS
moodCode*: <= EVN
text*: ED [1..1]
Dispense
classCode*: <= SPLY
moodCode*: <= EVN
id*: II [1..1]
code*: CV CNE [1..1] <= EhrSupplyType
statusCode*: CS CNE [1..1] <= EhrSupplyState
availabilityTime*: TS [1..1]
quantity*: PQ [1..1]
MedicationRef
0..1 priorMedicationRef
inFulfillmentOf
typeCode*: <= FLFS

Figure 14 - Full MIM 3.1.10 EhrExtract message

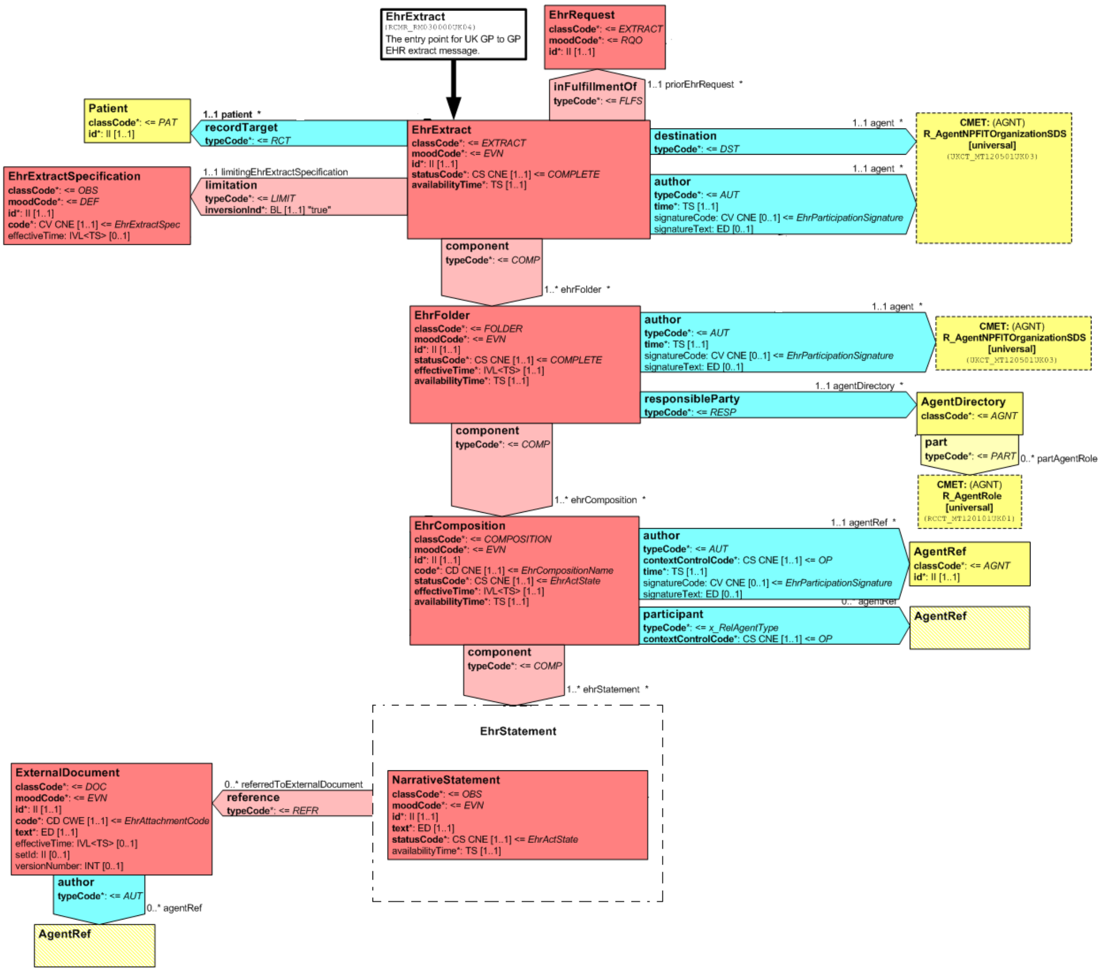


Figure 15 - Skeleton EhrExtract

Note that the skeleton structure is the same as that of the real message, until the "EhrStatement" – which in the skeleton contains only a "NarrativeStatement" referencing an external document. The GP2GP-specific CCLM post-processor understands how to handle this, and replaces the skeleton with an uncompressed copy of the original payload.

Critically, the skeleton is a valid form of the RCMR_IN030000UK03 message.

The example also illustrates "domain specific" reference information in CCLM. The reference to the "real" payload is shown in Figure 16 below:

```
<eb:Reference xlink:href="mid:1A60E99A-C40E-44E6-9CDA-296D36423267">
    <eb:Description xml:lang="en">Filename="EEB93039-4285-4937-AEDB-
    18844C14DC9inps.co.ukVision3.gzip" ContentType=text/xml
    Compressed=No LargeAttachment=No OriginalBase64=No Length=304751
    DomainData="X-GP2GP-Skeleton: Yes"</eb:Description>
</eb:Reference>
```

Figure 16 – Referencing large HL7 payloads

Note the "DomainData", which is interpreted in this case by GP2GP systems only.

AP2.5 Unsupported Content Types

In CCLM, the content type of an attachment is authoritatively carried in the reference description, in the "ContentType" field. This is because it is a decision the sender makes whether to compress an attachment (in which case the content type of the attachment is application/x-gzip or application/octet-stream. The EIS is ambiguous about content types – giving a list of allowed types, and supported encodings. The application/octet-stream is mentioned in the supported encodings table but not in the list of allowed types.

Carrying the content type in the reference, an object type not explicitly supported by EIS (for example a Microsoft Excel spreadsheet, or an OpenOffice document), can be shipped as application/octet-stream when it is attached to a COPC_IN010000UK01 message. When delivered, the real content type is then available to the CCLM receiver should that information be required.

Appendix 3: Decision Tracking

#	Decision	Reasoning
1	Determining support for CCLM in GP2GP	2 options were identified for this: 1) Use contract properties held within SDS. This approach meant additional work in having the COPC interaction added to the GP2GP service in SDS. Flexible TMS should allow this and this is the option being progressed with the NHS CFH Spine team. This option was chosen as it is in line with detection methods for other GP2GP interaction support. 2) Use the <eb:Description> to carry the Requestors CCLM status. This would keep the implementation and maintenance in the GP2GP supplier domain and it is not the first time this approach has been used.
2	Use of “continue” message	The CCLM solution required that an initial core message was acknowledged before the sending on Common Point to Point fragments. This prevents the high volume message parts entering TMS if the transfer is going to fail. The transfer may fail because: • The core message may be intercepted by Spine SEF as an illegitimate conversation. • Recipient may be off line • Message incorrectly addressed
3	Use of COPC message and the “continue” message	The CCLM solution needed to maintain the integrity of the meaning of the core message ACK / NACK irrespective of the use of CCLM or not, so the MCCI acknowledgement could not be used for the “continue”. It was felt that an MCCI NACK response code was inappropriate. The only other option was to send a positive acknowledgement in a COPC message so that it would not be parsed by TMS and could be processed differently by the receiving application.

Appendix 4: Structure of Common Point to Point Messages in GP2GP LM

The Common Point to Point Message is used for control and to carry attachments ‘overflowing’ the EHR Extract message. It is a simple generic message consisting of two HL7 classes: PayloadBody and PayloadInformation. Where there is an attachment with the message this is carried as a MIME attachment and not in the value attribute of these classes. The current monitoring and failed message solutions does not provide good visibility of MIME attachments or ebXML content and so monitoring and debugging information must be carried in the HL7 content; the PayloadInformation.value attribute is used for this.

Schematron

Schematron is the standard method of applying an additional layer of validation to HL7 messages. All GP2GP messages sent through the Common Point to Point COPC_IN000001UK01 interaction must be valid against the Schematron provided below.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron"
  schemaVersion="0.1" xmlns:gp="urn:nhs:names:services:gp2gp" xmlns:hl7="urn:hl7-org:v3">
  <sch:ns prefix="gp" uri="urn:nhs:names:services:gp2gp" />
  <sch:ns prefix="hl7" uri="urn:hl7-org:v3" />

  <sch:pattern id="PayloadInformation.value">
    <sch:title>Checking PayloadInformation.value</sch:title>
    <sch:rule context="hl7:PayloadInformation/hl7:value">
      <sch:assert test="gp:Gp2gpfragment"
        >The PayloadInformation.value element must have a Gp2gpfragment element in the GP2GP namespace</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:Version = '01'"
        >The Gp2gpfragment element must have a child element Version with value '01'</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:*[1][self::gp:Version]"
        >Version must be the first element within Gp2gpfragment</sch:assert>
      <sch:assert test="count(gp:Gp2gpfragment/gp:Recipients/gp:Recipient) = 1"
        >There must be a single Recipient element</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:*[2][self::gp:Recipients]"
        >Recipients must be the second element within Gp2gpfragment</sch:assert>
      <sch:assert test="count(gp:Gp2gpfragment/gp:From) = 1"
        >There must be a single From element</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:*[3][self::gp:From]"
        >From must be the third element within Gp2gpfragment</sch:assert>
      <sch:assert test="count(gp:Gp2gpfragment/gp:subject) = 1"
        >There must be a subject element</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:*[4][self::gp:subject]"
        >Subject must be the fourth element within Gp2gpfragment</sch:assert>
      <sch:assert test="count(gp:Gp2gpfragment/gp:message-id) = 1"
        >There must be a message-id element</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:message-id = /hl7:PayloadInformation/hl7:id/@root"
        >message-id element must match containing act ID (PayloadInformation/id/@root) and ebXML message ID
(soap:Envelope/soap:Header/ed:MessageHeader/eb:MessageData/eb:MessageId)</sch:assert>
      <sch:assert test="gp:Gp2gpfragment/gp:*[5][self::gp:message-id]"
        >message-id must be the fifth element within Gp2gpfragment</sch:assert>
    </sch:rule>
  </sch:pattern>

  <sch:pattern abstract="true" id="valid-ods-code">
    <sch:title>Checking ODS code format</sch:title>
    <sch:rule context="$ods_code">
      <!--regex only available in XPath 2, so unable to use matches(., '[A-Z0-9]*') -->
      <sch:assert
        test="
        string-length(.) &gt;= 3
        and string-length(.) &lt;= 12
        and normalize-space( translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890','')) = ''
        ">Element <sch:name /> content is not a valid ODS code format
      </sch:assert>
    </sch:rule>
  </sch:pattern>

  <sch:pattern id="recipient-ods-code" is-a="valid-ods-code">
    <sch:title>Checking Recipient ODS code format</sch:title>
    <sch:param name="ods_code"
      value="hl7:PayloadInformation/hl7:value/gp:Gp2gpfragment/gp:Recipients/gp:Recipient" />
  </sch:pattern>

  <sch:pattern id="sender-ods-code" is-a="valid-ods-code">
    <sch:title>Checking Sender ODS code format</sch:title>
    <sch:param name="ods_code"
      value="hl7:PayloadInformation/hl7:value/gp:Gp2gpfragment/gp:From" />
  </sch:pattern>

  <sch:pattern id="Payloadbody.value">
    <sch:title>Checking Payloadbody.value of GP2GP large message attachment</sch:title>
    <sch:rule context="hl7:pertinentPayloadBody/hl7:value">
      <sch:assert test="gp:Gp2gpfragment or hl7:reference"
        >Payloadbody.value must contain either a continue message or reference to attachment</sch:assert>
    </sch:rule>
  </sch:pattern>

  <sch:pattern id="Payloadbody.value-continue">
    <sch:title>Checking Payloadbody.value of CONTINUE message</sch:title>
    <sch:rule context="hl7:pertinentPayloadBody/hl7:value/gp:Gp2gpfragment">
      <sch:assert test="*[1][self::hl7:Message]"
        >Message must be the first element within Gp2gpfragment</sch:assert>
      <sch:assert test="count(hl7:Message[hl7:interactionId/@extension='MCCI_IN010000UK13']) = 1"
        >Payloadbody.value must contain an acknowledgement message</sch:assert>
      <sch:assert test="hl7:Message/hl7:acknowledgement/hl7:acknowledgementDetail/@typeCode = 'IF'"
        >AcknowledgementDetail typeCode must be 'IF' (For Information)</sch:assert>
      <sch:assert test="hl7:Message/hl7:acknowledgement/@typeCode = 'AA'"
        >Acknowledgement typeCode must be 'AA' (Application Acknowledgement)</sch:assert>
      <sch:assert test="gp:acknowledgedMessage"
        >Element acknowledgedMessage must be present</sch:assert>
      <sch:assert test=" gp:Gp2gpfragment/gp:message-id = ../hl7:id/@root "
        >acknowledgedMessage ID must match acknowledgement messageRef</sch:assert>
      <sch:assert test="*[2][self::gp:acknowledgedMessage]"
        >acknowledgedMessage must be the second element within Gp2gpfragment</sch:assert>
      <sch:assert test="gp:acknowledgedMessage/gp:id/@root"
        >ID for acknowledgedMessage must be provided</sch:assert>
    </sch:rule>
  </sch:pattern>

  <sch:pattern id="attachment-reference">
    <sch:title>Checking attachment reference</sch:title>
    <sch:rule context="hl7:pertinentPayloadBody/hl7:value/hl7:reference">
      <sch:assert test="@value">Attachment reference must have a value</sch:assert>
    </sch:rule>
  </sch:pattern>

</sch:schema>
```

Schema

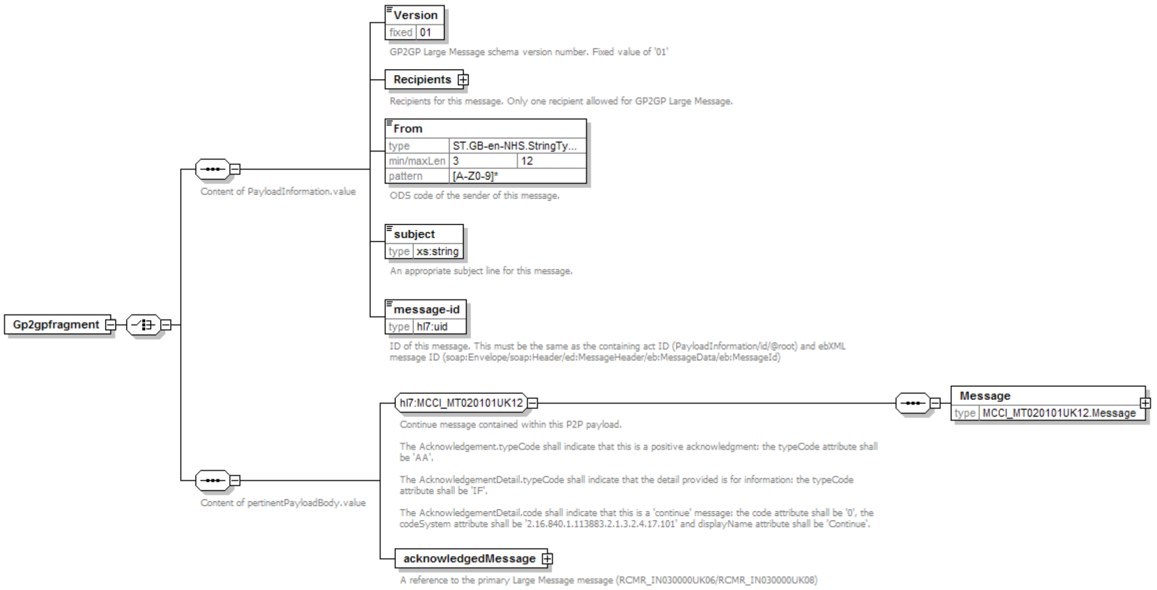
An XML Schema is provided below for the Gp2gpfragment in order to provide structure and documentation for the Gp2gpfragment in a familiar format. However there are difficulties in using XML Schema for validation. The HL7 datatype schema applies the processContent=strict attribute to the content of ED, meaning that this content will be ignored by the xml processor when validating the COPC_RM000001UK01 message as a whole, even where a schema is available. If the Gp2gpfragment elements are extracted then this schema may be used to validate these as separate documents.

XML Schema Documentation

schema location:	Schemas\GP2GP_LM.xsd
attribute form default:	**unqualified**
element form default:	**qualified**
targetNamespace:	**urn:nhs:names:services:gp2gp**

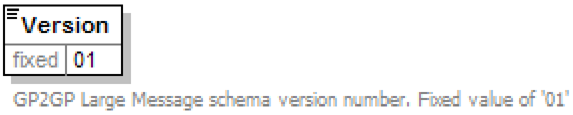
Elements
Gp2gpfragment

element **Gp2gpfragment**

diagram	
namespace	urn:nhs:names:services:gp2gp
properties	content complex
children	Version Recipients From subject message-id **Message** acknowledgedMessage

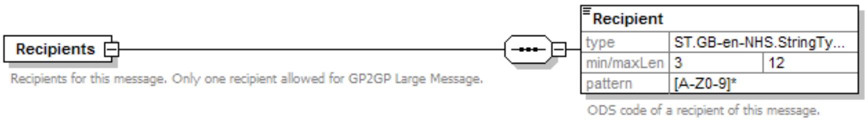
Version
fixed 01
GP2GP Large Message schema version number. Fixed value of '01'
Recipients
Recipients for this message. Only one recipient allowed for GP2GP Large Message.
From
type ST.GB-en-NHS.StringTy...
min/maxLen 3 12
pattern [A-Z0-9]*
ODS code of the sender of this message.
Content of PayloadInformation.value
subject
type xs:string
An appropriate subject line for this message.
Gp2gpfragment
message-id
type hl7:uid
ID of this message. This must be the same as the containing act ID (PayloadInformation/id/@root) and ebXML message ID (soap:Envelope/soap:Header/ed:MessageHeader/eb:MessageData/eb:MessageId)
hl7:MCCI_MT020101UK12
Message
type MCCI_MT020101UK12.Message
Continue message contained within this P2P payload.
The Acknowledgement.typeCode shall indicate that this is a positive acknowledgment: the typeCode attribute shall be 'AA'.
The AcknowledgementDetail.typeCode shall indicate that the detail provided is for information: the typeCode attribute shall be 'IF'.
The AcknowledgementDetail.code shall indicate that this is a 'continue' message: the code attribute shall be '0', the codeSystem attribute shall be '2.16.840.1.113883.2.1.3.2.4.17.101' and displayName attribute shall be 'Continue'.
Content of pertinentPayloadBody.value
acknowledgedMessage
A reference to the primary Large Message message (RCMR_IN030000UK06/RCMR_IN030000UK08)

element **Gp2gpfragment/Version**

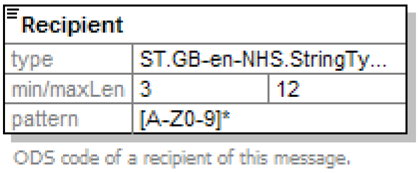
diagram	
namespace	urn:nhs:names:services:gp2gp
properties	isRef 0 fixed 01
annotation	documentation GP2GP Large Message schema version number. Fixed value of '01'

Version
fixed 01
GP2GP Large Message schema version number. Fixed value of '01'

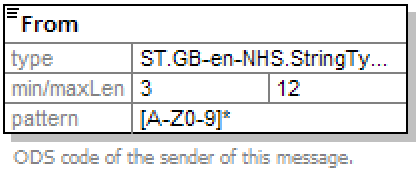
element **Gp2gpfragment/Recipients**

diagram	 Recipients Recipients for this message. Only one recipient allowed for GP2GP Large Message. Recipient type ST.GB-en-NHS.StringTy... min/maxLen 3 12 pattern [A-Z0-9]* ODS code of a recipient of this message.
namespace	urn:nhs:names:services:gp2gp
properties	isRef 0 content complex
children	Recipient
annotation	documentation Recipients for this message. Only one recipient allowed for GP2GP Large Message.

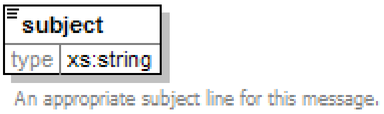
element **Gp2gpfragment/Recipients/Recipient**

diagram	 Recipient type ST.GB-en-NHS.StringTy... min/maxLen 3 12 pattern [A-Z0-9]* ODS code of a recipient of this message.
namespace	urn:nhs:names:services:gp2gp
type	**ST.GB-en-NHS.StringType10**
properties	isRef 0 content simple
facets	Kind Value annotation minLength 3 maxLength 12 pattern [A-Z0-9]*
annotation	documentation ODS code of a recipient of this message.

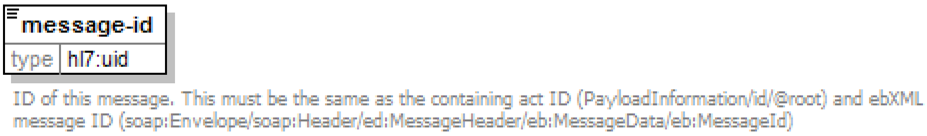
element **Gp2gpfragment/From**

diagram	 From type ST.GB-en-NHS.StringTy... min/maxLen 3 12 pattern [A-Z0-9]* ODS code of the sender of this message.
namespace	urn:nhs:names:services:gp2gp
type	**ST.GB-en-NHS.StringType10**
properties	isRef 0 content simple
facets	Kind Value annotation minLength 3 maxLength 12 pattern [A-Z0-9]*
annotation	documentation ODS code of the sender of this message.

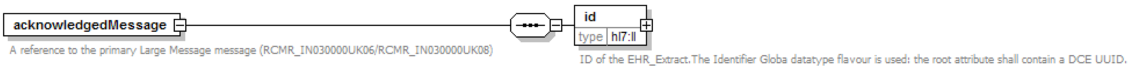
element **Gp2gpfragment/subject**

diagram	 subject type xs:string An appropriate subject line for this message.
namespace	urn:nhs:names:services:gp2gp
type	**xs:string**
properties	isRef 0 content simple
annotation	documentation An appropriate subject line for this message.

element **Gp2gpfragment/message-id**

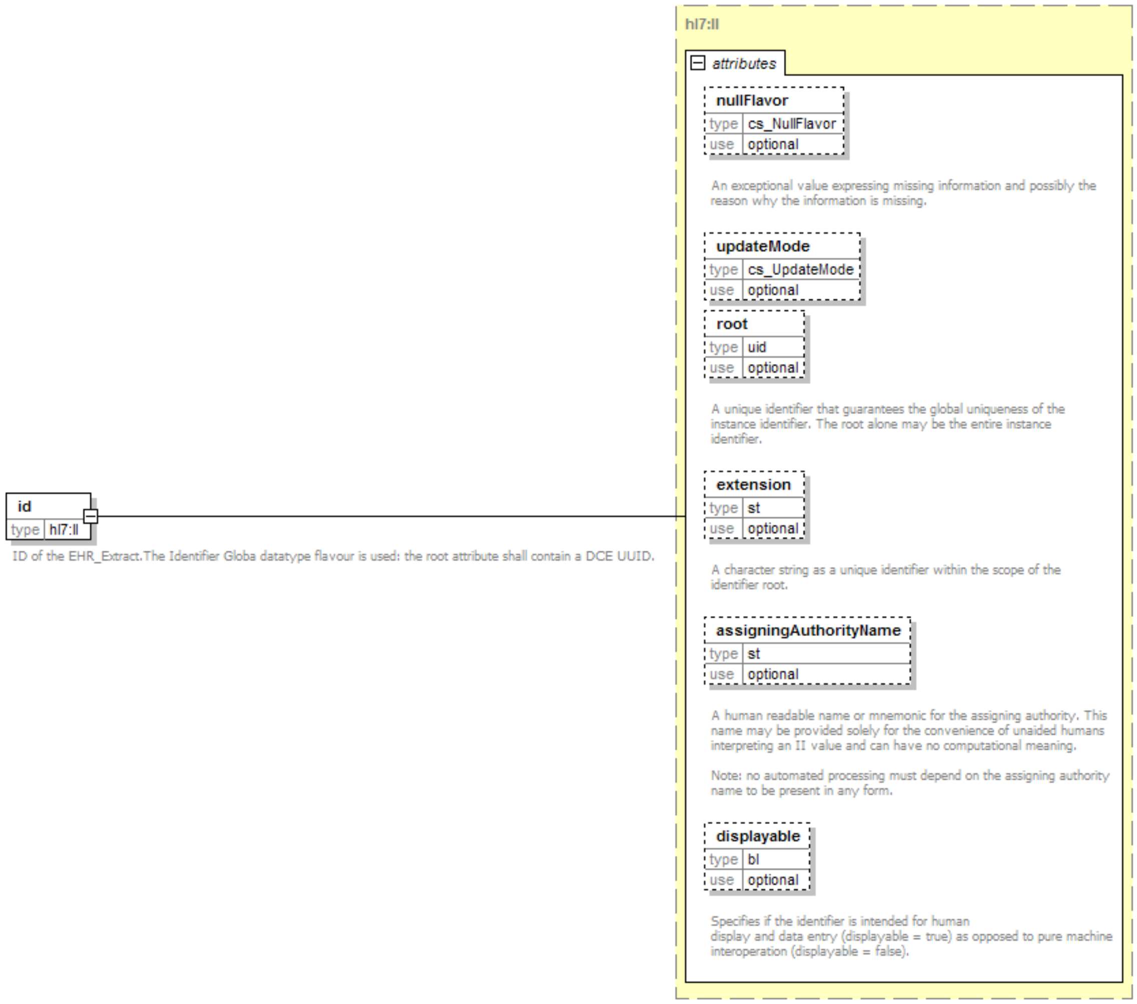
diagram	 message-id type hl7:uid ID of this message. This must be the same as the containing act ID (PayloadInformation/id/@root) and ebXML message ID (soap:Envelope/soap:Header/ed:MessageHeader/eb:MessageData/eb:MessageId)
namespace	urn:nhs:names:services:gp2gp
type	**hl7:uid**
properties	isRef 0 content simple
annotation	documentation ID of this message. This must be the same as the containing act ID (PayloadInformation/id/@root) and ebXML message ID (soap:Envelope/soap:Header/ed:MessageHeader/eb:MessageData/eb:MessageId)

element **Gp2gpfragment/acknowledgedMessage**

diagram	 acknowledgedMessage A reference to the primary Large Message message (RCMR_IN030000UK06/RCMR_IN030000UK08) id type hl7:II ID of the EHR_Extract.The Identifier Globa datatype flavour is used: the root attribute shall contain a DCE UUID.
namespace	urn:nhs:names:services:gp2gp
properties	isRef 0 content complex
children	id
annotation	documentation A reference to the primary Large Message message (RCMR_IN030000UK06/RCMR_IN030000UK08)

element **Gp2gpfragment/acknowledgedMessage/id**

diagram


id
type hl7:II
ID of the EHR_Extract.The Identifier Globa datatype flavour is used: the root attribute shall contain a DCE UUID.
hl7:II
attributes
nullFlavor
type cs_NullFlavor
use optional
An exceptional value expressing missing information and possibly the reason why the information is missing.
updateMode
type cs_UpdateMode
use optional
root
type uid
use optional
A unique identifier that guarantees the global uniqueness of the instance identifier. The root alone may be the entire instance identifier.
extension
type st
use optional
A character string as a unique identifier within the scope of the identifier root.
assigningAuthorityName
type st
use optional
A human readable name or mnemonic for the assigning authority. This name may be provided solely for the convenience of unaided humans interpreting an II value and can have no computational meaning.
Note: no automated processing must depend on the assigning authority name to be present in any form.
displayable
type bl
use optional
Specifies if the identifier is intended for human display and data entry (displayable = true) as opposed to pure machine interoperation (displayable = false).

namespace	urn:nhs:names:services:gp2gp
type	**hl7:II**
properties	isRef 0 content complex

attributes	Name	Type	Use	Default	Fixed	annotation
	nullFlavor	**hl7:cs_NullFlavor**	optional			documentation An exceptional value expressing missing information and possibly the reason why the information is missing.
	updateMode	**hl7:cs_UpdateMode**	optional			
	root	**hl7:uid**	optional			documentation A unique identifier that guarantees the global uniqueness of the instance identifier. The root alone may

				be the entire instance identifier.
	extension	hl7:st	optional	documentation A character string as a unique identifier within the scope of the identifier root.
	assigningAuthorityName	hl7:st	optional	documentation A human readable name or mnemonic for the assigning authority. This name may be provided solely for the convenience of unaided humans interpreting an II value and can have no computational meaning. Note: no automated processing must depend on the assigning authority name to be present in any form.
	displayable	hl7:bl	optional	documentation Specifies if the identifier is intended for human display and data entry (displayable = true) as opposed to pure machine interoperation (displayable = false).
annotation	documentation ID of the EHR_Extract.The Identifier Globa datatype flavour is used: the root attribute shall contain a DCE UUID.			

3.6. Source

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:nhs:names:services:gp2gp" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:hl7="urn:hl7-org:v3"
targetNamespace="urn:nhs:names:services:gp2gp" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="0.1">
  <xs:import namespace="urn:hl7-org:v3" schemaLocation="MCCI_MT020101UK12.xsd"/>
  <xs:element name="Gp2gpfragment">
    <xs:complexType>
      <xs:choice>
        <xs:sequence>
          <xs:annotation>
            <xs:documentation>Content of PayloadInformation.value</xs:documentation>
          </xs:annotation>
          <xs:element name="Version" fixed="01">
            <xs:annotation>
              <xs:documentation>GP2GP Large Message schema version number. Fixed value of '01'</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Recipients">
            <xs:annotation>
              <xs:documentation>Recipients for this message. Only one recipient allowed for GP2GP Large
Message.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Recipient" type="ST.GB-en-NHS.StringType10">
                  <xs:annotation>
                    <xs:documentation>ODS code of a recipient of this message.
                    </xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="From" type="ST.GB-en-NHS.StringType10">
            <xs:annotation>
              <xs:documentation>ODS code of the sender of this message.
              </xs:documentation>
            </xs:annotation>
            </xs:element>
            <xs:element name="subject" type="xs:string">
              <xs:annotation>
                <xs:documentation>An appropriate subject line for this message.
                </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="message-id" type="hl7:uid">
              <xs:annotation>
                <xs:documentation>ID of this message. This must be the same as the containing act ID
(PayloadInformation/id/@root) and ebXML message ID (soap:Envelope/soap:Header/ed:MessageHeader/eb:MessageData/eb:MessageId)
                </xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:sequence>
            <xs:annotation>
              <xs:documentation>Content of pertinentPayloadBody.value</xs:documentation>
            </xs:annotation>
            <xs:group ref="hl7:MCCI_MT020101UK12">
              <xs:annotation>
                <xs:documentation>Continue message contained within this P2P payload.

The Acknowledgement.typeCode shall indicate that this is a positive acknowledgment: the typeCode attribute shall be 'AA'.

The AcknowledgementDetail.typeCode shall indicate that the detail provided is for information: the typeCode attribute shall
be 'IF'.

The AcknowledgementDetail.code shall indicate that this is a 'continue' message: the code attribute shall be '0', the
codeSystem attribute shall be '2.16.840.1.113883.2.1.3.2.4.17.101' and displayName attribute shall be 'Continue'.
                </xs:documentation>
              </xs:annotation>
            </xs:group>
            <xs:element name="acknowledgedMessage">
              <xs:annotation>
                <xs:documentation>A reference to the primary Large Message message
(RCMR_IN030000UK06/RCMR_IN030000UK08)</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="id" type="hl7:II">
                    <xs:annotation>
                      <xs:documentation>ID of the EHR_Extract.The Identifier Globa datatype flavour is used: the root
```

```
attribute shall contain a DCE UUID.</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="ST.GB-en-NHS.StringType10">
    <xs:annotation>
      <xs:documentation>
        This data type supports the sending of Organisation
        data as derived from
        ODS as alphanumeric string.
    </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:minLength value="3"/>
      <xs:maxLength value="12"/>
      <xs:pattern value="[A-Z0-9]*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Example Messages

Examples are attached to this specification. For ease of demonstration the maximum message size used to generate these message is 1MB rather than 5mb/50mb as used in Spine/Spine 2 respectively. These examples were generated by the reference implementation in NIS1.

Scenario

Examples use the following scenario:

Sending practice	**Receiving practice**
ODS Code: B83002	ODS Code: C81007
Party Key: RHM-801710	Party Key: RHM-803229
ASID: 715373337545	ASID: 276827251543

Patient

NHS Number: 9446 363 101

MS KATHRYN ONGARO

137 HARROWGATE LANE

STOCKTON-ON-TEES

CLEVELAND

TS19 8UT

3.7. Examples

Examples are split into two conversations indicated with highlighted background

Filename	Description
RCMR_IN030000UK06_6E242658-3D8E-11E3-A7DC-172BDA00FA67.ebxml	EHR_Extract referencing large attachments
COPC_IN000001UK01_FA039330-7E63-446A-8CA4-E9E0D00DA6E8_CONTINUE.ebxml	CONTINUE response
COPC_IN000001UK01_27C75ACD-3D93-11E3-A2CF-E1C5FFEB5098_video.ebxml	Attachment
COPC_IN000001UK01_5548EA64-3D95-11E3-B6AE-DF63BC51A0F2_FragmentIndex.ebxml	Message referencing split large attachment
COPC_IN000001UK01_3F594B5F-3FAF-11E3-B10C-8BC15657569C_Fragment0.ebxml	Split large attachment first chunk
COPC_IN000001UK01_3F35E4D9-3FAF-11E3-B10C-8BC15657569C_Fragment1.ebxml	Split large attachment second chunk
COPC_IN000001UK01_27BFE0BA-3D93-11E3-A2CF-E1C5FFEB5098_compressed.ebxml	Compressed attachment
RCMR_IN030000UK06_74799658-43DF-11E3-805D-BBA66E7A9031_skeleton.ebxml	Skeleton EHR extract

COPC_IN000001UK01_E0136D84-43E8-11E3-945E-5710B5A38718_hl7.ebxml	Compressed HL7
COPC_IN000001UK01_E01171B1-43E8-11E3-945E-5710B5A38718_mime.ebxml	MIME type not supported by Spine
COPC_IN000001UK01_EE1BA198-455D-11E3-BBE2-E36892D8683A_CompressedFragmentIndex.ebxml	Message referencing split compressed large attachment
COPC_IN000001UK01_23E1320C-455E-11E3-9C76-31DEF0104CB3_CompressedFragment0.ebxml	Split compressed large attachment first chunk
COPC_IN000001UK01_23EB6B3F-455E-11E3-9C76-31DEF0104CB3_CompressedFragment1.ebxml	Split compressed split large attachment second chunk