ITH icoserve
technology for healthcare

a siemens company

syngo.share / Release VA25B / 2018-02-26 / Revision 2690

HL7 Conformance Statement

Distributed by:

SIEMENS

Contents

Table Directory

Figure Directory

1 Introduction

The HL7 interface of *syngo*.share is designed according to the HL7 Messaging Standard, version 2.8.1 as well as the IHE IT Infrastructure Technical Framework, revision 11.0 and IHE Radiology Technical Framework, revision 13.0. The primary goal of the HL7 interface of *syngo*.share is the synchronization with third-party systems within healthcare settings. To achieve this goal *syngo*.share is able to receive and process various kinds of HL7 messages. To inform third-party system about changes, *syngo*.share also provides the possibility to generate and send HL7 messages. To establish a successful exchange of data with third-party systems, this document addresses all requirements and specialties of the HL7 interface of *syngo*.share. In the remainder of this section details of the HL7 communication as well as the HL7 message syntax are described. Afterwards, the HL7 inbound and outbound interface of *syngo*.share are discussed by specifying the layout of the supported HL7 messages, how they are processed or generated, as well as the workflows in which they occur. Since the HL7 inbound and outbound interface are strictly separated within *syngo*.share, they are covered in different sections.

1.1 HL7 Message Syntax

To be able to parse and in the following process HL7 messages, *syngo*.share requires that HL7 messages have been generated according to the guidelines of the HL7 Messaging Standard. On this account special attention has to be turned on the specification of the HL7 message delimiters (HL7 fields `MSH-1` and `MSH-2`), the character sets (HL7 field `MSH-18`), and the alternate character set handling scheme (HL7 field `MSH-20`) as well as the use of the HL7 segment terminator.

1.1.1 HL7 Message Delimiters

The HL7 message delimiters of a HL7 message are detected automatically by *syngo*.share and can be defined individually. To ensure that the detection succeeds it has to be guaranteed that the HL7 field separator is a `US-ASCII` character and the HL7 component separator, the HL7 subcomponent separator, the repetition separator, the escape character, and the truncation character are characters of the default character set of the HL7 message. However it is not allowed to use the `US-ASCII` characters carriage return (hexadecimal 0d), escape (hexadecimal 1b), and `"` (hexadecimal 22) as HL7 message delimiters because they are used as HL7 segment terminator, character set escape character (cf. alternate character set handling scheme `ISO 2022-1994`), and null character. Although the HL7 message delimiters can be defined individually, it is recommended by *syngo*.share to use the default HL7 message delimiters defined by the HL7 Messaging Standard. To guarantee that the end of a HL7 segment can be detected, *syngo*.share requires that each HL7 segment, also the last one, is terminated by the HL7 segment terminator. As HL7 segment terminator *syngo*.share expects a carriage return (hexadecimal 0d).

1.1.2 Character Sets

The HL7 interface of *syngo*.share is fully internationalized. To this end, the character sets used within a HL7 message are detected automatically by extracting the corresponding information from the HL7 field `MSH-18`. The first character set defines the default character set of a HL7 message. All other character sets define so called alternate character sets which can be used to specify characters which are not part of the default character set. If no character set is defined *syngo*.share assumes, in accordance with the HL7 Messaging Standard, that all characters used within a given HL7 message are part of US-ASCII. If alternate character sets are specified, the handling scheme of the alternate character sets (determines how character set switches are indicated) has to be defined by the HL7 field MSH-20. If the character sets or the alternate character set handling scheme of a HL7 message are incomplete or incorrect, *syngo*.share will not be able to parse the HL7 message. Table 1 lists the character sets, the MIME names of the character sets, the alternate character set handling schemes, and the escape sequences of the character sets with respect to the alternate character set handling schemes, that are supported by *syngo*.share.

Table 1: Character Sets and Alternate Character Set Handling Schemes

		Alternate Character Set Handling Schemes			
		2.3		ISO 2022-1994	
Character Set	MIME Name	GL	GR	GL	GR
ASCII	US-ASCII	C2842		2842	
8859/1	ISO-8859-1	C2842	C2D41	2842	2d41
8859/2	ISO-8859-2	C2842	C2D42	2842	2d42
8859/3	ISO-8859-3	C2842	C2D43	2842	2d43
8859/4	ISO-8859-4	C2842	C2D44	2842	2d44
8859/5	ISO-8859-5	C2842	C2D4C	2842	2d4c
8859/6	ISO-8859-6	C2842	C2D47	2842	2d47
8859/7	ISO-8859-6	C2842	C2D46	2842	2d46
8859/8	ISO-8859-8	C2842	C2D48	2842	2d48
8859/9	ISO-8859-9	C2842	C2D4D	2842	2d4d
8859/15	ISO-8859-15				
ISO IR6	US-ASCII	C2842		2842	
ISO IR14	JIS_X0201	C284A	C2949	284a	2949
ISO IR87	JIS_X0208-1990	M2442		2442	
ISO IR159	JIS_X0212-1990	M242844		242844	
KS X 1001	KS_X_1001-1997		M242943		242943
CNS 11643-1992	EUC-TW				
GB 18030-2000	GB18030				
BIG-5	Big5				
UNICODE UTF-8	UTF-8				
UNICODE UTF-16	UTF-16				
UNICODE UTF-32	UTF-32				

The character sets ISO IR87, ISO IR159, and KS X 1001 only can be used as alternate character sets because they do not build up on US-ASCII. In contrast the character sets 8859/15, CNS 11643-1992, GB 18030-2000, BIG-5, UNICODE UTF-8, UNICODE UTF-16, and UNICODE UTF-32 cannot be mixed with other character sets because for these character sets no escape sequences are provided. Note that in case of the character sets GB 18030-2000, UNICODE UTF-8, UNICODE UTF-16, and UNICODE UTF-32 that would be useless anyway because they can encode all characters of the Unicode Standard. The endianness of the character sets UNICODE UTF-16 and UNICODE UTF-32 is detected automatically by *syngo*.share. So both the little-endian and big-endian variants of UTF-16 and UTF-32 are supported. To allow switches between multiple character sets, *syngo*.share applies the rules defined by the HL7 Messaging Standard as well as ISO/IEC 2022 and ECMA-35. Following these standards, character sets are divided into a GL (lower bytes between 20 and 7f) and GR (higher bytes between a0 and ff) level. To ensure that the characters of the GL or GR level of a character set can be used, the desired character set switch has to be triggered by specifying the corresponding escape sequence. Note that the escape sequences of the alternate character set handling scheme `ISO 2022-1994`, mentioned in Table 1, are stated in hexadecimal. Since HL7 message delimiters have to be specified in the default character set, the scope of a character set switch is limited to the HL7 subcomponent in which it occurs. So, at the end of each HL7 subcomponent the character sets defined by the GL and GR level must be switched back to the default character set, if necessary.

1.2 HL7 Communication

To communicate with third-party systems *syngo*.share uses TCP/IP connections based on the Minimal Lower Layer Protocol specified by the HL7 Messaging Standard. Within HL7 communications *syngo*.share can act either as receiver or sender of HL7 messages. If *syngo*.share receives HL7 messages, it is assumed that the connecting third-party system administrates the HL7 connection. This means, the HL7 connection is not closed by *syngo*.share as long as no irreparable error occurs. Vice versa, if *syngo*.share sends HL7 messages to a third-party system it assumes that it controls the HL7 connection. To be robust against network problems, the HL7 outbound interface of *syngo*.share opens a HL7 connection if and only if HL7 messages have to be transmitted to a third-party system and closes an established HL7 connection as soon as all HL7 messages have been sent. To avoid that HL7 connections are established and terminated frequently, *syngo*.share waits a configured amount of time before it reestablishes a HL7 connection to exchange further HL7 messages.

To receive and send HL7 messages *syngo*.share follows either the original or enhanced processing rules specified by the HL7 Messaging Standard. A received HL7 message is processed according to the original processing rules if the accept acknowledgment type (see HL7 field `MSH-15`) and application acknowledgment type (see HL7 field `MSH-16`) are null or not present. If the accept acknowledgment type or application acknowledgment type are specified the enhanced processing rules are used to exchange the sent HL7 message. If *syngo*.share acts as a HL7 message sender it has to be configured whether the original or enhanced processing rules should be used to transmit HL7 messages. In the latter case *syngo*.share requires the transmission of HL7 accept acknowledgment messages but denies the exchange of HL7 application acknowledgment messages. To complement the two processing modes *syngo*.share also supports the Sequence Number Protocol defined by the HL7 Messaging Standard.

Each time *syngo*.share receives a HL7 message it is checked for syntactic and semantic correctness. If the HL7 message is syntactically incorrect it cannot be parsed by *syngo*.share and hence neither be stored nor processed. Since the sending third-party system cannot be informed about syntactic errors even if the transmission of an HL7 accept acknowledgment message has been requested, the HL7 connection is closed by *syngo*.share. To validate the HL7 message semantically, it is checked by *syngo*.share if the type of the HL7 message can be processed and the receiving facility specified by the HL7 field `MSH-6` is related to one of the configured tenants. If the type of the received HL7 message is not supported by *syngo*.share the received HL7 message is rejected. To this end a HL7 accept acknowledgment message with the acknowledgment code `AR` or `CR` (the former code is specified if the original processing rules are applied, the latter one if the enhanced processing rules are used) is sent to the sending third-party system if the receiving of the HL7 message has to be acknowledged. If the type of the received HL7 message can be processed by *syngo*.share but the receiving facility does not address any of the configured tenants, the HL7 message is stored on the file system within a configured error directory. To inform the sending third-party system about the occurred error, a HL7 accept acknowledgment message with the acknowledgment code `AE` or `CE` (the former code is specified if the original processing rules are applied, the latter one if the enhanced processing rules are used) is sent back, if demanded. In case that the syntactic and semantic checks succeed, the received HL7 message is stored by *syngo*.share and scheduled for processing. If the receiving of the HL7 message has to be acknowledged, this time a HL7 accept acknowledgment message specifying the acknowledgment code AA or CA (the former code is specified if the original processing rules are applied, the latter one if the enhanced processing rules are used) is sent. If a HL7 message is transferred according to the enhanced processing rules it might be necessary to exchange further information with the sending third-party system as soon as the HL7 message has been processed. To this end *syngo*.share is able to send so called HL7 application acknowledgment messages (see Section 3.4 for further information).

If *syngo*.share sends a HL7 message to a third-party system it always expects that the received HL7 message is acknowledged. In case of a positive acknowledgment *syngo*.share demands that the HL7 accept acknowledgment message specifies the acknowledgment code AA if the original processing rules are used to transmit the HL7 message and the acknowledgment code CA if the enhanced processing rules are used to exchange the HL7 message. Note that it can be individually configured in *syngo*.share which processing rules should be used to transfer HL7 messages.

The chronological order in which received HL7 messages are stored by *syngo*.share defines the chronological order in which the HL7 messages will be processed. So to guarantee that HL7 messages are processed correctly, they should always be sent in the correct chronological order. Vice versa, if *syngo*.share sends HL7 messages to a third-party system, it always guarantees that HL7 messages which depend on each other are transmitted in chronological order.

1.3 Preliminaries

In the remainder of the document it is assumed that for incoming HL7 messages R (C, O) is used to indicate that a HL7 segment, HL7 field, HL7 component, or HL7 subcomponent is required (conditionally required, optional). In case of outgoing HL7 messages R defines that a HL7 segment, HL7 field, HL7 component, or HL7 subcomponent is always specified by *syngo*.share whereas C and O denote that a HL7 segment, HL7 field, HL7 component, or HL7 subcomponent is just specified under certain circumstances or if *syngo*.share possesses the corresponding information. HL7 segments, HL7 fields, HL7 components, and HL7 subcomponents which are not mentioned, are ignored and hence not evaluated by *syngo*.share. For some of the listed HL7 segments, HL7 fields, HL7 components, or HL7 subcomponents length information are declared. This indicates, that the corresponding HL7 segment, HL7 field, HL7 component, or HL7 subcomponent is stored in the database which limits the number of characters that can be used to define it. If the length of a HL7 segment, HL7 field, HL7 component, or HL7 subcomponent exceeds the defined limit, the provided information cannot be stored and hence the HL7 message cannot be processed successfully. If the length of a HL7 segment, HL7 field, HL7 component, or HL7 subcomponent is not stated, the information is not stored by *syngo*.share although it might be needed to process the HL7 message correctly.

As required by the HL7 Messaging Standard it is assumed that all date and time information is formatted according to the following standardized format:

```
YYYY[MM[DD[HH[MM[SS[.S[S[S[S]]]]]]]]][+/-ZZZZ].
```

Timestamps which do not possess timezone information are assumed to be in local time. To avoid problems with timestamp conversions, all timestamps specified in HL7 messages sent by *syngo*.share are given in local time and hence do not specify any timezone offsets.

In general the HL7 Messaging Standard strictly differs between the insertion and update of information. Since in general it is inconvenient or event not possible for systems to decide if information has to be inserted or updated, *syngo*.share treats HL7 messages which insert or update information depending on the existence of the underlying objects (for example patients, visits, orders, etc.). That means, if a HL7 message is received by *syngo*.share which should insert an object that already exists, the information of the object is updated. Vice versa if a HL7 message is received by *syngo*.share which should update the information of an unknown object, the corresponding object is inserted.

2 HL7 Inbound Interface

2.1 Patient and Visit Administration

2.1.1 Operating Mode

The primary goal of the HL7 interface is the synchronization of patient and visit information. For this purpose *syngo*.share provides an interface which allows the third-party system that manages the patient and visit base (e.g., HIS or RIS) to insert, update, delete, and merge patients as well as to insert, update, delete, move, and merge visits by sending HL7 ADT messages. Each HL7 ADT message has to be dedicated to exactly one patient and mostly one visit. Since *syngo*.share does not support visits on account level it is assumed that visit information always specify a visit on visit level (cf. HL7 Messaging Standard).

Within *syngo*.share patients and visits are not just inserted via HL7 ADT messages. If a DICOM Image or generic file is archived and the corresponding patient or visit does not exist, a new patient or visit is inserted. Similarly, if an order is inserted via a HL7 OMI or HL7 ORM message but the addressed patient or visit is unknown, it is inserted (see Section 2.2). Since the patient and visit information associated with DICOM Images or generic files usually do not come directly from the third-party system which manages the patient and visit base, such information is of lower quality as if it would have been provided by HL7 ADT or HL7 ORM messages. Now to preserve the origin of a patient and visit as well as to prevent that high-quality information can be overwritten by low-quality information, patient and visit information are flagged as high-quality or low-quality information. Any patient or visit that has been inserted during the processing of a HL7 ADT, HL7 OMI, or HL7 ORM message is marked to be of high-quality. Patients and visits which have been inserted during import processes are flagged as low-quality patients and visits.

High-quality patients are identified by a patient ID whereas each low-quality patient is identified by a so called identification tuple consisting of a patient ID as well as the last name, first name, birth timestamp, and sex of the patient. The reason for the quite complex identification of low-quality patients is related to the fact that patient IDs of low-quality patients need not be correct. So to avoid that low-quality patients accidentally overwrite each other, the mentioned identification tuple is used to identify low-quality patients uniquely. In case of visits the situation is much simpler. Since visits can be identified just by a visit ID, both high-quality visits and low-quality visits are identified equally.

To insert, update, delete, and merge patients *syngo*.share sticks to the rules proposed by the HL7 Messaging Standard. That means, if a patient should be inserted or updated it is first checked if a high-quality patient with the specified patient ID or a low-quality patient with the defined identification tuple exists. If no patient is found, a high-quality patient with the information specified by the HL7 ADT message is inserted. If a high-quality patient is found it is first checked if the last name, first name, birth timestamp, or sex of the patient will be changed. If that is the case the found patient is invalidated by a newly inserted patient which represents an updated version of the found patient. If a low-quality patient is found it is updated and transformed into a high-quality patient. An invalidation tree is not constructed. If a patient has to be deleted it is first checked if the patient exists (remember that high-quality patients are identified by the specified patient IDs whereas low-quality patients are identified by the given identification tuples). If it exists it is deleted. Otherwise the HL7 ADT message is ignored. Finally, if two patients have to be merged it is first checked if the new patient exists. If that is the case the found patient is updated and transformed into a high-quality patient if needed. Otherwise an appropriate high-quality patient is inserted. Afterwards it is checked if the old patient exists. If that is the case the old patient is invalidated by the new patient to maintain a possibly existing invalidation tree.

To insert, update, delete, move, and merge visits *syngo*.share proceeds as follows: If a visit should be inserted or updated it is first checked if a high-quality or low-quality visit with the specified visit ID exists. If no visit is found, a high-quality visit with the information specified by the HL7 ADT message is inserted. Otherwise the found visit is updated and transformed into a high-quality visit. If a visit has to be deleted it is first checked if it exists (remember that high-quality and low-quality visits are identified by the stated visit IDs). If it exists it is deleted. Otherwise the HL7 ADT message is ignored. If a visit has to be moved from one patient to another patient it is first checked if the visit exists. If that is not the case, a high-quality visit belonging to the requested patient is inserted. Otherwise the found visit is updated, transformed into a high-quality visit if needed, and moved to the requested patient. Finally, if two visits have to be merged it is first checked if the new visit exists. If that is the case the found visit is updated and transformed into a high-quality visit if needed. Otherwise an appropriate high-quality visit is inserted. Afterwards it is checked if the old visit exists. If that is the case the old visit is invalidated by the new visit to maintain a possibly existing invalidation tree.

Note that the reason for the construction of invalidation trees can be traced back to the fact that patient and visit information of DICOM Images and generic files might have become obsolete when they are archived by *syngo*.share. So if a DICOM Image or generic file consists of old patient (visit) information, *syngo*.share tries to find a patient (visit) by searching for an invalidated patient (visit) with the given identification tuple (visit ID). If that succeeds the DICOM Image or generic file is assigned to the correct patient (visit) without further ado.

2.1.2 HL7 ADT Message Structure

To administrate patients and visits via HL7 ADT messages *syngo*.share supports the trigger events `A01` to `A04`, `A06` to `A08`, `A11` to `A13`, `A18`, `A23`, `A28`, `A29`, `A31`, `A34`, `A40`, `A42`, `A45`, `A47`, and `A50`. Other trigger events associated with HL7 ADT messages are currently not supported by *syngo*.share. The trigger events `A01`, `A02`, `A03`, `A04`, `A12`, and `A13` are used to insert or update a patient and visit. So a HL7 ADT message associated with one of these trigger events has to comply with the following structure:

`MSH PID [NK1] PV1 [ROL] [GT1]`

The trigger events `A06` and `A07` are used to insert or update a patient and visit. If a HL7 MRG segment is defined, the specified visit is merged with the visit specified by the HL7 PV1 segment. The structure of such a HL7 ADT message looks as follows:

`MSH PID [MRG] [NK1] PV1 [ROL] [GT1]`

With the trigger events `A08`, `A28`, and `A31` it is possible to insert or update a patient without specifying any visit. However, if a visit is present it is inserted or updated. With trigger event `A31` it is also possible to set the XAD-PID of referenced patients, if the message contains one patient ID marked with type `GPI` as XAD-PID. The HL7 ADT message associated with one of these trigger events has to comply with the following structure:

`MSH PID [NK1] [PV1] [ROL] [GT1]`

A HL7 ADT message equipped with the trigger event `A11` or `A23` deletes a visit. To this end it has to provide the following information:

MSH PID PV1

Next, the trigger events A18, A34, and A40 should be used in combination with a HL7 ADT message which merges two patients. Hence it has to specify the following HL7 segments:

MSH PID MRG [NK1] [GT1]

A HL7 ADT message that specifies the trigger event A29 deletes a patient. The structure of such a HL7 ADT message looks as follows:

MSH PID

The trigger event A42 is used to merge a visit. If the patient associated with the new visit does not exist it is created. Otherwise it is updated. Each HL7 ADT message associated with this trigger event has to meet the following structure:

MSH PID MRG [NK1] PV1 [ROL] [GT1]

The trigger event A45 is used to move a visit specified by the HL7 MRG segment. If the patient specified by the HL7 ADT message does not exist it is created. Otherwise it is updated. Furthermore if the HL7 ADT message specifies a visit via a HL7 PV1 segment, the moved visit is updated if the two visit IDs are identical. Otherwise the moved visit is merged with the new visit specified by the HL7 PV1 segment. Note that the new visit is created if it does not exist. The structure of such a HL7 ADT message looks as follows:

MSH PID MRG [NK1] [PV1] [ROL] [GT1]

A HL7 ADT message based on the trigger event A47 updates the patient ID of a patient. To perform this operation *syngo*.share invalidates the old patient by a newly created patient which represents an updated version of the old patient. HL7 ADT messages of this type have to provide the following information:

MSH PID MRG [NK1] [GT1]

Finally, the trigger event A50 is used to replace the visit ID of a visit. To guarantee a consistent construction of invalidation trees *syngo*.share invalidates the old visit by a newly created visit which represents an updated version of the old visit. Each HL7 ADT message associated with this trigger event has to comply with the following structure:

MSH PID MRG [NK1] PV1 [ROL] [GT1]

Independent from the used trigger event, the patient ID of the given patient has to be specified by the HL7 component PID-3.1 whereas the visit ID has to be defined by the HL7 component PV1-19.1. The specification of a HL7 ROL, HL7 NK1, and HL7 GT1 segment is possible whenever the used trigger event might cause the insertion or update of the stated patient or visit although the HL7 ROL, HL7 NK1, and HL7 GT1 segment might not be part of the official structures defined by the HL7 Messaging Standard.

2.2 Order Administration

2.2.1 Operating Mode

The HL7 inbound interface of *syngo*.share allows third-party systems (e.g., HIS or RIS) to insert, update, and delete orders, requested procedures, and scheduled procedure steps by sending HL7 OMI and HL7 ORM messages. Each HL7 OMI message must be dedicated to exactly one order whereas a HL7 ORM messages has to focus on a particular requested procedure of an order. Since HL7 OMI messages represent the state of the art and facilitate a more compact and simpler specification of orders, requested procedures, and scheduled procedure steps than HL7 ORM messages, they should be preferred. The processing of multiple orders within a HL7 OMI or HL7 ORM message is not supported.

To identify an order uniquely *syngo*.share uses either the placer order number or filler order number. Since the filler order number is usually more specific than the placer order number, the filler order number is used to identify an order whenever it is specified. If the filler order number is not given, the placer order number is used. In the remainder of the section we denote the ID used to identify an order as order ID. Each order consists of one or more requested procedures. To associate requested procedures with DICOM Studies *syngo*.share follows the guidelines proposed by the IHE Radiology Technical Framework. That means, to identify a requested procedure uniquely *syngo*.share uses the Study Instance UID of the resulting DICOM Study specified by the received HL7 OMI (see HL7 IPC segment) or HL7 ORM message (see HL7 ZDS segment). Finally each requested procedure is again subdivided into one or more scheduled procedure steps. To identify a scheduled procedure step uniquely *syngo*.share uses the so called scheduled procedure step ID.

To insert or update orders, requested procedures, and scheduled procedure steps *syngo*.share checks at first if an order with the given order ID exists. If that is the case the information of the order are updated. Otherwise the order is created using the information provided by the HL7 OMI or ORM message. If the patient or visit associated with the order does not exist, a high-quality patient or visit is inserted. If the patient or visit exists but is of low-quality, it is updated and transformed into a high-quality patient or visit (see Section 2.1 for detailed information about the handling of patients and visits). Afterwards it is checked for each specified requested procedure if the created or found order possesses a requested procedure with the specified Study Instance UID. If that is the case the corresponding requested procedure is updated. Otherwise it is created. Likewise for each scheduled procedure step it is checked if the dedicated requested procedure consists of a scheduled procedure step with the stated scheduled procedure step ID. If that is the case the corresponding scheduled procedure step is updated. Otherwise it is created.

If *syngo*.share receives a HL7 OMI or HL7 ORM message which requires the deletion of an order it is checked if the order exists using the specified order ID (remember that the order ID either refers to the filler order number or placer order number). If the order exists it is deleted inclusive all associated requested procedures and scheduled procedure steps. Otherwise the HL7 OMI or HL7 ORM message is ignored. Note that it is not possible to delete single requested procedures or scheduled procedure steps with HL7 OMI or HL7 ORM messages.

To synchronize requested procedures and DICOM Studies *syngo*.share provides the possibility to set up a so called metadata update (see also the System Documentation of *syngo*.share). In this particular case *syngo*.share checks, whenever a requested procedure is inserted or updated by a HL7 OMI or HL7 ORM message, if the addressed DICOM Study already exists. If that is the case, at first the metadata of the existing DICOM Study is synchronized by modifying the values of the following DICOM Tags:

- `AccessionNumber` (0008,0050): updated via the HL7 component `IPC-1.1` in case of a HL7 OMI message or the HL7 field `OBR-18` in case of a HL7 ORM message
- `IssuerOfAccessionNumberSequence` (0008,0051): updated via the HL7 components `IPC-1.2, IPC-1.3, IPC-1.4` in case of a HL7 OMI message or the HL7 components `OBR-3.2, OBR-3.3, OBR-3.4` in case of a HL7 ORM message
- `ReferringPhysiciansName` (0008,0090): updated via the HL7 field `PV1-8`
- `StudyDescription` (0008,1030): updated via the HL7 component `OBR-44.5`
- `RequestedProcedureID` (0040,1001): updated via the HL7 component `IPC-2.1` in case of a HL7 OMI message or the HL7 field `OBR-19` in case of a HL7 ORM message

Afterwards the existing DICOM Study is reallocated to the patient and visit associated with the order, if necessary. This guarantees that corrective workflows based on orders, as defined by the IHE Radiology Technical Framework, succeed independent from the quality of the involved patients and visits.

Another feature of *syngo*.share which is connected to the administration of orders is the providing of DICOM worklists. For this purpose *syngo*.share holds a DICOM worklist for each scheduled station AET that has been addressed by at least one HL7 OMI or HL7 ORM message. If a HL7 OMI or HL7 ORM message specifies an order that has been scheduled for processing its scheduled procedure steps are put on the worklists of the assigned scheduled station AETs. A scheduled procedure step is removed from a DICOM worklist if the corresponding DICOM Study has been archived or the associated order has been marked as completed by updating the status of the order via a HL7 OMI or HL7 ORM message. In the former case the Study Instance UID and scheduled procedure step IDs of the DICOM Study are used to identify affected scheduled procedure steps. In the latter case all scheduled procedure steps of the order are removed from the worklist. Further information about worklists can be obtained from the System Documentation of *syngo*.share as well as the DICOM Conformance Statement.

2.2.2 HL7 OMI Message Structure

To insert, update, and delete orders, requested procedures, and scheduled procedure steps via HL7 OMI messages *syngo*.share supports the trigger event `O23` as proposed by the HL7 Messaging Standard. Other trigger events associated with HL7 OMI messages are neither supported by the HL7 Messaging Standard nor by *syngo*.share. To ensure that *syngo*.share can process HL7 OMI messages they have to meet the following structure:

`MSH PID [NK1] [PV1] [ROL] [GT1] {ORC TQ1 OBR {IPC}}`

To distinguish between HL7 OMI messages which insert, update, and delete orders, requested procedures, and scheduled procedure steps *syngo*.share relies on the valid specification of the HL7 field `ORC-1`. To insert an order, requested procedure, or scheduled procedure step this HL7 field has to specify the value `NW`. The values `XO` or `DC` have to be used if an order, requested procedure, or scheduled procedure step should be updated (note that `XO` indicates an ordinary modification of an order, requested procedure, or scheduled procedure step whereas `DC` announces the discontinuation of an order, requested procedure, and scheduled procedure step). If an order has to be deleted the value `CA` has to be defined by the HL7 field `ORC-1`. If *syngo*.share is used to provide DICOM worklists the HL7 field `ORC-5` has to define the value `SC` in order to guarantee that the listed scheduled procedure steps are scheduled for processing. To mark an order as completed and hence to remove all scheduled procedure steps from worklists, `ORC-5` has to state the value `CM`. The placer order number and filler order number which are used by *syngo*.share to identify an order uniquely have to be specified by the HL7 components `OBR-2.1` and `OBR-3.1` as well as `ORC-2.1` and `ORC-3.1`. The Study Instance UID which is used to identify a requested procedure uniquely has to be specified by the HL7 component `IPC-3.1`. Likewise the HL7 component `IPC-4.1` has to define for each scheduled procedure step a unique scheduled procedure step ID. The scheduled station AET associated with each scheduled procedure step must be defined by the HL7 field `IPC-9`. Note that each HL7 OMI message is dedicated to exactly one order. To define more than one requested procedure, the HL7 ORC/ TQ1/OBR/IPC segment group has to be repeated appropriately often. Furthermore to divide a requested procedure into multiple scheduled procedure steps the HL7 IPC segment has to be repeated suitable often within the corresponding HL7 ORC/TQ1/OBR/IPC segment group. If more than one requested procedure (scheduled procedure step) is defined, it has to be guaranteed that information related to the order (order and requested procedure) do not vary. Beside that the HL7 PV1 segment should be present if and only if the underlying order belongs to a visit. Furthermore, the specification of a HL7 ROL and HL7 NK1 segment is possible although it is not part of the official structure defined by the HL7 Messaging Standard.

2.2.3 HL7 ORM Message Structure

To insert, update, and delete orders, requested procedures, and scheduled procedure steps via HL7 ORM messages *syngo*.share supports the trigger event `O01` as proposed by the HL7 Messaging Standard. Other trigger events associated with HL7 ORM messages are neither supported by the HL7 Messaging Standard nor by *syngo*.share. To ensure that *syngo*.share can process HL7 ORM messages they have to meet the following structure:

MSH PID [NK1] [PV1] [ROL] [GT1] {ORC OBR} ZDS

To distinguish between HL7 ORM messages which insert, update, and delete orders, requested procedures, and scheduled procedure steps *syngo*.share relies on the valid specification of the HL7 field ORC-1. To insert an order, requested procedure, or scheduled procedure step this HL7 field has to specify the value NW. The values XO or DC have to be used if an order, requested procedure, or scheduled procedure step should be updated (note that XO indicates an ordinary modification of an order, requested procedure, or scheduled procedure step whereas DC announces the discontinuation of an order, requested procedure, and scheduled procedure step). If an order has to be deleted the value CA has to be defined by the HL7 field ORC-1. If *syngo*.share is used to provide DICOM worklists the HL7 field ORC-5 has to define the value SC in order to guarantee that the listed scheduled procedure steps are scheduled for processing. To mark an order as completed and hence to remove all scheduled procedure steps from worklists, ORC-5 has to state the value CM. The placer order number and filler order number which are used by *syngo*.share to identify an order uniquely have to be specified by the HL7 components OBR-2.1 and OBR-3.1 as well as ORC-2.1 and ORC-3.1. The Study Instance UID which is used to identify a requested procedure uniquely has to be specified by the HL7 component ZDS-1.1. Likewise the HL7 field OBR-20 has to define for each scheduled procedure step a unique scheduled procedure step ID. The scheduled station AET associated with each scheduled procedure step must be defined by the HL7 field OBR-21. Note that each HL7 ORM message is dedicated to exactly one requested procedure. To define more than one scheduled procedure step, the HL7 ORC/OBR segment group has to be repeated appropriately often. However, in this particular case it is important that information related to the order and requested procedure do not vary. Beside that the HL7 PV1 segment should be present if and only if the underlying order belongs to a visit. Furthermore, the specification of a HL7 ROL and HL7 NK1 segment is possible although it is not part of the official structure defined by the HL7 Messaging Standard.

2.3 Archiving of Generic Files

2.3.1 Operating Mode

To archive generic files various interfaces are provided by *syngo*.share (cf. System Documentation of *syngo*.share). The most frequently used interface is based on the HL7 Messaging Standard. To archive generic files via the HL7 interface both HL7 MDM and HL7 ORU messages can be used. However, it is strictly recommended to use HL7 MDM messages instead of HL7 ORU messages because the use of HL7 MDM messages is favored by the HL7 Messaging Standard since they allow a more fine-grained specification of generic files. In addition some functionalities like the XDS registration of a generic file in an affinity domain rely on metadata which can only be specified by a HL7 MDM message.

In general all possible types of generic files can be archived. To allow third-party systems to synchronize their data with *syngo*.share it is possible to insert, update, and delete generic files. Each HL7 MDM and HL7 ORU message must be dedicated to exactly one generic file. The binary data of a generic file can be transmitted by reference or by value. In the former case the generic file has to be written to a share and referenced via an UNC-path which meets the following format:

- Windows: \\<host>\<share>\<path>
- Linux: //<host>/<share>/<path>

In the latter case the binary data has to be encapsulated in the HL7 MDM or HL7 ORU message. Since it is not allowed to transmit non-printable characters within a HL7 message, it is necessary to encode the binary data so that the sent HL7 MDM or HL7 ORU message is syntactically correct and hence can be parsed by *syngo*.share. Actually, *syngo*.share supports the following encodings:

- `A`: The binary data is encoded as a HL7 escaped string.
- `Base64`: The binary data is Base64 encoded. It is recommended to ignore the maximum line length of 76 characters required by RFC 2045. If not, line breaks have to be HL7 escaped.
- `Hex`: The binary data has to be encoded as a hexadecimal string.
- `UU`: The binary data has to be uu encoded. Note that non-printable characters have to be HL7 escaped.

To identify a generic file uniquely *syngo*.share uses two IDs: a generic file ID and an application ID. The application ID is needed to identify the third-party system that archives the generic file whereas the generic file ID is used to identify the generic file among the generic files inserted by the sending third-party system. Each generic file is archived in *syngo*.share within a so called generic container. Generic containers are used to group generic files. They are identified by a generic container ID and a coding system ID. Since a generic container might be reused for patients and shared between third-party systems the combination of generic container ID and coding system ID need not be unique. However, to ensure that generic containers can be used in a meaningful way it is is guaranteed by *syngo*.share that a patient does not possess two generic containers with identical generic container ID and coding system ID with regard to a certain visit and organizational unit. So the integration of a generic file into a generic container takes places with respect to the associated patient, visit, and organizational unit.

The approach used by *syngo*.share to insert or update a generic file depends on the type of the HL7 message. If a generic file should be updated via a HL7 MDM message, according to the HL7 Messaging Standard, a new generic file ID should be generated which is linked to the original generic file by specifying the old generic file ID. Since the generation of a new generic file ID might not be possible or inconvenient, *syngo*.share supports the update of generic files without changing the generic file ID. Now to insert or update a generic file via a HL7 MDM message *syngo*.share primarily checks if a generic file with the specified new generic file ID and application ID or old generic file ID and application ID already exists. If neither the new generic file ID nor the old generic file ID (might be undefined) yields a generic file archived by *syngo*.share the generic file is stored based on the patient, visit, and organizational unit information specified by the HL7 MDM message. If the patient or visit does not exist, a low-quality patient or visit is inserted. An update of an existing patient or visit is not performed (see Section 2.1 for detailed information about the handling of patients and visits). If the old generic file ID and application ID references a generic file stored by *syngo*.share but the new generic file ID and application ID does not, the metadata of the generic file identified by the old generic file ID and application ID are updated and the old generic file ID is replaced by the new generic file ID. If the binary data of the generic file changed too, a new version of the generic file is stored. If the new generic file ID and application ID addresses a generic file archived by *syngo*.share but the old generic file (might be undefined) and application ID does not, the generic file identified by the new generic file ID and application ID is updated (metadata and binary data if present). If both, the new generic file ID and application ID as well as the old generic file ID and application ID refer to a generic file archived by *syngo*.share an error is reported if the new generic file ID and the old generic file ID are different. If the generic file IDs are identical, the metadata of the found generic file is updated without changing the generic file ID. Furthermore if the binary data of the generic file changed too, a new version of the generic file is archived.

The processing of HL7 ORU messages which insert or update generic files is much easier than the one of HL7 MDM messages because generic file IDs cannot be changed via HL7 ORU messages. So to insert or update a generic file via a HL7 ORU message *syngo*.share primarily checks if a generic file with the specified generic file ID and application ID already exists. If that is not the case, the generic file is archived based on the patient, visit, and organizational unit information specified by the HL7 ORU message. If the patient or visit does not exist, a low-quality patient or visit is inserted. An update of an existing patient or visit is not performed (see Section 2.1 for detailed information about the handling of patients and visits).

If the generic file exists, the metadata of the generic file are updated. If the binary data of the generic file changed too, a new version of the generic file is stored. Note, if a generic file is updated via a HL7 MDM or HL7 ORU message any specified patient, visit, and organizational unit information are ignored.

The approach used by *syngo*.share to delete generic files is independent form the type of the HL7 message. To delete a generic file *syngo*.share simply checks if a generic file with the specified generic file ID and application ID exists. If that is the case the corresponding generic file is deleted. Afterwards it is checked if the generic container, to which the generic file belongs, just consists of deleted generic files. If that is the case the generic container is deleted too. If the generic file, which should be deleted, does not exist, *syngo*.share ignores the sent HL7 MDM or HL7 ORU message.

2.3.2 HL7 MDM Message Structure

To insert, update, and delete generic files via HL7 MDM messages *syngo*.share supports the trigger events `T02`, `T09`, `T10`, and `T11`. Other trigger events associated with HL7 MDM messages are currently not supported by *syngo*.share. The trigger event `T02` is used to insert a generic file, `T09` is used to update the metadata of a generic file, `T10` is used to update the metadata and binary data of a generic file, and `T11` is used to delete a generic file. To ensure that *syngo*.share can process HL7 MDM messages associated with trigger events `T02` and `T10` these HL7 messages have to comply with the following structure:

`MSH PID [NK1] [PV1] [ROL] [GT1] [OBR] TXA OBX`

The structure of HL7 MDM messages associated with trigger events `T09` and `T11` is a little bit simpler because they do not provide any binary data. On this account it is not necessary to specify any HL7 OBX segment. So the structure of such HL7 MDM messages looks as follows:

`MSH PID [NK1] [PV1] [ROL] [GT1] [OBR] TXA [OBX]`

Independent from the used trigger event the generic file ID has to be specified by the HL7 component `TXA-12.1`. The application ID has to be defined by the HL7 field `MSH-3` whereby the HL7 component `MSH-3.1` is evaluated if it is specified and the HL7 component `MSH-3.2` is used if `MSH-3.1` has been left empty. The generic container ID and the coding system ID have to be defined by the HL7 components `TXA-24.1` and `TXA-24.3`. Note that in case of HL7 ORU messages the generic file ID, generic container ID, and coding system ID are specified by different HL7 components. It also has to be taken into account that the HL7 PV1 segment should be present if and only if the underlying generic file belongs to a visit. The specification of a HL7 ROL, HL7 NK1, and HL7 GT1 segment is possible although they are not part of the official structure defined by the HL7 Messaging Standard. The same applies to the HL7 OBX segment in case of HL7 MDM messages associated with the trigger event `T09` — although it is not part of the official structure defined by the HL7 Messaging Standard it can be specified in order to update metadata. Besides that it must be pointed out that *syngo*.share supports the specification of at most one HL7 OBR segment. The specification of a HL7 ORC segment, as requested by the HL7 Messaging Standard, is not necessary. Furthermore, in accordance with the HL7 Messaging Standard it is not allowed to transmit the binary data of a generic file in a HL7 MDM message based on the trigger event `T02` or `T10` using more than one HL7 OBX segment.

2.3.3 HL7 ORU Message Structure

To insert, update, and delete generic files via HL7 ORU messages *syngo*.share supports the trigger event `R01` as proposed by the HL7 Messaging Standard. Other trigger events associated with HL7 ORU messages are not supported by *syngo*.share. To ensure that *syngo*.share can process HL7 ORU messages within this context they have to meet the following structure:

`MSH PID [NK1] [PV1] [ROL] [GT1] [OBR] OBX`

To distinguish between HL7 ORU messages which insert or update generic files and HL7 ORU messages which delete generic files *syngo*.share relies on the valid specification of the HL7 fields `OBR-25` and `OBX-11`. In order to insert or update a generic file the two HL7 fields have to be specify the value `F`. If a generic file should be deleted, a `D` has to be stated. Because HL7 ORU messages do not possess HL7 TXA segments the IDs used to identify generic files and generic containers are defined by other HL7 components than in case of HL7 MDM messages. Especially, the generic file ID has to be defined by the HL7 component `OBX-3.1` and the generic container ID and the coding system ID have to be defined by the HL7 components `OBX-3.4` and `OBX-3.6`. The application ID has to be defined by the HL7 field `MSH-3`, similarly as in case of HL7 MDM messages. Note that the HL7 PV1 segment should be present if and only if the underlying generic file belongs to a visit. The specification of a HL7 ROL and HL7 GT1 segment is possible although it is not part of the official structure defined by the HL7 Messaging Standard. Besides that it must be pointed out that *syngo*.share supports the specification of at most one HL7 OBR segment. The specification of a HL7 ORC segment is not necessary. Furthermore, in accordance with the HL7 Messaging Standard it is not allowed to transmit the binary data of a generic file using more than one HL7 OBX segment.

2.4 Sharing of DICOM Studies and Generic Containers

2.4.1 Operating Mode

Via the HL7 inbound interface it is possible to share DICOM Studies and generic containers so that they can be accessed via the module *webbox* by designated users. To share DICOM Studies and generic containers HL7 ORU messages have to be used. To allow third-party systems to administrate shares of DICOM Studies and generic containers it is possible to create and delete shares. Each HL7 ORU message has to be dedicated to exactly one DICOM Study or generic container. The number of users for which a share should be created or deleted is not restricted. Since each share allows exactly one user to access a specific DICOM Study or generic container, each HL7 ORM message has to specify for all listed users if a share with respect to the referenced DICOM Study or generic container should be created or an existing share for the given DICOM Study or generic container should be deleted. So, with one HL7 ORU message it is possible to create and delete shares. To ensure that *syngo*.share can identify the DICOM Study or generic container which should be shared or for which a share should be deleted uniquely, the reference pointer of the DICOM Study or generic container has to be specified. Note that reference pointers can be obtained from *syngo*.share automatically via HL7 ORU messages (see Section 3.2 for further information). To identify a user *syngo*.share expects that the login name of the user is stated.

2.4.2 HL7 ORU Message Structure

To administrate DICOM Study and generic container shares via HL7 ORU messages *syngo*.share supports the trigger event `R01`. So based on this fact the transmitted HL7 ORU messages have to meet the following structure:

```
MSH OBX {PRT}
```

To distinguish between shares that have to be created for a DICOM Study or generic container and shares which should be deleted, *syngo*.share relies on the valid specification of the HL7 field `PRT-2`. In order to share a DICOM Study or generic container `PRT-2` has to specify the value `LI`. If an existing share should be deleted, the value `UN` has to be stated. The reference pointer which identifies the DICOM Study or generic container that should be shared has to be specified by the HL7 component `OBX-5.1`. The login name of the user has to be defined by the HL7 component `PRT-1.1`. Note, to specify more than one user the HL7 PRT segment has to be repeated appropriately often.

2.5 Deletion of DICOM Studies and Generic Containers

2.5.1 Operating Mode

Via the HL7 inbound interface it is possible to delete DICOM Studies and generic containers. Since this way of deletion is not standardized it is recommended to use KOS Rejection Notes to delete DICOM Images (see the *syngo*.share of DICOM Conformance Statement) and HL7 MDM or HL7 ORU messages, as specified by Section 2.3, to delete generic files. If the standardized deletion is out of the question, HL7 ORU messages can be used to delete DICOM Studies and generic containers. Thereby each HL7 ORU message must be dedicated to exactly one DICOM Study or generic container. To ensure that *syngo*.share can identify the DICOM Study or generic container which should be deleted uniquely, the reference pointer of the DICOM Study or generic container has to be specified. Note that reference pointers can be obtained from *syngo*.share automatically via HL7 ORU messages (see Section 3.2 for further information).

2.5.2 HL7 ORU Message Structure

To delete DICOM Studies and generic containers via HL7 ORU messages *syngo*.share supports the trigger event `R01`. Based on this fact the transmitted HL7 ORU messages have to meet the following structure:

```
MSH OBX
```

The reference pointer which identifies the DICOM Study or generic container that should be deleted has to be specified by the HL7 component `OBX-5.1`.

2.6 Administration of HL7 ZPA Messages

2.6.1 Operating Mode

The HL7 inbound interface in *syngo*.share allows third-party systems to change the Accession Number and Study Instance UID of DICOM studies and to move DICOM studies from one patient to another via ZPA messages. The corresponding order gets deleted. This interface should only be used in *syngo*.plaza installations where this kind of messages is important for synchronisation. Otherwise, HL7 order messages and DICOM KOS objects should be used to realize this type of workflow.

2.6.2 Limitations

This proprietary interface has the following limitations:

- Changes performed using HL7 ZPA messages are not persistent. If the HL7 message is received before the DICOM images, the patient context and the Accession Number of the DICOM images will not be corrected. If a DICOM image is stored for an already corrected DICOM study, the Accession Number will be changed back to the original value.
- During the move of the DICOM study to another patient, the visit context gets lost in *syngo*.share.
- In case of an error the messages get skipped and are not blacklisted. Therefore the history of the incoming HL7 ZPA messages could not be granted.
- If the Study Instance UID was changed, a reimport of the DICOM study will cause a UID conflict.

2.6.3 HL7 ZPA Message Structure

syngo.share supports HL7 ZPA messages with the trigger events `I05` and `S05`. `I05` messages use the Accession Number and `S05` the Study Instance UID to identify the DICOM studies. The transmitted HL7 ZPA messages have to meet the following structure:

`MSH PID [MRG] ZPA`

If the patient context of the referenced DICOM studies should be checked the HL7 segment `MRG` should be specified. By default the processor has a modification, which allows the processing of ZPA messages intended for *syngo*.plaza. Therefore `MRG-4` is mapped into `MRG-1` and the segment `ZSP` is mapped into the `MRG` segment, if no `MRG` segment is present.

2.7 HL7 Segments

2.7.1 HL7 MSH Segment

The HL7 MSH segment of an incoming HL7 message has to provide the following information:

Table 2: HL7 MSH Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 Field Separator		R	
2	Encoding Characters		R	
3.1	Sending Application Namespace ID	64	C	
3.2	Sending Application Universal ID	64	C	
4.1	Sending Facility Namespace ID	50	C	
4.2	Sending Facility Universal ID	50	C	
5.1	Receiving Application Namespace ID		O	
5.2	Receiving Application Universal ID		O	
6.1	Receiving Facility Namespace ID	100	C	
6.2	Receiving Facility Universal ID	100	C	
7	HL7 Message Timestamp		O	
9.1	HL7 Message Code		R	
9.2	HL7 Message Trigger Event		R	
9.3	HL7 Message Structure		O	
10	HL7 Message Control ID		R	
11.1	Processing ID		O	
12.1	Version ID		O	
13	HL7 Message Sequence Number		O	
15	Accept Acknowledgment Type		C	
16	Application Acknowledgment Type		C	
18	Character Set		C	yes
20	Alternate Character Set Handling Scheme		C	

The HL7 components `MSH-3.1` and `MSH-3.2` have to define the sending application of the HL7 message uniquely. It is required by *syngo*.share that at least one of the two HL7 components is specified. If `MSH-3.1` is left empty the value of `MSH-3.2` is taken to identify the sending application. In all other cases `MSH-3.1` is evaluated. Note to insert, update, or delete a generic file the determined sending application is used as application ID to identify the third-party system that is responsible for the generic file uniquely (cf. Section 2.3). The HL7 components `MSH-4.1` and `MSH-4.2` are used to identify the sending facility of the HL7 message. It is required that at least one of the two HL7 components is specified because *syngo*.share uses this information to identify the organizational unit that is affected by the HL7 message. If `MSH-4.1` is left empty the value of `MSH-4.2` is taken to retrieve the organizational unit. In all other cases `MSH-4.1` is evaluated. The receiving application (HL7 components `MSH-5.1` and `MSH-5.2`) is evaluated in a similar vein as the sending application. Since the processing of a HL7 messages does not rely on this information, it need not be specified. The HL7 components `MSH-6.1` and `MSH-6.2` are used to obtain the receiving facility of the HL7 message which again is used by *syngo*.share to identify the tenant that is affected by the HL7 message. The HL7 components `MSH-6.1` and `MSH-6.2` are evaluated in a similar way as the sending facility. This means if `MSH-6.1` is not specified `MSH-6.2` has to define the receiving facility. The HL7 fields `MSH-15` and `MSH-16` define on the one hand the processing rules that have to be applied to exchange the HL7 message and on the other hand the conditions under which a HL7 accept acknowledgment message and HL7 application acknowledgment message have to be sent if the enhanced processing rules are in effect. To ensure that the original processing rules are used, both HL7 fields have to be left empty. If the enhanced processing rules should be applied, at least one of the two HL7 fields has to be defined. Last but not least the character sets which have been used to generate the HL7 message have to be specified by the HL7 field `MSH-18`. If more than on character set is in use, the handling scheme of the alternate character sets has to be defined by the HL7 field `MSH-20`. For possible values of `MSH-18` and `MSH-20` confer Section 1.1.

2.7.2 HL7 PID Segment

The HL7 PID segment of an incoming HL7 message has to provide the following information:

Table 3: HL7 PID Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 PID Segment ID		O	
3.1	Patient ID	50	R	yes
3.4.1	Patient ID Assigning Authority Namespace ID	20	C	yes
3.4.2	Patient ID Assigning Authority Universal ID	200	C	yes
3.4.3	Patient ID Assigning Authority Universal ID Type	15	C	yes
3.5	Patient ID Type		R	yes
5.1	Patient Last Name	64	O	yes
5.2	Patient First Name	64	O	yes
5.3	Patient Middle Name	64	O	yes
5.4	Patient Name Suffix	64	O	yes
5.5	Patient Name Prefix	64	O	yes
5.7	Patient Name Type		C	yes
5.8	Patient Name Representation		C	yes
6	Mothers Maiden Name	150	O	
7	Patient Birth Timestamp	24	O	
8.1	Patient Sex	1	O	
10	Patient Race	50	O	

Table 3: HL7 PID Segment, Inbound

Table 3: HL7 PID Segment, Inbound

Index	Description	Length	Optionality	Repeatable
11	Patient Address	350	O	
16	Patient Marital Status	50	O	
17	Patient Religion	20	O	
18	Patient Account Number	250	O	
23	Patient Birth Place	350	O	
24	Multiple Birth Indicator	1	O	
26	Patient Citizenship	50	O	
27	Patient Profession	100	O	
29	Patient Death Timestamp	24	O	
40	Patient Telecommunication Information	200	O	

The HL7 field `PID-1` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 PID segment). The HL7 components `PID-3.1` and `PID-3.5` have to be used to specify the patient ID as well as alternate patient ID of the patient addressed by the HL7 PID segment. The specification of the patient ID is obligatory whereas the specification of an alternate patient ID is optional. To differ between the two IDs, the HL7 component `PID-3.5` has to specify the value `PI` or `PN` if `PID-3.1` defines the patient ID and the value `NH`, `NI`, or `SS` if `PID-3.1` specifies an alternate patient ID. Note that the value `NI` indicates that the alternate patient ID represents a unique national identifier, `NH` states that the alternate patient ID defines a national health service number, and `SS` denotes that the alternate patient ID identifies a social security number. If `PID-3.5` contains the value `GPI`, `PID-3.1` contains the XAD-PID and all patient IDs of the message get linked with this XAD-PID. The assigning authority of the patient ID and alternate patient ID has to be defined by the HL7 component `PID-3.4`. In case of the patient ID, the assigning authority should be specified if an authority differing from the default authority of the site is meant. Otherwise the default authority is used. For the assigning authority of the alternate patient ID the HL7 subcomponent `PID-3.4.1` is evaluated if it is specified and the HL7 subcomponent `PID-3.4.2` is used if `PID-3.4.1` has been left empty. The HL7 components `PID-5.1` to `PID-5.8` are evaluated to obtain the names of the patient. If `PID-5.7` contains a `B` the repetition specifies the birth name. Otherwise it is assumed that the specified name is the legal name which could also be indicated by the value `L`. Because *syngo*.share is able to store the alphabetic, ideographic, and phonetic name of a patient, the HL7 component `PID-5.8` has to specify an `A` in the first case, an `I` for the ideographic name of the patient, and a `P` if the phonetic name of the patient is given.

2.7.3 HL7 MRG Segment

The HL7 MRG segment of an incoming HL7 ADT message consists of the following information:

Table 4: HL7 MRG Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1.1	Prior Patient ID		C	
1.4.1	Prior Patient ID Assigning Authority Namespace ID	20	C	
1.4.2	Prior Patient ID Assigning Authority Universal ID	200	C	
1.4.3	Prior Patient ID Assigning Authority Universal ID Type	15	C	
1.5	Prior Patient ID Type		C	

Table 4: HL7 MRG Segment, Inbound

Table 4: HL7 MRG Segment, Inbound

Index	Description	Length	Optionality	Repeatable
5.1	Prior Visit ID		C	
5.4.1	Prior Visit ID Assigning Authority Namespace ID	20	C	
5.4.2	Prior Visit ID Assigning Authority Universal ID	200	C	
5.4.3	Prior Visit ID Assigning Authority Universal ID Type	15	C	
7	Prior Patient Name		O	

If a HL7 MRG segment is used to merge a patient, the HL7 component `MRG-1.1` has to define the patient ID of the patient that should be invalidated. Furthermore the HL7 component `MRG-1.5` has to specify the value `PI` or `PN` to ensure that *syngo*.share detects the patient ID. If a HL7 MRG segment is used to move or merge a visit, the HL7 component `MRG-5.1` has to specify the visit ID of the visit that should be moved or invalidated. The assigning authority of the prior patient ID (prior visit ID) has to be defined by the HL7 component `MRG-1.4` (`MRG-5.4`) if an authority differing from the default authority of the site is meant. Otherwise the default authority is used. If a HL7 MRG segment is used in combination with HL7 ZPA messages the HL7 component `MRG-1.1` and the HL7 field `MRG-7` could be specified to validate the source patient, the DICOM study belongs to.

2.7.4 HL7 NK1 Segment

The HL7 NK1 segment of an incoming HL7 message consists of the following information:

Table 5: HL7 NK1 Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 NK1 Segment ID		O	
2	Next of Kin Name	250	O	
3	Relationship	50	O	
4	Next of Kin Address	350	O	
40	Next of Kin Telecommunication Information	200	O	

The HL7 field `NK1-1` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 NK1 segment).

2.7.5 HL7 PV1 Segment

The HL7 PV1 segment of an incoming HL7 message consists of the following information:

Table 6: HL7 PV1 Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 PV1 Segment ID		O	
2	Patient Class	50	O	
3	Assigned Patient Location	80	O	

Table 6: HL7 PV1 Segment, Inbound

Table 6: HL7 PV1 Segment, Inbound

Index	Description	Length	Optionality	Repeatable
6	Prior Patient Location	80	O	
7	Attending Doctor	150	O	
8	Referring Doctor	150	O	
9	Consulting Doctor	250	O	
10	Hospital Service	50	O	
11	Temporary Patient Location	80	O	
15	Ambulatory Status	50	O	
16	VIP Indicator	50	O	
17	Admitting Doctor	150	O	
19.1	Visit ID	64	R	
19.4.1	Visit ID Assigning Authority Namespace ID	20	C	
19.4.2	Visit ID Assigning Authority Universal ID	200	C	
19.4.3	Visit ID Assigning Authority Universal ID Type	15	C	
36	Discharge Disposition	50	O	
37	Discharge Patient Location	50	O	
39	Servicing Facility	100	O	
44	Admit Timestamp	24	O	
45	Discharge Timestamp	24	O	
50.1	Alternate Visit ID	64	O	
50.4	Alternate Visit ID Assigning Authority	50	O	
51.1	Visit Indicator		O	

The HL7 field `PV1-1` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 PV1 segment). The HL7 field `PV1-9` should not be defined. Instead a HL7 ROL segment should be added to the HL7 message which specified the consulting doctor. Nevertheless, if the consulting doctor is defined by the HL7 field `PV1-9` but not by a HL7 ROL segment, the HL7 field `PV1-9` is still evaluated. The HL7 component `PV1-19.1` has to define the visit ID and the HL7 component `PV1-51.1` should specify the value V to indicate that the information provided by the HL7 PV1 segment are dedicated to the visit level (note that *syngo*.share does not support visits related to the account level). The assigning authority of the visit ID (alternate visit ID) has to be defined by the HL7 component `PV1-19.4` (`PV1-50.4`). In case of the visit ID, the assigning authority should be specified if an authority differing from the default authority of the site is meant. Otherwise the default is used. For the assigning authority of the alternate visit ID the HL7 subcomponent `PV1-50.4.1` is evaluated if it is specified and the HL7 subcomponent `PV1-50.4.2` is used if `PV1-50.4.1` has been left empty.

2.7.6 HL7 ROL Segment

The HL7 ROL segment of an incoming HL7 message consists of the following information:

Table 7: HL7 ROL Segment, Inbound

Index	Description	Length	Optionality	Repeatable
2	Action Code		R	

Table 7: HL7 ROL Segment, Inbound

Table 7: HL7 ROL Segment, Inbound

Index	Description	Length	Optionality	Repeatable
3.1	Role		R	
4	Person	250	O	

The HL7 component `ROL-3.1` has to specify the value `CP` if the HL7 ROL segment states the consulting doctor. In this particular case the HL7 field `PV1-9` should not be defined.

2.7.7 HL7 GT1 Segment

The HL7 GT1 segment of an incoming HL7 message consists of the following information:

Table 8: HL7 GT1 Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 GT1 Segment ID		O	
3	Guarantor Name	250	O	
5	Guarantor Address	350	O	
8	Guarantor Birth Timestamp	24	O	
12	Guarantor Social Security Number	50	O	
16	Employer Name	250	O	
17	Employer Address	350	O	

The HL7 field `GT1-1` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 GT1 segment).

2.7.8 HL7 ORC Segment

In general the information provided by a HL7 ORC segment as well as the structure of a HL7 ORC segment depend on the type of the underlying HL7 message. The HL7 inbound interface of *syngo*.share just evaluates HL7 ORC segments in combination with HL7 OMI and HL7 ORM messages. HL7 OMI as well as HL7 ORM messages use HL7 ORC segments to insert, update, and delete orders, requested procedures, and scheduled procedure steps (see Section 2.2). For this purpose each HL7 ORC segment of an incoming HL7 OMI and HL7 ORM message has to provide the following information:

Table 9: HL7 ORC Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	Order Control Code		R	
2.1	Placer Order Number	199	C	
2.2	Placer Order Number Assigning Authority Namespace ID	20	C	
2.3	Placer Order Number Assigning Authority Universal ID	200	C	
2.4	Placer Order Number Assigning Authority Universal ID Type	15	C	
3.1	Filler Order Number	199	C	

Table 9: HL7 ORC Segment, Inbound

Table 9: HL7 ORC Segment, Inbound

Index	Description	Length	Optionality	Repeatable
3.2	Filler Order Number Assigning Authority Namespace ID	20	C	
3.3	Filler Order Number Assigning Authority Universal ID	200	C	
3.4	Filler Order Number Assigning Authority Universal ID Type	15	C	
5	Order Status	128	C	
7.4	Start Timestamp	24	C	
10	Order Enterer	255	O	
12	Order Provider	255	O	
13	Enterers Location	80	O	
14	Order Call Back Phone Number	250	O	
16	Order Control Code Reason	250	O	
17	Entering Organization	60	O	
21	Facility	255	O	

The HL7 field `ORC-1` determines the function of the order specified by the HL7 ORC segment. Possible values are `NW`, `DC`, `XO`, and `CA` where `NW` causes the insertion of the order, `DC` gives rise to the discontinuation of the order, `XO` entails the update of the order, and last but not least `CA` causes the deletion of the order. The HL7 components `ORC-2.1` and `ORC-3.1` define the placer order number and filler order number of the order and are used by *syngo*.share to identify it uniquely. So at least one of the two IDs has to be specified. The assigning authority of the placer order number (filler order number) has to be defined by the HL7 components `ORC-2.2` to `ORC-2.4` (`ORC-3.2` to `ORC-3.4`) if an authority differing from the default authority of the site is meant. Otherwise the default authority is used. The HL7 field `ORC-5` defines the status of the order. If the scheduled procedure steps of the order have been scheduled for processing the value `SC` has to be stated. To mark the scheduled procedure steps associated with the order as completed, the value `CM` has to be used. Finally, in case of a HL7 ORM message the HL7 component `ORC-7.4` specifies the start timestamp of the listed scheduled procedure steps. For that reason the specification of the HL7 component `ORC-7.4` is mandatory. If the HL7 ORC segment is part of a HL7 OMI message the HL7 component `ORC-7.4` should not be specified because the start timestamp of the scheduled procedure steps is defined by a separate HL7 TQ1 segment. Note that the HL7 components and HL7 fields `ORC-2.1`, `ORC-3.1`, `ORC-7.4`, `ORC-12`, and `ORC-14` should specify the same values as the HL7 components and HL7 fields `OBR-2.1`, `OBR-3.1`, `OBR-27.4`, `OBR-16`, and `OBR-17` of the associated HL7 OBR segment.

2.7.9 HL7 TQ1 Segment

Each HL7 TQ1 segment of an incoming HL7 OMI message consists of the following information:

Table 10: HL7 TQ1 Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 TQ1 Segment ID		O	
7	Start Timestamp	24	R	

The HL7 field `TQ1-1` should specify a consecutive number (note that HL7 OMI messages might contain more than one HL7 TQ1 segment). The HL7 field `TQ1-7` specifies the start timestamp of the scheduled procedure steps that are part of the HL7 ORC/TQ1/OBR/IPC segment group to which the HL7 TQ1 segment belongs.

2.7.10 HL7 OBR Segment

The information provided by a HL7 OBR segment as well as the structure of a HL7 OBR segment depend, like for a HL7 ORC segment, on the type of the underlying HL7 message. The HL7 inbound interface of *syngo*.share evaluates HL7 OBR segments in combination with HL7 OMI, HL7 ORM, HL7 MDM, and HL7 ORU messages. HL7 OMI and ORM messages make use of the information provided by HL7 OBR segments to insert and update orders, requested procedures, and scheduled procedures steps (see Section 2.2). Within HL7 MDM and HL7 ORU messages the information provided by HL7 OBR segments is just used for informational purposes (see Section 2.3). Although HL7 OMI and HL7 ORM messages have the same purpose the structure of the used HL7 OBR segments is quite different. Each HL7 OBR segment of an incoming HL7 OMI message has to provide the following information:

Table 11: HL7 OBR Segment (HL7 OMI Message), Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 OBR Segment ID		O	
2.1	Placer Order Number	199	C	
2.2	Placer Order Number Assigning Authority Namespace ID	20	C	
2.3	Placer Order Number Assigning Authority Universal ID	200	C	
2.4	Placer Order Number Assigning Authority Universal ID Type	15	C	
3.1	Filler Order Number	199	C	
3.2	Filler Order Number Assigning Authority Namespace ID	20	C	
3.3	Filler Order Number Assigning Authority Universal ID	200	C	
3.4	Filler Order Number Assigning Authority Universal ID Type	15	C	
4.1	Universal Service ID	128	O	
4.2	Universal Service Description	128	O	
12	Danger Code	60	O	
13	Relevant Clinical Information	999	O	
15	Specimen Source	300	O	
16	Order Provider	255	O	
17	Order Call Back Phone Number	250	O	
30	Transportation Mode	50	O	
31	Reason For Study	300	O	
34	Technician	200	O	
44.1	Procedure Code	128	O	
44.2	Procedure Description	128	O	
44.5	Requested Procedure Description	128	O	

The HL7 field `OBR-1` should specify a consecutive number (note that HL7 OMI messages might contain more than one HL7 OBR segment). The HL7 components `OBR-2.1` and `OBR-3.1` have to define the placer order number and filler order number of the order. Since *syngo*.share uses the two IDs to identify orders, at least one of the two IDs has to be specified. The assigning authority of the placer order number (filler order number) has to be defined by the HL7 components `ORC-2.2` to `ORC-2.4` (`ORC-3.2` to `ORC-3.4`) if an authority differing from the default authority of the site is meant. Otherwise the default authority is used. Finally the HL7 component `OBR-4.1` specifies the universal service ID — an identifier for the order which usually comes from an industry standard such as SNOMED or LOINC. Note that the HL7 components and HL7 fields `OBR-2.1`, `OBR-3.1`, `OBR-16`, and `OBR-17` should specify the same values as the HL7 components and HL7 fields `ORC-2.1`, `ORC-3.1`, `ORC-12`, and `ORC-14` of the associated HL7 ORC segment.

The HL7 OBR segment of an incoming HL7 ORM message has to provide the following information:

Table 12: HL7 OBR Segment (HL7 ORM Message), Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 OBR Segment ID		O	
2.1	Placer Order Number	199	C	
2.2	Placer Order Number Assigning Authority Namespace ID	20	C	
2.3	Placer Order Number Assigning Authority Universal ID	200	C	
2.4	Placer Order Number Assigning Authority Universal ID Type	15	C	
3.1	Filler Order Number	199	C	
3.2	Filler Order Number Assigning Authority Namespace ID	20	C	
3.3	Filler Order Number Assigning Authority Universal ID	200	C	
3.4	Filler Order Number Assigning Authority Universal ID Type	15	C	
4.1	Universal Service ID	128	O	
4.2	Universal Service Description	128	O	
4.4	Protocol Code	128	O	
4.5	Protocol Description	128	O	
4.11	Scheduled Procedure Step Description	128	O	
12	Danger Code	60	O	
13	Relevant Clinical Information	999	O	
15	Specimen Source	300	O	
16	Order Provider	255	O	
17	Order Call Back Phone Number	250	O	
18	Accession Number	255	O	
19	Requested Procedure ID	199	R	
20	Scheduled Procedure Step ID	199	R	
21	Scheduled Station AET	255	O	
24	Modality	50	O	
27.4	Start Timestamp	24	R	
30	Transportation Mode	50	O	
31	Reason For Study	300	O	
34	Technician	200	O	
44.1	Procedure Code	128	O	
44.2	Procedure Description	128	O	
44.5	Requested Procedure Description	128	O	

The HL7 fields and HL7 components `OBR-1`, `OBR-2.1` to `OBR-2.4`, `OBR-3.1` to `OBR-3.4`, and `OBR-4.1` have to be specified in the same way as described for HL7 OMI messages. The HL7 field `OBR-18` is evaluated to obtain the accession number of the order. If the accession number is not specified, *syngo*.share generates an appropriate ID automatically. For the assigning authority of the accession number either the assigning authority of the filler order number is used or the assigning authority of placer order number if the filler order number is not defined. The HL7 field OBR-19 has to specify a unique ID for the requested procedure addressed by the HL7 ORM message. It is used by *syngo*.share as an alternate unique identifier for the stated requested procedure. Likewise the HL7 field OBR-20 has to define a unique ID for the scheduled procedure step specified by the HL7 ORC/OBR segment group to which the HL7 OBR segment belongs. The HL7 component OBR-27.4 specifies the start timestamp of the listed scheduled procedure steps. For that reason the specification of the HL7 component ORC-7.4 is mandatory. Note that the HL7 components and HL7 fields OBR-2.1, OBR-3.1, OBR-27.4, OBR-16, and OBR-17 should specify the same values as the HL7 components and HL7 fields ORC-2.1, ORC-3.1, ORC-7.4, ORC-12, and ORC-14 of the associated HL7 ORC segment.

If a HL7 OBR segment is part of an incoming HL7 MDM or HL7 ORU message which inserts, updates, or deletes a generic file it has to possess the following structure:

Table 13: HL7 OBR Segment (HL7 MDM and ORU Message), Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 OBR Segment ID		O	
2.2	Placer Order Number Assigning Authority Namespace ID	20	C	
2.3	Placer Order Number Assigning Authority Universal ID	200	C	
2.4	Placer Order Number Assigning Authority Universal ID Type	15	C	
3.2	Filler Order Number Assigning Authority Namespace ID	20	C	
3.3	Filler Order Number Assigning Authority Universal ID	200	C	
3.4	Filler Order Number Assigning Authority Universal ID Type	15	C	
18.1	Accession Number	255	O	
18.2	Accession Number Assigning Authority Namespace ID	20	C	
18.3	Accession Number Assigning Authority Universal ID	200	C	
18.4	Accession Number Assigning Authority Universal ID Type	15	C	
24	Practice Setting Code	256	C	
31	Reason For Study		O	

The HL7 field OBR-1 should specify the value 1 (note that HL7 MDM and HL7 ORU messages do not support the processing of more than one HL7 OBR segment). The HL7 field OBR-18 is evaluated to obtain the accession number of the generic file. As accession number assigning authority the assigning authority of the filler order number is used or the one of the placer order number if HL7 Field OBR-3 has been left empty. The HL7 components OBR-2.2 to OBR-2.4 or OBR-3.2 to OBR-3.4 should be defined, if an authority differing from the default authority of the site is meant. Otherwise the default authority is used. The HL7 field OBR-24 has to specify the practice setting code of the generic file referenced by the HL7 message. In case of an incoming HL7 MDM message which inserts or updates a generic file the specification of the practice setting code is required if the generic file should be registered in an affinity domain (see IHE IT Infrastructure Technical Framework for further information). To this end the following HL7 components have to be defined:

Table 14: HL7 OBR-24 Field, Inbound

Index	Description	Length	Optionality
24.1	Practice Setting Code ID	256	R
24.2	Practice Setting Code Display Name	256	O

Table 14: HL7 OBR-24 Field, Inbound

Table 14: HL7 OBR-24 Field, Inbound

Index	Description	Length	Optionality
24.3	Practice Setting Code Coding Scheme	256	R

In case of HL7 ORU messages the specification of the practice setting code is optional because HL7 ORU messages do not provide enough information to register generic files in affinity domains.

2.7.11 HL7 IPC Segment

Each HL7 IPC segment of an incoming HL7 OMI message consists of the following information:

Table 15: HL7 IPC Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1.1	Accession Number	255	R	
1.2	Accession Number Assigning Authority Namespace ID	20	C	
1.3	Accession Number Assigning Authority Universal ID	200	C	
1.4	Accession Number Assigning Authority Universal ID Type	15	C	
2.1	Requested Procedure ID	199	R	
3.1	Study Instance UID	255	R	
4.1	Scheduled Procedure Step ID	199	R	
5.1	Modality	50	O	
6.1	Protocol Code	128	O	
6.2	Protocol Description	128	O	
6.5	Scheduled Procedure Step Description	128	O	
9	Scheduled Station AET	255	O	

The HL7 component `IPC-1.1` has to define the accession number of the order. The assigning authority of the accession number has to be defined by the HL7 components `IPC-1.2` to `IPC-1.4` if an authority differing from the default authority of the site is meant. Otherwise the default authority is used. The HL7 component `IPC-2.1` has to specify a unique ID for the requested procedure addressed by the HL7 ORC/TQ1/OBR/IPC segment group to which the HL7 IPC segment belongs. It is used by *syngo*.share as an alternate unique identifier for the stated requested procedure. Similarly the HL7 component `IPC-3.1` has to define the Study Instance UID of the DICOM Study that should be associated with requested procedure addressed by the underlying HL7 ORC/TQ1/OBR/IPC segment group. As mentioned in Section 2.2 the Study Instance UID is used as unique identifier for the affected requested procedure. The HL7 component `IPC-4.1` has to define a unique ID for the scheduled procedure step specified by the HL7 IPC segment. Note that *syngo*.share links the specified scheduled procedure step with the requested procedure addressed by the corresponding HL7 ORC/TQ1/OBR/IPC segment group.

2.7.12 HL7 TXA Segment

The HL7 TXA segment of an incoming HL7 MDM message consists of the following information:

Table 16: HL7 TXA Segment (HL7 MDM Message), Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 TXA Segment ID		O	
2.1	Generic File Type ID	64	C	
2.2	Generic File Type Description	100	C	
2.3	Generic File Type Coding System ID	64	C	
3	Generic File Content Presentation	256	C	
6	Generic File Creation Timestamp	24	C	
9	Generic File Originator	1000	C	
12.1	Generic File ID	256	R	
13.1	Prior Generic File ID		C	
16	Generic Filename	256	C	
17	Generic File Completion Status	50	O	
18	Generic File Confidentiality Status	256	C	
22	Generic File Authenticator	1000	C	
24.1	Generic Container ID	256	R	
24.2	Generic Container Description	250	O	
24.3	Generic Container Coding System ID	64	R	
25	Generic File Description	250	C	

The HL7 field `TXA-1` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 TXA segment). The information stated by the HL7 components `TXA-2.1`, `TXA-2.2`, and `TXA-2.3` as well as the HL7 fields `TXA-3`, `TXA-9`, `TXA-18` and `TXA-22` are primarily needed to register the generic file specified by the HL7 TXA segment in an affinity domain. In such a case *syngo*.share requires that `TXA-2.1`, `TXA-2.2`, and `TXA-2.3` specify the class code, `TXA-3` the format code, `TXA-9` the author, healthcare facility type code, and institution, `TXA-18` the confidentiality code, and `TXA-22` the authenticator of the generic file, by providing the following information:

Table 17: HL7 TXA Segment (HL7 MDM Message), Additional Fields, Inbound

Index	Description	Length	Optionality
2.1	Class Code ID	256	R
2.2	Class Code Display Name	256	O
2.3	Class Code Coding Scheme	256	R
3.1	Format Code ID	256	R
3.2	Format Code Display Name	256	O
3.3	Format Code Coding Scheme	256	R
9.1	Originator Person ID	1000	O
9.2	Originator Last Name	1000	R
9.3	Originator First Name	1000	R
9.4	Originator Middle Name	1000	O
9.5	Originator Name Suffix	1000	O
9.6	Originator Name Prefix	1000	O
9.14.1	Healthcare Facility Type Code ID	1000	R

Table 17: HL7 TXA Segment (HL7 MDM Message), Additional Fields, Inbound

Table 17: HL7 TXA Segment (HL7 MDM Message), Additional Fields, Inbound

Index	Description	Length	Optionality
9.14.2	Healthcare Facility Type Code Display Name	1000	O
9.14.3	Healthcare Facility Type Code Coding Scheme	1000	R
9.23.1	Institution OID	1000	R
9.23.2	Institution Name	1000	R
18.1	Confidentiality Code ID	256	C
18.2	Confidentiality Display Name	256	C
18.3	Confidentiality Code Coding Scheme	256	C
22.1	Authenticator Person ID	1000	O
22.2	Authenticator Last Name	1000	R
22.3	Authenticator First Name	1000	R
22.4	Authenticator Middle Name	1000	O
22.5	Authenticator Name Suffix	1000	O
22.6	Authenticator Name Prefix	1000	O

The HL7 field `TXA-6` should always be specified. However *syngo*.share only requires the specification of the creation timestamp if the generic file has to be registered in an affinity domain. The HL7 component `TXA-12.1` has to specify the ID of the referenced generic file which is used together with the application ID, specified by the HL7 component `MSH-3.1` or `MSH-3.2` (`MSH-3.2` is just used if `MSH-3.1` is not defined), to identify the generic file uniquely. If the generic file ID is changed, the old generic file ID has to be specified by the HL7 component `TXA-13.1` to guarantee that updates can be performed correctly. The name of the generic file (inclusive its file extension) is obtained from the HL7 field `TXA-16`. Its specification is mandatory if the generic file should be inserted or updated by a new version which possesses a different file extension. The HL7 components `TXA-24.1` and `TXA-24.3` define the generic container ID and coding system ID of the generic container which contains the generic file. Finally, the HL7 field `TXA-25` has to specifies a description of the generic file. If the generic file should be registered in an affinity domain, the specification of `TXA-25` is mandatory. Note that the HL7 field `TXA-6` should specify the same value as the HL7 fields `OBR-7` and `OBX-14`.

2.7.13 HL7 OBX Segment

The information provided by a HL7 OBX segment as well as the structure of a HL7 OBX segment depend on the type of the object or observation that is addressed by the HL7 OBX segment. The HL7 inbound interface of *syngo*.share evaluates HL7 OBX segments in combination with HL7 MDM and HL7 ORU messages in order to insert, update, or delete generic files or to reference DICOM Studies or generic containers by specifying reference pointers. HL7 OBX segments which belong to HL7 MDM or HL7 ORU messages that are intended to insert, update, or delete generic files have to be further divided into HL7 OBX segments which belong to HL7 MDM messages and HL7 OBX segments which are part of HL7 ORU messages. A HL7 OBX segment that belongs to an incoming HL7 MDM message which is used to insert or update a generic file has to provide the following information:

Table 18: HL7 OBX Segment (HL7 MDM and ORU Message), Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 OBX Segment ID		O	

Table 18: HL7 OBX Segment (HL7 MDM and ORU Message), Inbound

Table 18: HL7 OBX Segment (HL7 MDM and ORU Message), Inbound

Index	Description	Length	Optionality	Repeatable
2	Value Type		R	
4	HL7 OBX Segment Sub-ID		O	
5	Value		R	
8	Generic File Interpretation Code	1000	C	
11	Generic File Result Status		O	
14	Generic File Creation Timestamp		O	

The HL7 fields `OBX-1` and `OBX-4` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 OBX segment). The HL7 field `OBX-2` has to specify the type of the HL7 field `OBX-5`, in particular ED or RP. If `OBX-2` specifies the value ED the binary data of the generic file addressed by the HL7 OBX segment has to be encapsulated in the HL7 MDM message. In this special case the HL7 field `OBX-5` has to meet the following structure:

Table 19: HL7 OBX-5 Field (Binary Data), Inbound

Index	Description	Content	Optionality
5.2	Mime Type	100	R
5.3	Mime Subtype	100	O
5.4	Encoding		R
5.5	Binary Data		R

The HL7 components `OBX-5.2` and `OBX-5.3` specify the MIME type and MIME subtype of the generic file. Valid values can be obtained from the website of the Internet Assigned Number Authority. Since it is not allowed to transmit non-printable characters within a HL7 message, the binary data of the generic file, specified by the HL7 component `OBX-5.5`, has to be encoded. To ensure that *syngo*.share can decode the binary data, the used encoding has to be specified by the HL7 component `OBX-5.4` (see Section 2.3 for possible values). If the HL7 field `OBX-2` specifies the value RP the binary data of the generic file has to be referenced using an UNC-path. To this end `OBX-5` has to define of the following information:

Table 20: HL7 OBX-5 Field (UNC-Path), Inbound

Index	Description	Content	Optionality
5.1	UNC-Path		R
5.3	Mime Type	100	R
5.4	Mime Subtype	100	O

The HL7 field `OBX-8` defines an interpretation code. If the generic file should be registered in an affinity domain *syngo*.share requires that the specified interpretation code defines the content type code of the generic file (see IHE IT Infrastructure Technical Framework for further information). If that is the case the following HL7 components have to be defined:

Table 21: HL7 OBX-8 Field, Inbound

Index	Description	Length	Optionality
8.1	Content Type Code ID	1000	R
8.2	Content Type Code Display Name	1000	O
8.3	Content Type Code Coding Scheme	1000	R

Otherwise the HL7 field `OBX-8` can be left empty. Finally, the HL7 field `OBX-11` should always specify the value `F`. Note that the HL7 field `OBX-14` should specify the same value as the HL7 field `TXA-6`.

In contrast, a HL7 OBX segment that belongs to an incoming HL7 ORU message, which is used to insert, update, or delete a generic file, has to provide the following information:

Table 22: HL7 OBX Segment (HL7 ORU Message), Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 OBX Segment ID		O	
2	Value Type		C	
3.1	Generic File ID	256	R	
3.2	Generic File Description	250	O	
3.4	Generic Container ID	256	R	
3.5	Generic Container Description	250	O	
3.6	Generic Container Coding System ID	64	O	
3.9	Generic Filename	256	C	
4	HL7 OBX Segment Sub-ID		O	
5	Value		C	
8	Generic File Interpretation Code	1000	O	
11	Generic File Result Status		R	
14	Generic File Creation Timestamp	24	O	

The HL7 fields `OBX-1` and `OBX-4` should specify the value `1` (note that *syngo*.share cannot process more than one HL7 OBX segment). The HL7 field `OBX-2` specifies the type of the HL7 field `OBX-5`. If the HL7 field `OBX-5` contains the binary data of the generic file the value `ED` has to be stated. If the binary data of the generic file is referenced via an UNC-path, the value `RP` has to be defined. If no binary data is given (for example in case of a pure metadata update) the HL7 field `OBX-2` should be left empty. The HL7 component `OBX-3.1` has to specify the ID of the given generic file which is used together with the application ID, specified by the HL7 component `MSH-3.1` or `MSH-3.2` (`MSH-3.2` is just used if `MSH-3.1` is not defined), to identify the generic file uniquely. The HL7 components `OBX-3.4` and `OBX-3.4` define the generic container ID and coding system ID of the generic container which contains the referenced generic file. The name of the generic file (inclusive its file extension) is obtained from the HL7 component `OBX-3.9`. Its specification is mandatory if the generic file should be inserted or updated by a new version which possesses a different file extension. To interpret the value of the HL7 field `OBX-5` *syngo*.share uses the type information specified by the HL7 field `OBX-2`. If `OBX-2` specifies the value `ED` the binary data of the generic file has to be encapsulated in the HL7 ORU message and if `OBX-2` specifies the value `RP` the binary data of the generic file has to be referenced using an UNC-path. If `OBX-2` is not defined `OBX-5` also should not be specified. Note that the structure of the HL7 field `OBX-5` varies depending on whether `OBX-2` specifies the value `ED` or `RP` (see description of HL7 OBX segments of incoming HL7 MDM messages which are used to insert or update generic files). The HL7 field `OBX-8` defines an interpretation code. In case of an incoming HL7 MDM message which inserts or updates a generic file this information is used as content type code, if the generic file is registered in an affinity domain. In case of HL7 ORU messages that never happens because HL7 ORU messages do not provide enough information to register generic files in affinity domains. So the specification of the HL7 field OBX-8 is unregulated. The HL7 field OBX-11 is used to differ between HL7 ORU messages which insert or update generic files and HL7 ORU messages which delete generic files. In the former case OBX-11 has to specify a F whereas in the latter case a D has to be defined. Last but not least the HL7 field OBX-14 defines the creation timestamp of the generic file. It should be defined whenever a generic file is inserted or the binary data of a generic file is updated. Note that the HL7 fields OBX-11 and OBX-14 should specify the same values as the HL7 fields OBR-25 and OBR-7.

If a HL7 OBX segment is used to reference a document (DICOM Study or generic container) via a reference pointer it has to specify the following information:

Table 23: HL7 OBX Segment, Referencing Documents, Inbound

Index	Description	Length	Optionality	Repeatable
1	HL7 OBX Segment ID		O	
2	Value Type		R	
4	HL7 OBX Segment Sub-ID		O	
5.1	Reference Pointer		R	
11	Document Result Status		O	
14	Document Creation Timestamp		O	

The HL7 fields OBX-1 and OBX-4 should specify the value 1 (note that *syngo*.share cannot process more than one HL7 OBX segment). The HL7 field OBX-2 specifies the type of the HL7 field OBX-5. Since the document addressed by the HL7 OBX segment is identified via a reference pointer the value RP has to be stated. The HL7 component OBX-5.1 has to specify the reference pointer which identifies the affected document. The HL7 field OBX-11 describes the status of the referenced document and should specify the value D if it has been deleted and the value F otherwise. Note that the HL7 fields OBX-11 and OBX-14 should specify the same values as the HL7 fields OBR-25 and OBR-7.

2.7.14 HL7 PRT Segment

Each HL7 PRT segment of an incoming HL7 ORU message consists of the following information:

Table 24: HL7 PRT Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1.1	Login Name	255	R	
2	Action Code		R	
4	Participation		O	
5	Person		O	
8	Organization		O	
9	Location		O	
10	Device		O	

The HL7 component PRT-1.1 has to specify the login name of the addressed user. As action code (HL7 field PRT-2) one of the values specified in Section 2.4 has to be defined.

2.7.15 HL7 ZDS Segment

The HL7 ZDS segment of an incoming HL7 ORM message consists of the following information:

Table 25: HL7 ZDS Segment, Inbound

Index	Description	Length	Optionality	Repeatable
1.1	Study Instance UID		R	

The HL7 component ZDS-1.1 has to define the Study Instance UID of the DICOM Study that should be associated with the requested procedure addressed by the HL7 ORM message. As mentioned in Section 2.2 the Study Instance UID is used as unique identifier for the affected requested procedure.

2.7.16 HL7 ZPA Segment

The HL7 ZPA segment of an incoming HL7 ZPA I05 message consists of the following information:

Table 26: HL7 ZPA Segment (HL7 ZPA I05 Message), Inbound

Index	Description	Length	Optionality	Repeatable
2.1.1	Source Accession Number		R	
2.1.2	Source Accession Number Assigning Authority Namespace ID	20	C	
2.1.3	Source Accession Number Assigning Authority Universal ID	200	C	
2.1.4	Source Accession Number Assigning Authority Universal ID Type	15	C	
2.2.1	Target Accession Number		O	
2.2.2	Target Accession Number Assigning Authority Namespace ID	20	C	
2.2.3	Target Accession Number Assigning Authority Universal ID	200	C	
2.2.4	Target Accession Number Assigning Authority Universal ID Type	15	C	

The HL7 subcomponent ZPA-2.1.1 has to define the Source Accession Number of the DICOM studies that should be processed. HL7 subcomponent ZPA-2.2.1 could be set, to change the Accession Number of the given DICOM studies. The assigning authority of the Source Accession Number (Target Accession Number) has to be defined by the HL7 subcomponents ZPA-2.1.2 to ZPA-2.1.4 (ZPA-2.2.2 to ZPA-2.2.4) if an authority differing from the default authority of the site is meant. Otherwise the default authority is used.

The HL7 ZPA segment of an incoming HL7 ZPA S05 message consists of the following information:

Table 27: HL7 ZPA Segment (HL7 ZPA S05 Message), Inbound

Index	Description	Length	Optionality	Repeatable
1.1	Source Study Instance UID		R	
1.2	Target Study Instance UID		O	

Table 27: HL7 ZPA Segment (HL7 ZPA S05 Message), Inbound

Table 27: HL7 ZPA Segment (HL7 ZPA S05 Message), Inbound

Index	Description	Length	Optionality	Repeatable
2.2.1	Target Accession Number		O	
2.2.2	Target Accession Number Assigning Authority Namespace ID	20	C	
2.2.3	Target Accession Number Assigning Authority Universal ID	200	C	
2.2.4	Target Accession Number Assigning Authority Universal ID Type	15	C	

The HL7 component `ZPA-1.1` has to define the Source Study Instance UID of the DICOM study that should be processed. HL7 component `ZPA-1.2` could be set, to change the Study Instance UID and HL7 subcomponent `ZPA-2.2.1` could be set, to change the Accession Number of the DICOM study. The assigning authority of the Target Accession Number has to be defined by the HL7 subcomponents `ZPA-2.2.2` to `ZPA-2.2.4` if an authority differing from the default authority of the site is meant. Otherwise the default authority is used.

3 HL7 Outbound Interface

3.1 Routing of Generic Files

3.1.1 Operating Mode

In some workflows it might be necessary to route generic files that have been archived by *syngo*.share to third-party systems. To make use of this feature an action of type GenericRouter has to be configured for the module *EventServer* (see the *syngo*.share System Documentation for further information). Once a GenericRouter has been created, it automatically routes generic files via HL7 MDM messages to a configured third-party system. If an already routed generic file is modified, the routing is repeated to ensure that the receiving third-party system is up to date.

In general all possible types of generic files can be routed. Each HL7 MDM message is dedicated to exactly one generic file. The binary data of a generic file is transmitted either by reference or by value. In the former case the generic file is written to a share and referenced by an UNC-path which meets the following format:

- Windows: \\<host>\<share>\<path>
- Linux: //<host>/<share>/<path>

In the latter case the data is encapsulated in the HL7 MDM message. Since it is not allowed to transmit non-printable characters within a HL7 message *syngo*.share encodes binary data as a Base64 string. To ensure that the receiving third-party system can identify a generic file uniquely *syngo*.share specifies two IDs: a generic file ID and an application ID. The application ID is defined to ensure that the receiving third-party system can identify *syngo*.share uniquely whereas the generic file ID is stated to allow the third-party system to identify the generic file among the routed generic files. Note that these IDs are not identical with the original IDs used by *syngo*.share to identify the generic file. Each generic file is archived in *syngo*.share within a so called generic container. Generic containers are used by *syngo*.share to group generic files. They are identified by a generic container ID and a coding system ID. To inform a third-party system about the generic container to which a routed generic file belongs, its generic container ID and coding system ID are specified. Note that a generic container ID and coding system ID need not be unique since generic containers might be reused for patients and shared between third-party systems. However, to ensure that generic containers can be used in a meaningful way it is guaranteed by *syngo*.share that a patient does not possess two generic containers with identical generic container ID and coding system ID with regard to a certain visit and organizational unit. In general *syngo*.share routes a generic file if one of the following events occurs:

- import of a generic file
- undeletion of a generic file
- update (metadata or binary data) of a generic file
- update of a patient associated with a generic file
- update of a visit associated with a generic file

3.1.2 HL7 MDM Message Structure

To route generic files via HL7 MDM messages *syngo*.share uses the trigger event `T02` as proposed by the HL7 Messaging Standard. Thereby the transmitted HL7 MDM messages exhibit the following structure:

`MSH PID [PV1] [ORC] OBR TXA OBX`

The generic file ID is specified by the HL7 component `TXA-12.1` and the application ID is defined by the HL7 component `MSH-3.1`. Note that these IDs are not identical with the original IDs used by *syngo*.share to identify the routed generic file. The original generic file ID and application ID are defined by the HL7 components `OBX-3.1` and `OBX-3.3` however, according to the HL7 Messaging Standard, they should not be used to identify the routed generic file. The generic container ID and the coding system ID are specified by the HL7 components `TXA-24.1` and `TXA-24.3`. The HL7 PV1 and HL7 ORC segment are present if and only if the underlying generic file consists of visit and order information.

3.2 Referencing of DICOM Studies and Generic Containers

3.2.1 Operating Mode

In *syngo*.share a reference pointer is a string which refers either to a DICOM Study or generic container. It can be used by third-party systems (e.g., HIS, RIS, etc.) to access, share, or delete a DICOM Study or generic container without holding the actual data. To this end the URL interface of module *view* has to be called (see Interface and Integration Manual). To receive reference pointers an action of type ReferencePointerNotifier has to be configured for the module *EventServer* (see the *syngo*.share System Documentation for further information). Once a ReferencePointerNotifier has been created, it automatically sends reference pointers via HL7 ORU messages to a configured third-party system. Each HL7 ORU message is dedicated to exactly one reference pointer. If a DICOM Study or generic container is modified in a way such that the changes influence the operational reliability of a previously sent reference pointer (for example, a DICOM Study or generic container is deleted) *syngo*.share informs the third-party system about those changes. To this end *syngo*.share transmits a HL7 ORU message which either updates or deletes the affected reference pointer (note that there is no difference between HL7 ORU messages which insert reference pointers and HL7 ORU messages which update existing reference pointers). In case that a third-party system cannot deal with HL7 ORU messages which update reference pointers, *syngo*.share offers the possibility to delete reference pointers before they are updated. If so, *syngo*.share first sends a HL7 ORU message which deletes a reference pointer before it transmits the actual HL7 ORU message that updates the affected reference pointer. In general *syngo*.share inserts, updates, or deletes a reference pointer if one of the following events occurs:

- import of a DICOM Study or generic container
- undeletion of a DICOM Study or generic container
- update (metadata or binary data) of a DICOM Study or generic container
- deletion of a DICOM Study or generic container
- update of a patient associated with a DICOM Study or generic container
- update of a visit associated with a DICOM Study or generic container

In addition, reference pointers of DICOM Studies or generic containers which do not any longer satisfy the conditions specified by the configured ReferencePointerNotifier are also deleted. In case the settings of the receiving application regarding assigning authorities would conflict with authorities given in the HL7 ORU message, it is not sent. Further details about reference pointers can be found in the System Documentation of *syngo*.share.

3.2.2 HL7 ORU Message Structure

To insert, update, and delete reference pointers via HL7 ORU messages *syngo*.share uses the trigger event `R01`. So, the transmitted HL7 ORU messages exhibit the following structure:

`MSH` `PID` [`PV1`] {[`ORC`] `OBR`} `OBX`

If the DICOM study, generic container, or generic file is not associated with an order only `OBR` is given. Otherwise the pair `ORC` and `OBR` is given per associated order, which in case of a generic container or generic file can be multiple times. To distinguish between HL7 ORU messages which insert or update reference pointers and HL7 ORU messages which delete reference pointers, the HL7 fields `OBR-25` and `OBX-11` are set to `F` in the former case and `D` in the latter one. The HL7 segments `PV1` and `ORC` are present if and only if the underlying document consists of visit or order information.

3.3 Query Messages

3.3.1 Operating Mode

The *ScanImportServer* module of *syngo*.share, which is designed to import PDF files generated by a mass scanning device, uses data from a generated index file to map the imported PDF file to a patient/visit combination known to *syngo*.share. However, if the patient/visit combination is not known to *syngo*.share or the data read from the index file ist not sufficient, a HL7 QRY message can be configured to be sent to a third-party sytem, typically a HIS or RIS. The response messages are not evaluated directly by *syngo*.share, but can be forwarded to an arbitrary third-party system.

3.3.2 HL7 QRY message structure

syngo.share uses original mode HL7 QRY messages. The supported trigger events of HL7 QRY are `A19`, indicating that an HL7 ADR response is expected, and `Q01`, indicating that an HL7 DSR response is expected. The messages therefore exhibit the following structure:

```
MSH QRD
```

3.4 Acknowledging of Processed HL7 Messages

3.4.1 Operating Mode

If HL7 messages are received by *syngo*.share according to the enhanced processing rules it might be necessary to exchange further information regarding the processing of HL7 messages via so called HL7 application acknowledgment messages. To this end an action of type Hl7MessageProcessingNotifier has to be configured for the module *EventServer* (see the *syngo*.share System Documentation for further information). Once a Hl7MessageProcessingNotifier has been created the transmission of a HL7 application acknowledgment message is automatically conducted as soon as a HL7 message has been processed which requires the exchange of further information.

In general *syngo*.share is able to generate and transmit HL7 application acknowledgment messages for all processable HL7 messages. Each HL7 application acknowledgment message is dedicated to exactly one HL7 message and specifies the status of the processing. To ensure that a third-party system knows which HL7 message has been acknowledged by a certain HL7 application acknowledgment message, the control ID of the initial HL7 message is specified. In general a HL7 application acknowledgment message is transmitted if the following event occurs:

- processing of a HL7 message

In addition it must be guaranteed that the application acknowledgment type (see HL7 field `MSH-16`) specified by the original HL7 messages goes well with the processing status of the HL7 message. That means if the original HL7 message specifies the application acknowledgment type `SU` (`ER`), a HL7 application acknowledgment message is sent if and only if the processing of the HL7 message succeeded (failed). If the original HL7 message specifies the application acknowledgment type `AL`, a HL7 application acknowledgment message is sent regardless of whether the processing of the HL7 message succeeded or failed.

3.4.2 HL7 Application Acknowledgment Message Structure

To notify third-party systems about the processing of HL7 message *syngo*.share supports the transmission of HL7 application acknowledgment messages related to HL7 ADT messages with trigger events `A01` to `A04`, `A06` to `A08`, `A11` to `A13`, `A18`, `A23`, `A28`, `A29`, `A31`, `A34`, `A40`, `A42`, `A45`, `A47`, or `A50`, HL7 OMI messages with trigger event `O23`, HL7 ORM messages with trigger event `O01`, HL7 MDM messages with trigger events `T02`, `T09`, `T10`, or `T11`, and HL7 ORU messages with trigger event `R01`. The transmitted HL7 application acknowledgment messages meet the following structure:

```
MSH MSA
```

The trigger event specified by a HL7 application acknowledgment message depend on the trigger event specified by the acknowledged HL7 message. The status of the processed HL7 message is specified by the HL7 field `MSA-1`.

3.5 HL7 Segments

3.5.1 HL7 MSH Segment

The HL7 MSH segment of an outgoing HL7 message consists of the following information:

Table 28: HL7 MSH Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 Field Separator	|	R	
2	Encoding Characters	^~\&	R	
3.1	Sending Application		R	
4.1	Sending Facility Namespace ID		C	
4.2	Sending Facility Universal ID		R	
4.3	Sending Facility Universal ID Type	L	R	
5.1	Receiving Application		R	
6.1	Receiving Facility		R	
7	Timestamp of HL7 Message		R	
9.1	HL7 Message Code		R	
9.2	HL7 Message Trigger Event		R	
9.3	HL7 Message Structure		R	
10	HL7 Message Control ID		R	
11.1	Processing ID	P	R	
12.1	Version ID	`2.8.1`	R	
13	HL7 Message Sequence Number		R	
15	Accept Acknowledgment Type		C	
16	Application Acknowledgment Type		C	
18	Character Set		R	yes
20	Alternate Character Set Handling Scheme		C	

The HL7 component `MSH-3.1` specifies per default the project name of the server which transmits the HL7 message. In case that the default value is not adequate, the sending application can be configured. The HL7 component `MSH-4.1` specifies the organizational unit of the underlying DICOM Study, generic container, or generic file if the HL7 message is used to administrate a reference pointer or route a generic file. Like for `MSH-3.1` the default value can be changed by configuring an adequate value. The HL7 component `MSH-4.2` specifies the default receiving facility of the affected tenant and cannot be configured. The values that should be used to specify the HL7 components `MSH-5.1` and `MSH-6.1` as well as the HL7 fields `MSH-18` and `MSH-20` have to be configured. Note that `MSH-20` is always given if `MSH-18` specifies more than one character set. The values of the HL7 fields `MSH-15` and `MSH-16` depend on the processing rules that have been configured. If the original processing rules are applied to exchange the HL7 message both HL7 fields are left empty. If the enhanced processing rules are are used, the transmission of a HL7 accept acknowledgment messages is required but the exchange of a HL7 application acknowledgment messages is denied. To this end the HL7 field `MSH-15` specifies the value `AL` and the HL7 field `MSH-16` defines the value `NE`.

3.5.2 HL7 PID Segment

The HL7 PID segment of an outgoing HL7 message consists of the following information:

Table 29: HL7 PID Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 PID Segment ID	1	O	
3.1	Patient ID		R	yes
3.4.1	Patient ID Assigning Authority Namespace ID		R	yes
3.4.2	Patient ID Assigning Authority Universal ID		R	yes
3.4.3	Patient ID Assigning Authority Universal ID Type		R	yes
3.5	Patient ID Type		R	yes
5.1	Patient Last Name		O	
5.2	Patient First Name		O	
5.3	Patient Middle Name		O	
5.4	Patient Name Suffix		O	
5.5	Patient Name Prefix		O	
5.7	Patient Name Type		R	
5.8	Patient Name Representation	A	R	
7	Patient Birth Timestamp		O	
8.1	Patient Sex		O	
10	Patient Race		O	
11	Patient Address		O	
18	Patient Account Number		O	
26	Patient Citizenship		O	

The HL7 components `PID-3.1` and `PID-3.5` are used to specify the patient ID as well as alternate patient ID of the referenced patient. To differ between the two IDs, the HL7 component `PID-3.5` contains the value `PI` if `PID-3.1` specifies the patient ID and the value NH, `NI`, or SS if `PID-3.1` specifies an alternate patient ID. Note that the value `NH` indicates that the alternate patient ID represents a unique national identifier, NH states that the alternate patient ID defines a national health service number, and SS denotes that the alternate patient ID identifies a social security number. In case of the patient ID the assigning authority is given in `PID-3.4`, whereas the alternate patient ID assigning authority is given in `PID-3.4.1`. The HL7 components `PID-5.1` to `PID-5.5` define the alphabetic name of the patient indicated by the value A specified by the HL7 component `PID-5.8`. The ideographic and phonetic names of the patient are not stated. The first repetition contains the legal name indicated by a `L` in `PID-5.7`. If the patient has a birth name there will exist a repetition with the value B in `PID-5.7`.

3.5.3 HL7 PV1 Segment

The HL7 PV1 segment of an outgoing HL7 message consists of the following information:

Table 30: HL7 PV1 Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 PV1 Segment ID	`1`	O	
2	Patient Class		O	
19.1	Visit ID		R	
19.4.1	Visit ID Assigning Authority Namespace ID		O	
19.4.2	Visit ID Assigning Authority Universal ID		O	
19.4.3	Visit ID Assigning Authority Universal ID Type		O	
50.1	Alternate Visit ID		O	
50.4	Alternate Visit ID Assigning Authority		O	
51.1	Visit Indicator	`V`	R	

The assigning authority of the visit ID (alternate visit ID) is specified by the HL7 component `PV1-19.4` (`PV1-50.4`). The HL7 component `PV1-51.1` specifies the value V to indicate that the information provided by the HL7 PV1 segment are dedicated to the visit level (note that *syngo*.share does not support visits related to the account level).

3.5.4 HL7 ORC Segment

The HL7 ORC segment of an outgoing HL7 message consists of the following information:

Table 31: HL7 ORC Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	Order Control Code	`OP`	R	
2.1	Placer Order Number		C	
2.2	Placer Order Number Assigning Authority Namespace ID		C	
2.3	Placer Order Number Assigning Authority Universal ID		C	
2.4	Placer Order Number Assigning Authority Universal ID Type		C	
3.1	Filler Order Number		C	
3.2	Filler Order Number Assigning Authority Namespace ID		C	
3.3	Filler Order Number Assigning Authority Universal ID		C	
3.4	Filler Order Number Assigning Authority Universal ID Type		C	
5	Order Status		O	

The HL7 components `ORC-2.1` and `ORC-3.1` define the placer order number and filler order number of the order that is associated with the referenced DICOM Study, generic container, or generic file. Since *syngo*.share needs at least one of the two IDs to store an order, it is guaranteed that at least one of the two HL7 components is specified. In `ORC-2.2` to `ORC-2.4` and `ORC-3.2` to `ORC-3.4` the assigning authorities of the placer order number and the filler order number are specified. The assigning authority information is only present if the corresponding ID exists. Note that the HL7 components `ORC-2.1` to `ORC-2.4` and `ORC-3.1` to `ORC-3.4` specify the same values as the HL7 components `OBR-2.1` to `OBR-2.4` and `OBR-3.1` to `OBR-3.4`.

3.5.5 HL7 OBR Segment

The HL7 OBR segment of an outgoing HL7 message consists of the following information:

Table 32: HL7 OBR Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 OBR Segment ID	1	R	
2.1	Placer Order Number		C	
2.2	Placer Order Number Assigning Authority Namespace ID		C	
2.3	Placer Order Number Assigning Authority Universal ID		C	
2.4	Placer Order Number Assigning Authority Universal ID Type		C	
3.1	Filler Order Number		C	
3.2	Filler Order Number Assigning Authority Namespace ID		C	
3.3	Filler Order Number Assigning Authority Universal ID		C	
3.4	Filler Order Number Assigning Authority Universal ID Type		C	
4.1	Universal Service ID		C	
4.2	Universal Service Description		O	
7	Observation Creation Timestamp		O	
18	Accession Number		R	
19	Requested Procedure ID		C	
22	Observation Insertion Timestamp		R	
24	Practice Setting Code		C	
25	Result Status		R	

The HL7 components OBR-2.1 and OBR-3.1 define the placer order number and filler order number of the order that is associated with the referenced DICOM Study, generic container, or generic file. Since *syngo*.share needs at least one of the two IDs to store an order, it is guaranteed that at least one of the two HL7 components is specified if an appropriate order exists. Otherwise both HL7 components are left empty. Likewise the HL7 components OBR-4.1 and OBR-19 are specified if and only if the referenced DICOM Study, generic container, or generic file is associated with an order. In OBC-2.2 to OBC-2.4 and OBC-3.2 to OBC-3.4 the assigning authorities of the placer order number and the filler order number are specified. The assigning authority information is only present if the corresponding ID exists. If a generic file is routed via a HL7 MDM message the HL7 field OBR-24 defines the practice setting code of the generic file. It is guaranteed that the practice setting code is specified whenever the generic file is registered in an affinity domain. Finally, the HL7 field OBR-25 is used to indicate the status of the referenced DICOM Study, generic container, or generic file. If the object has been deleted a D is specified. Otherwise OBR-25 contains the value F. Note that the HL7 components and HL7 fields OBR-2.1, OBR-3.1, OBR-7, and OBR-25 specify the same values as the HL7 components and HL7 fields ORC-2.1, ORC-3.1, OBX-14, and OBX-11.

3.5.6 HL7 TXA Segment

The HL7 TXA segment of an outgoing HL7 MDM message consists of the following information:

Table 33: HL7 TXA Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 TXA Segment ID	1	R	
2	Generic File Type		C	
3	Generic File Content Presentation		C	
6	Generic File Creation Timestamp		C	
9	Generic File Originator		C	
12.1	Generic File ID		R	
16	Generic Filename		R	
17	Generic File Completion Status		O	
18	Generic File Confidentiality Status		C	
22	Generic File Authenticator		C	
24.1	Generic Container ID		R	
24.2	Generic Container Description		O	
24.3	Generic Container Coding System ID		R	
25	Generic File Description		C	

The information stated by the HL7 fields TXA-2, TXA-3, TXA-9, TXA-18, and TXA-22 define the type, content, originator, confidentiality status, and authenticator of the generic file addressed by the HL7 TXA segment. Note that *syngo*.share uses this information as class code, format code, author, healthcare facility type code, institution, confidentiality code, and authenticator if a generic file is registered in an affinity domain (see Section 2.3 and the IHE IT Infrastructure Technical Framework for further information). For this reason, the mentioned HL7 components and HL7 fields are specified whenever the HL7 TXA segment refers to a generic file which has been registered in an affinity domain. The same rule applies to the HL7 field TXA-6 — it is guaranteed that TXA-6 specifies the creation timestamp of the generic file if it has been registered in an affinity domain. The HL7 component TXA-12.1 defines the ID of the given generic file which should be used together with the application ID, specified by the HL7 component MSH-3.1, to identify it uniquely. The name of the referenced generic file (inclusive its file extension) is specified by the HL7 field TXA-16. The HL7 components TXA-24.1 and TXA-24.3 define the generic container ID and coding system ID of the generic container which contains the generic file. Finally, the HL7 field TXA-25 specifies the description of the generic file. It is present whenever the information of the HL7 TXA segment refers to a generic file which has been registered in an affinity domain. Note that the HL7 field TXA-6 specifies the same value as the HL7 fields OBR-7 and OBX-14.

3.5.7 HL7 OBX Segment

The information provided by a HL7 OBX segment as well as the structure of a HL7 OBX segment depend on the type of the object or observation that is addressed by the HL7 OBX segment. Within *syngo*.share HL7 OBX segments are used in combination with HL7 MDM messages to route generic files or within HL7 ORU messages to send reference pointers linking to DICOM Studies or generic containers to third-party systems. In the former case the HL7 OBX segment of an outgoing HL7 MDM message consists of the following information:

Table 34: HL7 OBX Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 OBX Segment ID	`1`	R	
2	Value Type		R	
3.1	Generic File ID		R	
3.2	Generic File Description		O	
3.3	Generic File System ID		R	
3.4	Generic Container ID		R	
3.5	Generic Container Description		O	
3.6	Generic Container Coding System ID		O	
4	HL7 OBX Segment Sub-ID	`1`	R	
5	Value		R	
8	Generic File Interpretation Code		C	
11	Generic File Result Status	F	R	
14	Generic File Creation Timestamp		O	

The HL7 components `OBX-3.1` and `OBX-3.3` specify the original IDs of the addressed generic file which are used by *syngo*.share to identify it uniquely. It is guaranteed that the combination of both IDs occurs exactly once. The HL7 components `OBX-3.4` and `OBX-3.6` identify the generic container which consists of the referenced generic file. Remember that generic containers are used by *syngo*.share to group generic files. Hence the combination of both IDs need not be unique (see Section 2.3 for further information). To be able to interpret the HL7 field `OBX-5` it is necessary to evaluate the HL7 field `OBX-2`. If `OBX-2` specifies the value ED the binary data of the generic file is encapsulated in the HL7 MDM message. In this particular case the HL7 field `OBX-5` possesses the following structure:

Table 35: HL7 OBX-5 Field (Binary Data), Outbound

Index	Description	Content	Optionality
5.2	Mime Type		R
5.3	Mime Subtype		O
5.4	Encoding	`Base64`	R
5.5	Binary Data		R

Note that *syngo*.share encodes the binary data of a generic file as a Base64 string. Hence the HL7 component `OBX-5.4` always specifies the value `Base64`. If the HL7 field `OBX-2` specifies the value RP the binary data of the generic file is referenced using an UNC-path. To this end `OBX-5` consists of the following information:

Table 36: HL7 OBX-5 Field (UNC-Path), Outbound

Index	Description	Content	Optionality
5.1	UNC-Path		R
5.3	Mime Type		R
5.4	Mime Subtype		O

Finally, the HL7 field OBX-8 defines an interpretation code. Note that *syngo*.share uses the interpretation code of a generic file as content type code if it is registered in an affinity domain (see Section 2.3 and the IHE IT Infrastructure Technical Framework for further information). For this reason, the HL7 field is specified whenever the information of the HL7 OBX segment refers to a generic file which has been registered in an affinity domain. Note that the HL7 field OBX-11 defines the same value as the HL7 field OBR-25 and the HL7 field OBX-14 specifies the same value as the HL7 fields OBR-7 and TXA-6.

If a HL7 OBX segment is used within an outgoing HL7 ORU message which inserts, updates, or deletes a reference pointer of a document (DICOM Study or generic container) it consists of the following information:

Table 37: HL7 OBX Segment (HL7 ORU Message), Outbound

Index	Description	Content	Optionality	Repeatable
1	HL7 OBX Segment ID	1	R	
2	Value Type	RP	R	
3.1	Document Instance UID		R	
3.2	Document Description		O	
3.4	Document ID		R	
4	HL7 OBX Segment Sub-ID	1	R	
5.1	Reference Pointer		R	
5.3	Generic File IDs		O	
5.5	Document Instance Number		R	
8.1	Document Composition		R	
11	Document Result Status		R	
14	Document Creation Timestamp		O	
15.1	Organizational Unit Name		R	
15.2	Organizational Unit Description		R	
15.4	Alternate Organizational Unit Name		O	
15.5	Alternate Organizational Unit Description		O	

The HL7 component OBX-3.1 specifies the UID of the referenced document that is used by *syngo*.share to identify it uniquely. If the document represents a DICOM Study the UID is identical to the Study Instance UID. In case of a generic container, it specifies an internal UID generated by *syngo*.share. The HL7 component OBX-3.4 specifies an alternate ID of the referenced document which is internally used by *syngo*.share to identify the document uniquely. The HL7 subcomponents of the HL7 component OBX-5.3 specify the generic file IDs of the generic files that belong to the referenced document if the document represents a generic container. Otherwise the HL7 component is left empty. Note that the generic file IDs have been generated by the third-party systems which have archived the generic files. The HL7 component OBX-5.5 defines the number of DICOM Images in case of a DICOM Study and the number of generic files otherwise. Next the HL7 component OBX-8.1 is used to define some interpretation codes. In case of a DICOM Study the modalities of the DICOM Series are listed. If the reference pointer is concerned with a generic container the file extensions of all associated generic files are specified. Finally, the HL7 field OBX-11 is used to differ between HL7 ORU messages which insert or update reference pointers and HL7 ORU messages which delete reference pointers. In the former case OBX-11 contains a F whereas in the latter case a D is specified. Note that the HL7 fields OBX-11 and OBX-14 specify the same values as the HL7 fields OBR-25 and OBR-7.

3.5.8 HL7 QRD Segment

The structure of the query description provided in an HL7 QRD segment depends slightly on the chosen trigger event type. In case of a QRY A19 message, the segment of an outgoing message has the following structure:

Table 38: HL7 QRD Segment (HL7 QRY A19 Message), Outbound

Index	Description	Content	Optionality	Repeatable
1	Query Date/Time		R	
2	Query Format Code	D	R	
3	Query Priority	I	R	
4	Query ID		R	
7.1	Quantity Limited Request	1	R	
7.2	Quantity Limited Request Unit	RD	R	
8	Visit ID		O	
9	What Subject Filter	APN	R	

In case of a QRY Q01 message, the structure shows up as follows:

Table 39: HL7 QRD Segment (HL7 QRY Q01 Message), Outbound

Index	Description	Content	Optionality	Repeatable
1	Query Date/Time		R	
2	Query Format Code	R	R	
3	Query Priority	I	R	
4	Query ID		R	
7.1	Quantity Limited Request	30	R	
7.2	Quantity Limited Request Unit	RD	R	
8.2	Visit ID		O	
9	What Subject Filter	APA	R	
10	What Department Data Code	ALL	R	
12	Query Results Level	T	O	

For both structurs, the generation time of the outgoing message is used for the Query Date/Time `QRD-1`. The ID given in `QRD-4` corresponds to the one of `MSH-10`.

3.5.9 HL7 MSA Segment

The HL7 MSA segment of an outgoing HL7 application acknowledgment message consists of the following information:

Table 40: HL7 MSA Segment, Outbound

Index	Description	Content	Optionality	Repeatable
1	Acknowledgment Code		R	
2	HL7 Message Control ID		R	

The HL7 component `MSA-1` defines the status of the processed HL7 message. If the HL7 message has been processed successfully the value AA is stated. If an error occurred the value AR is specified. The HL7 field `MSA-2` defines the control ID of the acknowledged HL7 message.

A Example Workflows

A.1 Ordering of DICOM Studies

One of the major tasks of *syngo*.share is the archiving of DICOM Images. To guarantee the integrity and consistency of the stored DICOM Images *syngo*.share fulfills all requirements of the IHE profiles Scheduled Workflow, Patient Information Reconciliation, and Scheduled Workflow.b. In this context, the HL7 inbound interface is responsible for a correct and continuous actualization and propagation of patient, visit, and order information, as illustrated in Figure 1. To ensure that stored patient and visit information are up to date *syngo*.share receives HL7 ADT messages. In contrast, HL7 OMI or HL7 ORM messages are

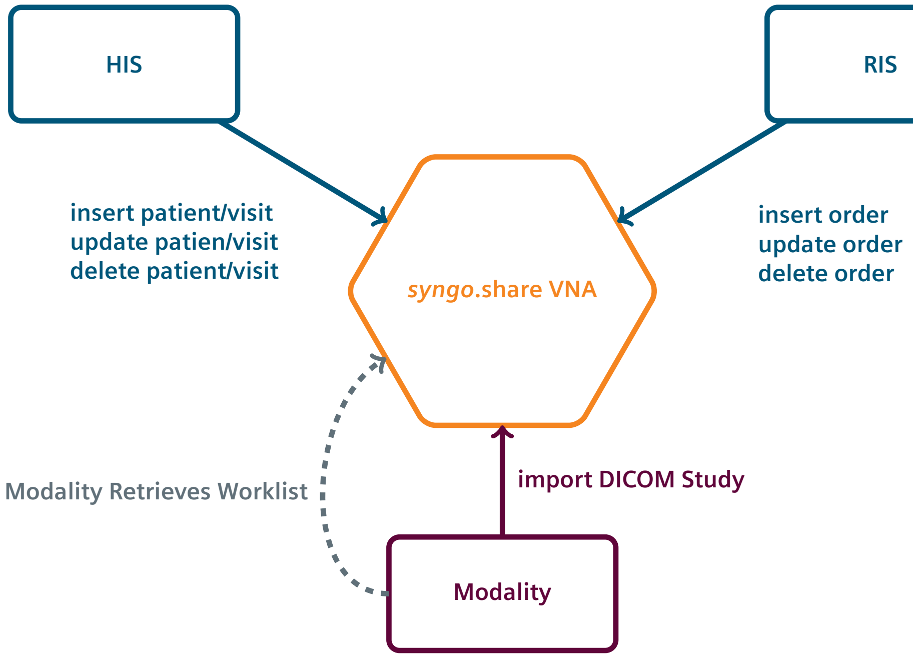

Figure 1: Ordering Workflow

sent to *syngo*.share to ensure that the ordering information of DICOM Images are updated and DICOM Images are associated with the intended patients. Usually, patients and visits are administrated by a HIS whereas orders are managed by a RIS. To allow a fine grained modification of patient, visit and order information, various kinds of HL7 ADT, HL7 OMI, and HL7 ORM messages are supported. In the following these HL7 messages are illustrated with help of a typical patient encounter which reflects the registration of a patient, the performance of an order, and the update and deletion of order, visit, and patient information.

If a patient arrives at a healthcare facility, the personal data of the patient are stored and forwarded to *syngo*.share. To this end a HL7 ADT A04 message, similar to the one shown in Example 1 is sent to *syngo*.share. To medicate the arrived patient it might be necessary to create DICOM Images of a specific part of the patients body. To schedule the creation of the necessary DICOM Images, the HIS instructs the RIS to schedule an appropriate order. The information of the created order are propagated to *syngo*.share by a HL7 OMI O23 or HL7 ORM O01 message as shown in Examples 2 and 3. In between the order is processed by a modality which sends the created DICOM Images to *syngo*.share. In case that order information changes during the creation of the DICOM Images or after the completion of the order, the RIS updates the transmitted order information by sending an HL7 OMI O23 or HL7 ORM O01 message as shown in Examples 4 and 5. To ensure that the order information of the archived DICOM Images (accession number, referring physician, DICOM Study description, and requested procedure ID) are updated by *syngo*.share too, the metadata update has to be activated. During the further medication of the patient it might happen that patient and visit information are completed or corrected. If personal data is modified, the HIS informs *syngo*.share about the conducted changes by sending a HL7 ADT A08 message similar to the one shown in Example 6. If the identification of the patient or visit changes because

1. the patient ID (visit ID) is replaced by another patient ID (visit ID),
2. the visit is moved to another patient, or
3. the patient (visit) is merged with another patient (visit)

the HIS sends

1. a HL7 ADT A47 message as illustrated by Example 7 (a HL7 ADT A50 message as shown in Example 8),
2. a HL7 ADT A45 message as defined in Example 9, or a
3. a HL7 ADT A40 message as specified by Example 10 (a HL7 ADT A42 messages as stated by Example 11).

to *syngo*.share. Note that all those changes also affect the metadata of the DICOM Images which are associated with the manipulated patient and visit. In case that an order, visit, or patient has been created by mistake it can be deleted. In the former case the RIS sends a HL7 OMI O23 or HL7 ORM O01 message to *syngo*.share. HL7 messages of this kind are illustrated in Examples 12 and 13. If a visit or patient is deleted by the HIS, an appropriate HL7 ADT A23 or HL7 ADT A29 message as shown in Examples 14 and 15 is sent to *syngo*.share.

Example 1. The following HL7 ADT A01 message inserts a patient with patient ID `patientId` and a visit with visit ID `visitId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A04^ADT_A01|1|P|2.8.1
   ||||||UNICODE UTF-8
PID|1||patientId^^^assigningAuthority^PI~patientSsn^^^assigningAuthority
   ^SS||lastName^firstName^middleName^suffix^prefix^^^A
   |mothersMaidenLastName^mothersMaidenFirstName^mothersMaidenMiddleName
   ^mothersMaidenSuffix^mothersMaidenPrefix^^^A|20000102030405.1234
   |M||race^raceDescription|patientStreet^^patientCity^patientState^1234
   ^patientCountry|||||M^Married|CHR^Christian|accountNumber^^
   ^assigningAuthority|||||birthPlace|Y|1|A^austrian|profession
   ^professionDescription|||||||||||||^PRN^CP^patientCommunicationAddress
   ^0043^0512^89059^^^^^+4351289059
```

```
NK1|1|nextKinLastName^nextKinFirstName^nextKinMiddleName^nextKinSuffix
   ^nextKinPrefix^^^A|PAR^parent|nextKinStreet^^nextKinCity^nextKinState
   ^1234^nextKinCountry||||||||||||||||||||||||||||||||||||^PRN^CP
   ^nextKinCommunicationAddress^0043^0512^89060^^^^^+4351289060
PV1|1|I^inpatient|^room^bed^facility|||^priorRoom^priorBed^priorFacility
   |attendingDoctor^attendingDoctorLastName^attendingDoctorFirstName
   ^attendingDoctorMiddleName^attendingDoctorSuffix^attendingDoctorPrefix
   ^^^^^^^^^A|referringDoctor^referringDoctorLastName
   ^referringDoctorFirstName^referringDoctorMiddleName
   ^referringDoctorSuffix^referringDoctorPrefix^^^^^^^^^A||MED
   ^medicalService|^temporaryRoom^temporaryBed^temporaryFacility||||A4
   ^disoriented|vipIndicator^vipIndicatorDescription|admittingDoctor
   ^admittingDoctorLastName^admittingDoctorFirstName
   ^admittingDoctorMiddleName^admittingDoctorSuffix^admittingDoctorPrefix
   ^^^^^^^^^A||visitId^^^assigningAuthority|||||||||||||||||||
   |servicingFacility^servicingFacilityDescription||||
   |20000102030405.1234|||||alternateVisitId^^^assigningAuthority|V
ROL||LI|CP|consultingDoctor^consultingDoctorLastName
   ^consultingDoctorFirstName^consultingDoctorMiddleName
   ^consultingDoctorSuffix^consultingDoctorPrefix^^^^^^^^^A
GT1|1||guarantorLastName^guarantorFirstName^guarantorMiddleName
   ^guarantorSuffix^guarantorPrefix^^^A||guarantorStreet^^guarantorCity
   ^guarantorState^1234^guarantorCountry|||19900102030405.1234|||
   |guarantorSsn||||guarantorEmployerLastName^guarantorEmployerFirstName
   ^guarantorEmployerMiddleName^guarantorEmployerSuffix
   ^guarantorEmployerPrefix^^^A|guarantorEmployerStreet^
   ^guarantorEmployerCity^guarantorEmployerState^1234
   ^guarantorEmployerCountry
```

Example 2. The following HL7 OMI O23 message inserts an order with placer order number `placerOrderNumber` and filler order number `fillerOrderNumber`. It is scheduled for a patient with patient ID `patientId` during a visit with visit ID `visitId` and consists of a requested procedure with Study Instance UID `1.2.3.4.5.6.7.8.9` and requested procedure ID `requestedProcedureId`. The requested procedure in turn consists of a single scheduled procedure step with scheduled procedure step ID `scheduledProcedureStepId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||OMI^O23^OMI_O23|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
NK1|...
PV1|...
ROL|...
GT1|...
ORC|NW|placerOrderNumber|fillerOrderNumber||SC|||||enterer
   ^entererLastName^entererFirstName^entererMiddleName^entererSuffix
   ^entererPrefix^^^^^^^^^A||provider^providerLastName^providerFirstName
   ^providerMiddleName^providerSuffix^providerPrefix^^^^^^^^^A|
   ^^^facility|^PRN^CP^callBackCommunicationAddress^0043^0512^89059^^^^
   ^+4351289059||controlCodeReasonId^controlCodeReasonDescription
   |enteringOrganizationId^enteringOrganizationDescription|||
   |facilityName^facilityNameTypeCode
TQ1|1|quantity||||duration|20000102030405.1234||priority
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription||||||||dangerCodeId^dangerCodeDescription
   |relevantClinicalInformationId^relevantClinicalInformationDescription|
   |specimenSourceName^specimenSourceDescription|provider
   ^providerLastName^providerFirstName^providerMiddleName^providerSuffix
   ^providerPrefix^^^^^^^^^A|^PRN^CP^callBackCommunicationAddress^0043
   ^0512^89059^^^^^+4351289059|||||||||||||transportationMode
   |reasonForStudyId^reasonForStudyDescription|||technician
   &technicianLastName&technicianFirstName&technicianMiddleName
   &technicianSuffix&technicianPrefix||||||||||procedureCodeId
   ^procedureCodeDescription^^^requestedProcedureDescription
IPC|accessionNumber|requestedProcedureId|1.2.3.4.5.6.7.8.9
```

```
   |scheduledProcedureStepId|CT|protocolCode^protocolDescription^^
   ^scheduledProcedureStepDescription|||aet
```

Note that the HL7 PID, HL7 NK1, HL7 PV1, HL7 ROL, and HL7 GT1 segments are identical to the ones defined in Example 1.

Example 3. The following HL7 ORM O01 message inserts an order with placer order number `placerOrderNumber` and filler order number `fillerOrderNumber`. It is scheduled for a patient with patient ID `patientId` during a visit with visit ID `visitId` and consists of a requested procedure with Study Instance UID `1.2.3.4.5.6.7.8.9` and requested procedure ID `requestedProcedureId`. The requested procedure in turn consists of a single scheduled procedure step with scheduled procedure step ID `scheduledProcedureStepId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ORM^O01^ORM_O01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
NK1|...
PV1|...
ROL|...
GT1|...
ORC|NW|placerOrderNumber|fillerOrderNumber||SC||quantity^interval
   ^duration^20000102030405.1234^^priority|||enterer^entererLastName
   ^entererFirstName^entererMiddleName^entererSuffix^entererPrefix^^^^^^^
   ^^A||provider^providerLastName^providerFirstName^providerMiddleName
   ^providerSuffix^providerPrefix^^^^^^^^^A|^^^facility|^PRN^CP
   ^callBackCommunicationAddress^0043^0512^89059^^^^^+4351289059|
   |controlCodeReasonId^controlCodeReasonDescription
   |enteringOrganizationId^enteringOrganizationDescription|||
   |facilityName^facilityNameTypeCode
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription^^protocolCode^protocolDescription^^^^^
   ^scheduledProcedureStepDescription||||||||dangerCodeId
   ^dangerCodeDescription|relevantClinicalInformationId
   ^relevantClinicalInformationDescription||specimenSourceName
   ^specimenSourceDescription|provider^providerLastName^providerFirstName
   ^providerMiddleName^providerSuffix^providerPrefix^^^^^^^^^A|^PRN^CP
   ^callBackCommunicationAddress^0043^0512^89059^^^^^+4351289059
   |accessionNumber|requestedProcedureId|scheduledProcedureStepId|aet||
   |CT|||quantity^interval^duration^20000102030405.1234^^priority||
   |transportationMode|reasonForStudyId^reasonForStudyDescription||
   |technician&technicianLastName&technicianFirstName
   &technicianMiddleName&technicianSuffix&technicianPrefix|||||||||
   |procedureCodeId^procedureCodeDescription^^
   ^requestedProcedureDescription
ZDS|1.2.3.4.5.6.7.8.9
```

Note that the HL7 PID, HL7 NK1, HL7 PV1, HL7 ROL, and HL7 GT1 segments are identical to the ones defined in Example 1.

Example 4. The following HL7 OMI O23 message updates an order with placer order number `placerOrderNumber` and filler order number `fillerOrderNumber`, a requested procedure with Study Instance UID `1.2.3.4.5.6.7.8.9`, and a scheduled procedure step with scheduled procedure step ID `scheduledProcedureStepId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||OMI^O23^OMI_O23|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
NK1|...
PV1|...
ROL|...
GT1|...
ORC|XO|placerOrderNumber|fillerOrderNumber||CM|||||updatedEnterer
   ^updatedEntererLastName^updatedEntererFirstName
   ^updatedEntererMiddleName^updatedEntererSuffix^updatedEntererPrefix^^^
   ^^^^^^A||updatedProvider^updatedProviderLastName
   ^updatedProviderFirstName^updatedProviderMiddleName
   ^updatedProviderSuffix^updatedProviderPrefix^^^^^^^^^A|^^
   ^updatedFacility|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512
   ^89059^^^^^+4351289059||updatedControlCodeReasonId
   ^updatedControlCodeReasonDescription|updatedEnteringOrganizationId
   ^updatedEnteringOrganizationDescription||||updatedFacilityName
   ^updatedFacilityNameTypeCode
TQ1|1|updatedQuantity||||updatedDuration|20000102030405.1234|
   |updatedPriority
OBR|1|placerOrderNumber|fillerOrderNumber|updatedUniversalServiceId
   ^updatedUniversalServiceDescription|||||||updatedDangerCodeId
   ^updatedDangerCodeDescription|updatedRelevantClinicalInformationId
   ^updatedRelevantClinicalInformationDescription|
   |updatedSpecimenSourceName^updatedSpecimenSourceDescription
   |updatedProvider^updatedProviderLastName^updatedProviderFirstName
   ^updatedProviderMiddleName^updatedProviderSuffix^updatedProviderPrefix
   ^^^^^^^^^A|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512^89059
   ^^^^^+4351289059||||||||||||updatedTransportationMode
   |updatedReasonForStudyId^updatedReasonForStudyDescription||
   |updatedTechnician&updatedTechnicianLastName
   &updatedTechnicianFirstName&updatedTechnicianMiddleName
   &updatedTechnicianSuffix&updatedTechnicianPrefix||||||||
   |updatedProcedureCodeId^updatedProcedureCodeDescription^^
   ^updatedRequestedProcedureDescription
IPC|updatedAccessionNumber|updatedRequestedProcedureId|1.2.3.4.5.6.7.8.9
   |scheduledProcedureStepId|MR|updatedProtocolCode
   ^updatedProtocolDescription^^^updatedScheduledProcedureStepDescription
   |||updatedAet
```

Note that the HL7 PID, HL7 NK1, HL7 PV1, HL7 ROL, and HL7 GT1 segments are identical to the ones defined in Example 1.

Example 5. The following HL7 ORM O01 message updates an order with placer order number `placerOrderNumber` and filler order number `fillerOrderNumber`, a requested procedure with Study Instance UID `1.2.3.4.5.6.7.8.9`, and a scheduled procedure step with scheduled procedure step ID `scheduledProcedureStepId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ORM^O01^ORM_O01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
NK1|...
PV1|...
ROL|...
GT1|...
ORC|XO|placerOrderNumber|fillerOrderNumber||CM||updatedQuantity
   ^updatedInterval^updatedDuration^20000102030405.1234^^updatedPriority
   |||updatedEnterer^updatedEntererLastName^updatedEntererFirstName
   ^updatedEntererMiddleName^updatedEntererSuffix^updatedEntererPrefix^^^
   ^^^^^^A||updatedProvider^updatedProviderLastName
   ^updatedProviderFirstName^updatedProviderMiddleName
   ^updatedProviderSuffix^updatedProviderPrefix^^^^^^^^^A|^^
   ^updatedFacility|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512
   ^89059^^^^^+4351289059||updatedControlCodeReasonId
   ^updatedControlCodeReasonDescription|updatedEnteringOrganizationId
   ^updatedEnteringOrganizationDescription||||updatedFacilityName
   ^updatedFacilityNameTypeCode
OBR|1|placerOrderNumber|fillerOrderNumber|updatedUniversalServiceId
   ^updatedUniversalServiceDescription^^updatedProtocolCode
   ^updatedProtocolDescription^^^^^
   ^updatedScheduledProcedureStepDescription|||||||updatedDangerCodeId
   ^updatedDangerCodeDescription|updatedRelevantClinicalInformationId
   ^updatedRelevantClinicalInformationDescription|
   |updatedSpecimenSourceName^updatedSpecimenSourceDescription
   |updatedProvider^updatedProviderLastName^updatedProviderFirstName
   ^updatedProviderMiddleName^updatedProviderSuffix^updatedProviderPrefix
   ^^^^^^^^^A|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512^89059
   ^^^^^+4351289059|updatedAccessionNumber|updatedRequestedProcedureId
   |scheduledProcedureStepId|updatedAet|||MR|||updatedQuantity
   ^updatedInterval^updatedDuration^20000102030405.1234^^updatedPriority
   |||updatedTransportationMode|updatedReasonForStudyId
   ^updatedReasonForStudyDescription|||updatedTechnician
   &updatedTechnicianLastName&updatedTechnicianFirstName
   &updatedTechnicianMiddleName&updatedTechnicianSuffix
   &updatedTechnicianPrefix||||||||||updatedProcedureCodeId
   ^updatedProcedureCodeDescription^^
   ^updatedRequestedProcedureDescription
ZDS|1.2.3.4.5.6.7.8.9
```

Note that the HL7 PID, HL7 NK1, HL7 PV1, HL7 ROL, and HL7 GT1 segments are identical to the ones defined in Example 1.

Example 6. The following HL7 ADT A08 message updates a patient with patient ID `patientId` and a visit with visit ID `visitId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A08^ADT_A01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|1||patientId^^^assigningAuthority^PI~updatedPatientSsn^^
   ^updatedAssigningAuthority^SS||updatedLastName^updatedFirstName
   ^updatedMiddleName^updatedSuffix^updatedPrefix^^^A
   |updatedMothersMaidenLastName^updatedMothersMaidenFirstName
   ^updatedMothersMaidenMiddleName^updatedMothersMaidenSuffix
   ^updatedMothersMaidenPrefix^^^A|20000102030405.1234|M||updatedRace
   ^updatedRaceDescription|updatedPatientStreet^^updatedPatientCity
   ^updatedPatientState^1234^updatedPatientCountry|||||M^Married|CHR
   ^Christian|updatedAccountNumber^^^updatedAssigningAuthority||||
   |updatedBirthPlace|Y|1|A^austrian|updatedProfession
   ^updatedProfessionDescription|||||||||||||^PRN^CP
   ^updatedPatientCommunicationAddress^0043^0512^89059^^^^^+4351289059
NK1|1|updatedNextKinLastName^updatedNextKinFirstName
   ^updatedNextKinMiddleName^updatedNextKinSuffix^updatedNextKinPrefix^^
   ^A|PAR^parent|updatedNextKinStreet^^updatedNextKinCity
   ^updatedNextKinState^1234^updatedNextKinCountry|||||||||||||||||||||||
   |||||||||||||^PRN^CP^updatedNextKinCommunicationAddress^0043^0512
   ^89060^^^^^+4351289060
PV1|1|I^inpatient|^updatedRoom^updatedBed^updatedFacility|||
   ^updatedPriorRoom^updatedPriorBed^updatedPriorFacility
   |updatedAttendingDoctor^updatedAttendingDoctorLastName
   ^updatedAttendingDoctorFirstName^updatedAttendingDoctorMiddleName
   ^updatedAttendingDoctorSuffix^updatedAttendingDoctorPrefix^^^^^^^^^A
   |updatedReferringDoctor^updatedReferringDoctorLastName
   ^updatedReferringDoctorFirstName^updatedReferringDoctorMiddleName
   ^updatedReferringDoctorSuffix^updatedReferringDoctorPrefix^^^^^^^^^A|
   |MED^medicalService|^updatedTemporaryRoom^updatedTemporaryBed
   ^updatedTemporaryFacility||||A4^disoriented|updatedVipIndicator
   ^updatedVipIndicatorDescription|updatedAdmittingDoctor
   ^updatedAdmittingDoctorLastName^updatedAdmittingDoctorFirstName
   ^updatedAdmittingDoctorMiddleName^updatedAdmittingDoctorSuffix
   ^updatedAdmittingDoctorPrefix^^^^^^^^^A||visitId^^^assigningAuthority
   ||||||||||||||||||||updatedServicingFacility
   ^updatedServicingFacilityDescription|||||20000102030405.1234|||||
   |updatedAlternateVisitId^^^updatedAssigningAuthority|V
ROL||LI|CP|updatedConsultingDoctor^updatedConsultingDoctorLastName
   ^updatedConsultingDoctorFirstName^updatedConsultingDoctorMiddleName
   ^updatedConsultingDoctorSuffix^updatedConsultingDoctorPrefix^^^^^^^^^A
GT1|1||updatedGuarantorLastName^updatedGuarantorFirstName
   ^updatedGuarantorMiddleName^updatedGuarantorSuffix
   ^updatedGuarantorPrefix^^^A||updatedGuarantorStreet^
   ^updatedGuarantorCity^updatedGuarantorState^1234
   ^updatedGuarantorCountry|||19900102030405.1234||||updatedGuarantorSsn|
   |||updatedGuarantorEmployerLastName^updatedGuarantorEmployerFirstName
   ^updatedGuarantorEmployerMiddleName^updatedGuarantorEmployerSuffix
   ^updatedGuarantorEmployerPrefix^^^A|updatedGuarantorEmployerStreet^
   ^updatedGuarantorEmployerCity^updatedGuarantorEmployerState^1234
   ^updatedGuarantorEmployerCountry
```

Example 7. The following HL7 ADT A47 message updates the patient ID of a patient. The prior patient ID `patientId` is defined by the HL7 MRG segment whereas the new patient ID `updatedPatientId` is defined by the HL7 PID segment.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A47^ADT_A44|1|P|2.8.1||||
   |UNICODE UTF-8
PID|1||updatedPatientId^^^updatedAssigningAuthority^PI~updatedPatientSsn^
   ^^updatedAssigningAuthority^SS||updatedLastName^updatedFirstName
   ^updatedMiddleName^updatedSuffix^updatedPrefix^^^A
   |updatedMothersMaidenLastName^updatedMothersMaidenFirstName
   ^updatedMothersMaidenMiddleName^updatedMothersMaidenSuffix
   ^updatedMothersMaidenPrefix^^^A|20000102030405.1234|M||updatedRace
   ^updatedRaceDescription|updatedPatientStreet^^updatedPatientCity
   ^updatedPatientState^1234^updatedPatientCountry|||||M^Married|CHR
   ^Christian|updatedAccountNumber^^^updatedAssigningAuthority||||
   |updatedBirthPlace|Y|1|A^austrian|updatedProfession
   ^updatedProfessionDescription|||||||||||||^PRN^CP
   ^updatedPatientCommunicationAddress^0043^0512^89059^^^^^+4351289059
MRG|patientId^^^assigningAuthority^PI
NK1|...
GT1|...
```

Note that the HL7 NK1 and HL7 GT1 segments are identical to the ones defined in Example 6.

Example 8. The following HL7 ADT A50 message updates the visit ID of a visit. The prior visit ID `visitId` is defined by the HL7 MRG segment whereas the new visit ID `updatedVisitId` is defined by the HL7 PV1 segment.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A50^ADT_A50|1|P|2.8.1||||
   |UNICODE UTF-8
PID|...
MRG|||||visitId^^^assigningAuthority
NK1|...
PV1|1|I^inpatient|^updatedRoom^updatedBed^updatedFacility|||
   ^updatedPriorRoom^updatedPriorBed^updatedPriorFacility
   |updatedAttendingDoctor^updatedAttendingDoctorLastName
   ^updatedAttendingDoctorFirstName^updatedAttendingDoctorMiddleName
   ^updatedAttendingDoctorSuffix^updatedAttendingDoctorPrefix^^^^^^^^^A
   |updatedReferringDoctor^updatedReferringDoctorLastName
   ^updatedReferringDoctorFirstName^updatedReferringDoctorMiddleName
   ^updatedReferringDoctorSuffix^updatedReferringDoctorPrefix^^^^^^^^^A|
   |MED^medicalService|^updatedTemporaryRoom^updatedTemporaryBed
   ^updatedTemporaryFacility||||A4^disoriented|updatedVipIndicator
   ^updatedVipIndicatorDescription|updatedAdmittingDoctor
   ^updatedAdmittingDoctorLastName^updatedAdmittingDoctorFirstName
   ^updatedAdmittingDoctorMiddleName^updatedAdmittingDoctorSuffix
   ^updatedAdmittingDoctorPrefix^^^^^^^^^A||updatedVisitId^^
   ^updatedAssigningAuthority||||||||||||||||||||updatedServicingFacility
   ^updatedServicingFacilityDescription|||||20000102030405.1234|||||
   |updatedAlternateVisitId^^^updatedAssigningAuthority|V
ROL||LI|CP|updatedConsultingDoctor^updatedConsultingDoctorLastName
   ^updatedConsultingDoctorFirstName^updatedConsultingDoctorMiddleName
   ^updatedConsultingDoctorSuffix^updatedConsultingDoctorPrefix^^^^^^^^^A
GT1|...
```

Note that the HL7 PID, HL7 NK1, and HL7 GT1 segments are identical to the ones defined in Example 7.

Example 9. The following HL7 ADT A45 moves a visit with visit ID `updatedVisitID` to a patient with patient ID `differentPatientId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A45^ADT_A45|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|1||differentPatientId^^^differentAssigningAuthority^PI
   ~differentPatientSsn^^^differentAssigningAuthority^SS|
   |differentLastName^differentFirstName^differentMiddleName
   ^differentSuffix^differentPrefix^^^A|differentMothersMaidenLastName
   ^differentMothersMaidenFirstName^differentMothersMaidenMiddleName
   ^differentMothersMaidenSuffix^differentMothersMaidenPrefix^^^A
   |20000102030405.1234|M||differentRace^differentRaceDescription
   |differentPatientStreet^^differentPatientCity^differentPatientState
   ^1234^differentPatientCountry|||||M^Married|CHR^Christian
   |differentAccountNumber^^^differentAssigningAuthority||||
   |differentBirthPlace|Y|1|A^austrian|differentProfession
   ^differentProfessionDescription|||||||||||||^PRN^CP
   ^differentPatientCommunicationAddress^0043^0512^89059^^^^^+4351289059
MRG|||||updatedVisitId^^^updatedAssigningAuthority
NK1|1|differentNextKinLastName^differentNextKinFirstName
   ^differentNextKinMiddleName^differentNextKinSuffix
   ^differentNextKinPrefix^^^A|PAR^parent|differentNextKinStreet^
   ^differentNextKinCity^differentNextKinState^1234
   ^differentNextKinCountry||||||||||||||||||||||||||||||||||||^PRN^CP
   ^differentNextKinCommunicationAddress^0043^0512^89060^^^^^+4351289060
GT1|1||differentGuarantorLastName^differentGuarantorFirstName
   ^differentGuarantorMiddleName^differentGuarantorSuffix
   ^differentGuarantorPrefix^^^A||differentGuarantorStreet^
   ^differentGuarantorCity^differentGuarantorState^1234
   ^differentGuarantorCountry|||19900102030405.1234|||
   |differentGuarantorSsn||||differentGuarantorEmployerLastName
   ^differentGuarantorEmployerFirstName
   ^differentGuarantorEmployerMiddleName^differentGuarantorEmployerSuffix
   ^differentGuarantorEmployerPrefix^^^A|differentGuarantorEmployerStreet
   ^^differentGuarantorEmployerCity^differentGuarantorEmployerState^1234
   ^differentGuarantorEmployerCountry
```

Note that the moving of the visit also entails the moving of the documents that are associated with the visit.

Example 10. The following HL7 ADT A40 message merges two patients. The patient ID `updatedPatientId` of the invalidated patient is defined by the HL7 MRG segment whereas the patient ID `differentPatientId` of the remaining patient is defined by the HL7 PID segment.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A40^ADT_A39|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
MRG|updatedPatientId^^^updatedAssigningAuthority^PI
NK1|...
GT1|...
```

Note that the HL7 PID, HL7 NK1, and HL7 GT1 segments are identical to the ones defined in Example 9.

Example 11. The following HL7 ADT A42 message merges two visits. The visit ID `updatedVisitId` of the invalidated visit is defined by the HL7 MRG segment whereas the visit ID `differentVisitId` of the remaining visit is defined by the HL7 PV1 segment.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A42^ADT_A39|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
MRG|||||updatedVisitId^^^updatedAssigningAuthority
NK1|...
PV1|1|I^inpatient|^differentRoom^differentBed^differentFacility|||
   ^differentPriorRoom^differentPriorBed^differentPriorFacility
   |differentAttendingDoctor^differentAttendingDoctorLastName
   ^differentAttendingDoctorFirstName^differentAttendingDoctorMiddleName
   ^differentAttendingDoctorSuffix^differentAttendingDoctorPrefix^^^^^^^^
   ^A|differentReferringDoctor^differentReferringDoctorLastName
   ^differentReferringDoctorFirstName^differentReferringDoctorMiddleName
   ^differentReferringDoctorSuffix^differentReferringDoctorPrefix^^^^^^^^
   ^A||MED^medicalService|^differentTemporaryRoom^differentTemporaryBed
   ^differentTemporaryFacility||||A4^disoriented|differentVipIndicator
   ^differentVipIndicatorDescription|differentAdmittingDoctor
   ^differentAdmittingDoctorLastName^differentAdmittingDoctorFirstName
   ^differentAdmittingDoctorMiddleName^differentAdmittingDoctorSuffix
   ^differentAdmittingDoctorPrefix^^^^^^^^^A||differentVisitId^^
   ^differentAssigningAuthority|||||||||||||||||||
   |differentServicingFacility^differentServicingFacilityDescription||||
   |20000102030405.1234||||||differentAlternateVisitId^^
   ^differentAssigningAuthority|V
ROL||LI|CP|differentConsultingDoctor^differentConsultingDoctorLastName
   ^differentConsultingDoctorFirstName
   ^differentConsultingDoctorMiddleName^differentConsultingDoctorSuffix
   ^differentConsultingDoctorPrefix^^^^^^^^^A
GT1|...
```

Note that the HL7 PID, HL7 NK1, and HL7 GT1 segments are identical to the ones defined in Example 9.

Example 12. The following HL7 OMI O23 message deletes an order with placer order number `placerOrderNumber` and filler order number `fillerOrderNumber`, a requested procedure with Study Instance UID `1.2.3.4.5.6.7.8.9`, and a scheduled procedure step with scheduled procedure step ID `scheduledProcedureStepId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||OMI^O23^OMI_O23|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
NK1|...
PV1|...
ROL|...
GT1|...
ORC|CA|placerOrderNumber|fillerOrderNumber||CA|||||updatedEnterer
   ^updatedEntererLastName^updatedEntererFirstName
   ^updatedEntererMiddleName^updatedEntererSuffix^updatedEntererPrefix^^^
   ^^^^^^A||updatedProvider^updatedProviderLastName
   ^updatedProviderFirstName^updatedProviderMiddleName
   ^updatedProviderSuffix^updatedProviderPrefix^^^^^^^^^A|^^
   ^updatedFacility|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512
   ^89059^^^^^+4351289059||updatedControlCodeReasonId
   ^updatedControlCodeReasonDescription|updatedEnteringOrganizationId
   ^updatedEnteringOrganizationDescription||||updatedFacilityName
   ^updatedFacilityNameTypeCode
TQ1|1|updatedQuantity||||updatedDuration|20000102030405.1234|
   |updatedPriority
OBR|1|placerOrderNumber|fillerOrderNumber|updatedUniversalServiceId
   ^updatedUniversalServiceDescription||||||||updatedDangerCodeId
   ^updatedDangerCodeDescription|updatedRelevantClinicalInformationId
   ^updatedRelevantClinicalInformationDescription|
   |updatedSpecimenSourceName^updatedSpecimenSourceDescription
   |updatedProvider^updatedProviderLastName^updatedProviderFirstName
   ^updatedProviderMiddleName^updatedProviderSuffix^updatedProviderPrefix
   ^^^^^^^^^A|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512^89059
   ^^^^^+4351289059|||||||||||||updatedTransportationMode
   |updatedReasonForStudyId^updatedReasonForStudyDescription||
   |updatedTechnician&updatedTechnicianLastName
   &updatedTechnicianFirstName&updatedTechnicianMiddleName
   &updatedTechnicianSuffix&updatedTechnicianPrefix|||||||||
   |updatedProcedureCodeId^updatedProcedureCodeDescription^^
   ^updatedRequestedProcedureDescription
```

```
IPC|updatedAccessionNumber|updatedRequestedProcedureId|1.2.3.4.5.6.7.8.9
   |scheduledProcedureStepId|MR|updatedProtocolCode
   ^updatedProtocolDescription^^^updatedScheduledProcedureStepDescription
   |||updatedAet
```

Note that the HL7 PID, HL7 NK1, HL7 PV1, HL7 ROL, and HL7 GT1 segments are identical to the ones defined in Example 11.

Example 13. The following HL7 ORM O01 message deletes an order with placer order number `placerOrderNumber` and filler order number `fillerOrderNumber`, a requested procedure with Study Instance UID `1.2.3.4.5.6.7.8.9`, and a scheduled procedure step with scheduled procedure step ID `scheduledProcedureStepId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ORM^O01^ORM_O01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
NK1|...
PV1|...
ROL|...
GT1|...
ORC|CA|placerOrderNumber|fillerOrderNumber||CA||updatedQuantity
   ^updatedInterval^updatedDuration^20000102030405.1234^^updatedPriority
   |||updatedEnterer^updatedEntererLastName^updatedEntererFirstName
   ^updatedEntererMiddleName^updatedEntererSuffix^updatedEntererPrefix^^^
   ^^^^^^A||updatedProvider^updatedProviderLastName
   ^updatedProviderFirstName^updatedProviderMiddleName
   ^updatedProviderSuffix^updatedProviderPrefix^^^^^^^^^A|^^
   ^updatedFacility|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512
   ^89059^^^^^+4351289059||updatedControlCodeReasonId
   ^updatedControlCodeReasonDescription|updatedEnteringOrganizationId
   ^updatedEnteringOrganizationDescription||||updatedFacilityName
   ^updatedFacilityNameTypeCode
OBR|1|placerOrderNumber|fillerOrderNumber|updatedUniversalServiceId
   ^updatedUniversalServiceDescription^^updatedProtocolCode
   ^updatedProtocolDescription^^^^^
   ^updatedScheduledProcedureStepDescription||||||||updatedDangerCodeId
   ^updatedDangerCodeDescription|updatedRelevantClinicalInformationId
   ^updatedRelevantClinicalInformationDescription|
   |updatedSpecimenSourceName^updatedSpecimenSourceDescription
   |updatedProvider^updatedProviderLastName^updatedProviderFirstName
   ^updatedProviderMiddleName^updatedProviderSuffix^updatedProviderPrefix
   ^^^^^^^^^A|^PRN^CP^updatedCallBackCommunicationAddress^0043^0512^89059
   ^^^^^+4351289059|updatedAccessionNumber|updatedRequestedProcedureId
   |scheduledProcedureStepId|updatedAet|||MR|||updatedQuantity
   ^updatedInterval^updatedDuration^20000102030405.1234^^updatedPriority
   |||updatedTransportationMode|updatedReasonForStudyId
   ^updatedReasonForStudyDescription|||updatedTechnician
   &updatedTechnicianLastName&updatedTechnicianFirstName
   &updatedTechnicianMiddleName&updatedTechnicianSuffix
   &updatedTechnicianPrefix||||||||||updatedProcedureCodeId
   ^updatedProcedureCodeDescription^^
   ^updatedRequestedProcedureDescription
ZDS|1.2.3.4.5.6.7.8.9
```

Note that the HL7 PID, HL7 NK1, HL7 PV1, HL7 ROL, and HL7 GT1 segments are identical to the ones defined in Example 11.

Example 14. The following HL7 ADT A23 message deletes a visit with visit ID `differentVisitId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A23^ADT_A21|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
PV1|...
```

Note that the deletion of the visit also entails the deletion of the documents that are associated with the visit. Furthermore the HL7 PID and HL7 PV1 segments are identical to the ones defined in Example 11.
Example 15. The following HL7 ADT A29 message deletes a patient with patient ID `differentPatientId`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ADT^A29^ADT_A21|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
```

Note that the deletion of the patient also entails the deletion of all visits and documents that are associated with the patient. Furthermore the HL7 PID and HL7 PV1 segments are identical to the ones defined in Example 11.

A.2 Archiving of Generic Files

The HL7 interface is one of the most frequently used interfaces to deal with generic files. To insert, update, and delete generic files various kinds of HL7 MDM and HL7 ORU messages are supported. If generic files are routed to third-party systems, *syngo*.share guarantees that updates are propagated by sending adequate HL7 MDM messages. The archiving workflow of generic files is illustrated in Figure 1. In the following the HL7 MDM and HL7 ORU messages used by third-party systems to insert, update, and delete generic files as well as the HL7 MDM messages that are used by *syngo*.share to route and synchronize generic files are illustrated with help of a typical patient encounter which reflects the creation, update, routing, and deletion of a report.

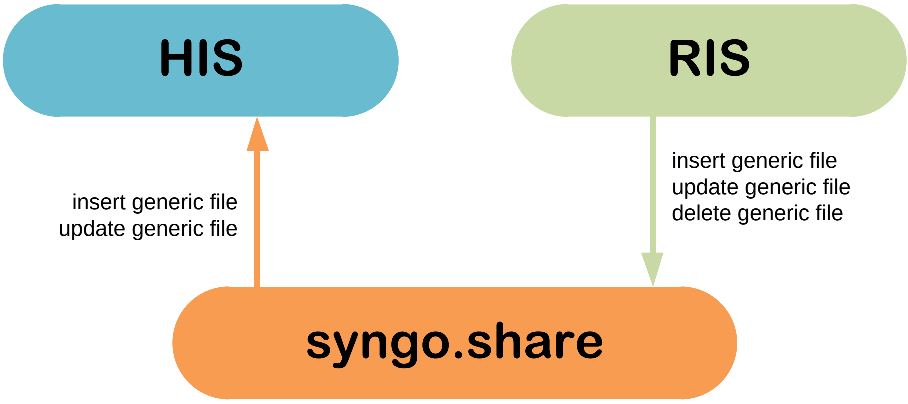

Figure 2: Generic Archiving Workflow

Assume that a patient has arrived at a healthcare facility, the personal data of the patient has been stored and forwarded to *syngo*.share, and an order has been scheduled by the RIS, propagated to *syngo*.share, and performed by a modality in order to be able to proceed with the medication of the arrived patient. At next the DICOM Study is analyzed by a radiologist who creates a report which is send by the RIS via a HL7 MDM T02 or HL7 ORU R01 message (see Examples 16 and 17) to *syngo*.share to ensure that it is archived. As soon as the report has been stored it is forwarded by *syngo*.share to the HIS via a HL7 MDM T02 message similar to the one of Example 18. In between it might happen that the radiologist observes and corrects some mistakes in the report. To synchronize the changes the RIS updates the previously transmitted generic file by sending a HL7 MDM T09, HL7 MDM T10, or HL7 ORU R01 message as illustrated by Examples 19, 20, 21, and 22. Once the update has been conducted, *syngo*.share informs the HIS about the made changes by sending a further HL7 MDM T02 message like the one of Example 23. Finally, to delete a generic file after a certain amount of time, the RIS sends a specific HL7 MDM T11 or HL7 ORU R01 message like the ones shown in Examples 24 and 25 to *syngo*.share.

Example 16. The following HL7 MDM T02 message inserts a generic file with generic file ID `genericFileId` and application ID `sendingApplication`. The generic file is stored in a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. It has been created during a visit with visit ID `visitId` of a patient with patient ID `patientId` and belongs to the organizational unit `organizationalUnit`. The data of the generic file is embedded as a Base64 encoded string in the HL7 MDM T02 message.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||MDM^T02^MDM_T02|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|1||patientId^^^assigningAuthority^PI||lastName^firstName^middleName
   ^suffix^prefix^^^A||20000102030405.1234|M
PV1|1||||||||||||||||||visitId^^^assigningAuthority||||||||||||||||||||||
   ||||||||||V
OBR|1||||||20000102030405.1234|||||||||||accessionNumber|||||
   |diagnosticServiceSectionId^diagnosticServiceSectionDisplayName
   ^diagnosticServiceSectionDescritpion
TXA|1|typeId^typeDisplayName^typeCodingScheme|contentPresentationId
   ^contentPresentationDisplayName^contentPresentationCodingScheme||
   |20000102030405.1234|||originator^originatorLastName
   ^originatorFirstName^originatorMiddleName^originatorSuffix
   ^originatorPrefix^^^^^^^^facilityTypeCodeId
   &facilityTypeCodeDisplayName&facilityTypeCodeCodingScheme^A||
   |genericFileId||||genericFilename.txt|completionStatus
   |confidentialityStatusId^confidentialityStatusDisplayName
   ^confidentialityStatusCodingScheme||||authenticator
   ^authenticatorLastName^authenticatorFirstName^authenticatorMiddleName
   ^authenticatorSuffix^authenticatorPrefix^^^^^^^^^^A|
   |genericContainerId^genericContainerDescription
   ^genericContainerCodingSystemId|genericFileDescription
OBX|1|ED||1|^text^plain^Base64^RXhhbXBsZQo=|||interpretationCodeId
   ^interpretationCodeDescription||||||20000102030405.1234
```

Since HL7 messages are not designed to transport data of big generic files, it is also possible to equipped a HL7 MDM T02 message with an UNC-path which points to the generic file that should be archived. In this particular case the HL7 MDM T02 message specifies the following information:

```
MSH|...
PID|...
PV1|...
OBR|...
TXA|...
OBX|1|RP||1|\E\\E\host\E\share\E\path^^application^pdf||
   |interpretationCodeId^interpretationCodeDescription|||||
   |20000102030405.1234
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, HL7 OBR, and HL7 TXA segments are identical to the ones of the previously defined HL7 MDM T02 message.

Example 17. The following HL7 ORU R01 message inserts a generic file with generic file ID `genericFileId` and application ID `sendingApplication`. The generic file is stored in a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. It has been created during a visit with visit ID `visitId` of a patient with patient ID `patientId` and belongs to the organizational unit `organizationalUnit`. The data of the generic file is embedded as a Base64 encoded string in the HL7 ORU R01 message.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ORU^R01^ORU_R01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
PV1|...
```

```
OBR|1||||||20000102030405.1234||||F|||||||accessionNumber|||||
   |diagnosticServiceSectionId^diagnosticServiceSectionDisplayName
   ^diagnosticServiceSectionDescritpion
OBX|1|ED|genericFileId^genericFileDescription^^genericContainerId
   ^genericContainerDescription^genericContainerCodingSystemId^^
   ^genericFilename.txt|1|^text^plain^Base64^RXhhbXBsZQo=||
   |interpretationCodeId^interpretationCodeDescription|||F||
   |20000102030405.1234
```

Note that the HL7 PID and HL7 PV1 segments are identical to the ones defined in Example 16. Since HL7 messages are not designed to transport data of big generic files, it is also possible to equipped a HL7 ORU R01 message with an UNC-path which points to the generic file that should be archived. In this particular case the HL7 ORU R01 message specifies the following information:

```
MSH|...
PID|...
PV1|...
OBR|...
OBX|1|RP|genericFileId^genericFileDescription^^genericContainerId
   ^genericContainerDescription^genericContainerCodingSystemId^^
   ^genericFilename.txt|1|\E\\E\host\E\share\E\path^^application^pdf||
   |interpretationCodeId^interpretationCodeDescription|||F||
   |20000102030405.1234
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, and HL7 OBR segments are identical to the ones of the previously defined HL7 ORU R01 message.

Example 18. The following HL7 MDM T02 message routes a generic file which has been archived by a third-party system using the generic file ID `genericFileId` and the application ID `sendingApplication`. The generic file is contained in a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. It belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`. The data of the generic file is embedded as a Base64 encoded string in the HL7 MDM T02 message.

```
MSH|^~\&|EventServer|organizationalUnit^configuredSendingFacility^L
   |configuredReceivingApplication|configuredReceivingFacility
   |20000102030405.1234||MDM^T02^MDM_T02|1|P|2.8.1|1|||||UNICODE UTF-8
PID|1||patientId^^^assigningAuthority^PI~patientSsn^^^assigningAuthority
   ^SS||lastName^firstName^middleName^suffix^prefix^^^A|
   |20000102030405.1234|M||race^raceDescription|patientStreet^
   ^patientCity^patientState^1234^patientCountry|||||||accountNumber^^
   ^assigningAuthority||||||||A^austrian
PV1|1|I^inpatient|||||||||||||||||visitId^^^assigningAuthority|||||||||||
   ||||||||||||||||||||alternateVisitId^^^assigningAuthority|V
ORC|OP|placerOrderNumber|fillerOrderNumber||CM
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription|||20000102030405.1234||||||||||
   |accessionNumber||||20000102030405.1234||diagnosticServiceSectionId
   ^diagnosticServiceSectionDisplayName
   ^diagnosticServiceSectionDescritpion|F
TXA|1|typeId^typeDisplayName^typeCodingScheme|contentPresentationId
   ^contentPresentationDisplayName^contentPresentationCodingScheme||
   |20000102030405.1234|||originator^originatorLastName
   ^originatorFirstName^originatorMiddleName^originatorSuffix
   ^originatorPrefix^^^^^^^^facilityTypeCodeId
   &facilityTypeCodeDisplayName&facilityTypeCodeCodingScheme^A||
   |1.2.3.4.5||||genericFilename.txt|completionStatus
   |confidentialityStatusId^confidentialityStatusDisplayName
   ^confidentialityStatusCodingScheme||||authenticator
   ^authenticatorLastName^authenticatorFirstName^authenticatorMiddleName
   ^authenticatorSuffix^authenticatorPrefix^^^^^^^^^^A|
   |genericContainerId^genericContainerDescription
   ^genericContainerCodingSystemId|genericFileDescription
```

```
OBX|1|ED|genericFileId^genericFileDescription^genericFileSystemId
   ^genericContainerId^genericContainerDescription
   ^genericContainerCodingSystemId|1|^text^plain^Base64^RXhhbXBsZQo=||
   |interpretationCodeId^interpretationCodeDescription|||F||
   |20000102030405.1234
```

The information provided by the HL7 TXA segment depend on the method that has been used to archive the generic file that should be routed. If it has been archived via a HL7 MDM T02 message, like the one of Example 16, the information shown above are provided. If a HL7 ORU R01 message similar to the one of Example 17 has been used to archive the generic file the HL7 TXA segment consists of the following information.

```
TXA|1|||||20000102030405.1234||||||1.2.3.4.5||||genericFilename.txt|F||||
   |||genericContainerId^genericContainerDescription
   ^genericContainerCodingSystemId|genericFileDescription
```

Since HL7 messages are not designed to transport data of big generic files, it is also possible to equipped a HL7 MDM T02 message with an UNC-path which points to the generic file that should be routed. In this particular case the HL7 MDM T02 message specifies the following information:

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|...
TXA|...
OBX|1|RP|genericFileId^genericFileDescription^genericFileSystemId
   ^genericContainerId^genericContainerDescription
   ^genericContainerCodingSystemId|1
   |//configuredHost/configuredShare/path^^application^pdf||
   |interpretationCodeId^interpretationCodeDescription|||F||
   |20000102030405.1234
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, HL7 ORC, HL7 OBR, and HL7 TXA segments are identical to the ones of the previously defined HL7 MDM T02 message.

Example 19. The following HL7 MDM T09 message updates the metadata of a generic file with generic file ID `genericFileId` and application ID `sendingApplication` as well as a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. Since only the metadata of a generic file are updated, no generic file is embedded or referenced in the HL7 MDM T09 message.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||MDM^T09^MDM_T01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
PV1|...
OBR|1||||||20000102030405.1234|||||||||||updatedAccessionNumber|||||
   |updatedDiagnosticServiceSectionId
   ^updatedDiagnosticServiceSectionDisplayName
   ^updatedDiagnosticServiceSectionDescritpion
TXA|1|updatedTypeId^updatedTypeDisplayName^updatedTypeCodingScheme
   |updatedContentPresentationId^updatedContentPresentationDisplayName
   ^updatedContentPresentationCodingScheme|||20000102030405.1234||
   |updatedOriginator^updatedOriginatorLastName
   ^updatedOriginatorFirstName^updatedOriginatorMiddleName
   ^updatedOriginatorSuffix^updatedOriginatorPrefix^^^^^^^
   ^updatedFacilityTypeCodeId&facilityTypeCodeDisplayName
   &updatedFacilityTypeCodeCodingScheme^A|||genericFileId|||
   |updatedGenericFilename.txt|updatedCompletionStatus
   |updatedConfidentialityStatusId
   ^updatedConfidentialityStatusDisplayName
   ^updatedConfidentialityStatusCodingScheme||||updatedAuthenticator
   ^updatedAuthenticatorLastName^updatedAuthenticatorFirstName
   ^updatedAuthenticatorMiddleName^updatedAuthenticatorSuffix
   ^updatedAuthenticatorPrefix^^^^^^^^^^A||genericContainerId
   ^updatedGenericContainerDescription^genericContainerCodingSystemId
   |updatedGenericFileDescription
OBX|1|||1||||updatedInterpretationCodeId
   ^updatedInterpretationCodeDescription|||||20000102030405.1234
```

Note that the HL7 PID and HL7 PV1 segments are identical to the ones defined in Example 16.

Example 20. The following HL7 MDM T10 message updates a generic file with generic file ID `genericFileId` and application ID `sendingApplication` as well as a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. The the new version of the generic file is referenced via an UNC path (it would be also possible to embed the new version of the generic file in the HL7 MDM T10 message).

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||MDM^T10^MDM_T02|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
PV1|...
OBR|...
TXA|...
OBX|1|RP||1|\E\\E\host\E\share\E\updatedPath^^application^pdf||
   |updatedInterpretationCodeId^updatedInterpretationCodeDescription|||||
   |20000102030405.1234
```

Note that the HL7 PID, HL7 PV1, HL7 OBR, and HL7 TXA segments are identical to the ones defined in Example 19.

Example 21. The following HL7 ORU R01 message updates the metadata of a generic file with generic file ID `genericFileId` and application ID `sendingApplication` as well as a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. Since only the metadata of a generic file are updated, no generic file is embedded or referenced in the HL7 ORU R01 message.

```
MSH|...
PID|...
PV1|...
OBR|1||||||20000102030405.1234||||F|||||||updatedAccessionNumber|||||
   |updatedDiagnosticServiceSectionId
   ^updatedDiagnosticServiceSectionDisplayName
   ^updatedDiagnosticServiceSectionDescritpion
OBX|1||genericFileId^updatedGenericFileDescription^^genericContainerId
   ^updatedGenericContainerDescription^genericContainerCodingSystemId^^
   ^updatedGenericFilename.txt|1||||updatedInterpretationCodeId
   ^updatedInterpretationCodeDescription|||F|||20000102030405.1234
```

Note that the HL7 MSH, HL7 PID, and HL7 PV1 segments are identical to the ones defined in Example 17.

Example 22. The following HL7 ORU R01 message updates a generic file with generic file ID `genericFileId` and application ID `sendingApplication` as well as a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. The the new version of the generic file is referenced via an UNC path (it would be also possible to embed the new version of the generic file in the HL7 ORU R01 message).

```
MSH|...
PID|...
PV1|...
OBR|...
OBX|1|RP|genericFileId^updatedGenericFileDescription^^genericContainerId
   ^updatedGenericContainerDescription^genericContainerCodingSystemId^^
   ^updatedGenericFilename.txt|1|\E\\E\host\E\share\E\updatedPath^
   ^application^pdf|||updatedInterpretationCodeId
   ^updatedInterpretationCodeDescription|||F|||20000102030405.1234
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, and HL7 OBR segments are identical to the ones defined in Example 21.

Example 23. The following HL7 MDM T02 message routes a generic file which has been updated by a third-party system using the generic file ID `genericFileId` and the application ID `sendingApplication`. The generic file is contained in a generic container with generic container ID `genericContainerId` and coding system ID `genericContainerCodingSystemId`. It belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`. The data of the generic file is embedded as a Base64 encoded string in the HL7 MDM T02 message (it would be also possible to reference the new version of the generic file via an UNC-path).

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription|||20000102030405.1234||||||||||
   |updatedAccessionNumber||||20000102030405.1234|
   |updatedDiagnosticServiceSectionId
   ^updatedDiagnosticServiceSectionDisplayName
   ^updatedDiagnosticServiceSectionDescritpion|F
TXA|1|updatedTypeId^updatedTypeDisplayName^updatedTypeCodingScheme
   |updatedContentPresentationId^updatedContentPresentationDisplayName
   ^updatedContentPresentationCodingScheme|||20000102030405.1234||
   |updatedOriginator^updatedOriginatorLastName
   ^updatedOriginatorFirstName^updatedOriginatorMiddleName
   ^updatedOriginatorSuffix^updatedOriginatorPrefix^^^^^^^
   ^updatedFacilityTypeCodeId&facilityTypeCodeDisplayName
   &updatedFacilityTypeCodeCodingScheme^A|||1.2.3.4.5|||
   |updatedGenericFilename.txt|updatedCompletionStatus
   |updatedConfidentialityStatusId
   ^updatedConfidentialityStatusDisplayName
   ^updatedConfidentialityStatusCodingScheme||||updatedAuthenticator
   ^updatedAuthenticatorLastName^updatedAuthenticatorFirstName
   ^updatedAuthenticatorMiddleName^updatedAuthenticatorSuffix
   ^updatedAuthenticatorPrefix^^^^^^^^^^A||genericContainerId
   ^updatedGenericContainerDescription^genericContainerCodingSystemId
   |updatedGenericFileDescription
OBX|1|ED|genericFileId^updatedGenericFileDescription^genericFileSystemId
   ^genericContainerId^updatedGenericContainerDescription
   ^genericContainerCodingSystemId|1|^text^plain^Base64
   ^VXBkYXRlZCBFeGFtcGxlCg==|||updatedInterpretationCodeId
   ^updatedInterpretationCodeDescription|||F|||20000102030405.1234
```

The information provided by the HL7 TXA segment depend on the method that has been used to archive and update the generic file that should be routed. If it has been archived and updated via HL7 MDM messages, like the ones of Example 16, 19, and 20, the information shown above are provided. If HL7 ORU messages similar to the ones of Examples 17, 21, and 22 have been used to archive and update the generic file the HL7 TXA segment consists of the following information.

```
TXA|1|||||20000102030405.1234||||||1.2.3.4.5|||
   |updatedGenericFilename.txt|F|||||||genericContainerId
   ^updatedGenericContainerDescription^genericContainerCodingSystemId
   |updatedGenericFileDescription
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, and HL7 ORC segments are identical to the ones defined in Example 18.

Example 24. The following HL7 MDM T11 message deletes a generic file with generic file ID `genericFileId` and application ID `sendingApplication`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||MDM^T11^MDM_T01|1|P|2.8.1|||||
   |UNICODE UTF-8
PID|...
PV1|...
OBR|...
TXA|...
```

Note that the HL7 PID, HL7 PV1, HL7 OBR, and HL7 TXA segments are identical to the ones defined in Example 19.

Example 25. The following HL7 ORU R01 message deletes a generic file with generic file ID `genericFileId` and application ID `sendingApplication`.

```
MSH|...
PID|...
PV1|...
OBR|1||||||20000102030405.1234||||D|||||||updatedAccessionNumber|||||
   |updatedDiagnosticServiceSectionId
   ^updatedDiagnosticServiceSectionDisplayName
   ^updatedDiagnosticServiceSectionDescritpion
OBX|1||genericFileId^updatedGenericFileDescription^^genericContainerId
   ^updatedGenericContainerDescription^genericContainerCodingSystemId^^
   ^updatedGenericFilename.txt|1||||updatedInterpretationCodeId
   ^updatedInterpretationCodeDescription|||D|||20000102030405.1234
```

Note that the HL7 MSH, HL7 PID, and HL7 PV1 segments are identical to the ones defined in Example 21.

A.3 Referencing of DICOM Studies and Generic Containers

Some of the mostly used methods to access, share, and delete DICOM Studies and generic containers are based on reference pointers. To ensure that third-party systems can use these methods to meet their requirements, reference pointers are automatically transmitted to third-party systems via HL7 ORU messages. Since the operational reliability of reference pointers might be influenced if DICOM Studies and generic containers are modified, *syngo*.share updates and deletes reference pointer automatically by transmitting appropriate HL7 ORU messages. The functionalities based on reference pointers are illustrated in Figure 3. In the following the HL7 ORU messages used by third-party systems to share and delete DICOM Studies and generic containers as well as the HL7 ORU messages that are used by *syngo*.share to transmit and synchronize reference pointers are illustrated with help of a typical patient encounter which reflects the the creation and update of a DICOM Study and report as well as the sharing and deletion of the created documents.

Assume that a patient has arrived at a healthcare facility, the personal data of the patient has been stored and forwarded to *syngo*.share, and an order has been scheduled by the RIS and propagated to *syngo*.share in order to be able to proceed with the medication of the arrived patient. At next the order is processed by a modality which sends the created DICOM Images to *syngo*.share. As soon as all DICOM Images have been archived, a reference pointer addressing the corresponding DICOM Study is sent to the dedicated third-party systems, in this particular example the HIS and RIS, via HL7 ORU R01 messages similar to the one shown in Example 26. Afterwards the DICOM Study is analyzed by a radiologist. During the appraisal the radiologist might recognize that some metadata of the created DICOM Study are incorrect. To correct the spotted mistakes, the RIS updates the performed order and thus the corresponding DICOM Study by sending an appropriate HL7 message (to ensure that *syngo*.share updates the order information of the DICOM Study, the metadata update has to be activated). Once the update has been conducted, the previously inserted reference pointer is updated via a HL7 ORU R01 message like the one of Example 27. At next the report created by the radiologist is archived by *syngo*.share. As before a reference pointer addressing the stored document is sent to the HIS and RIS via a HL7 ORU R01 message similar to the one

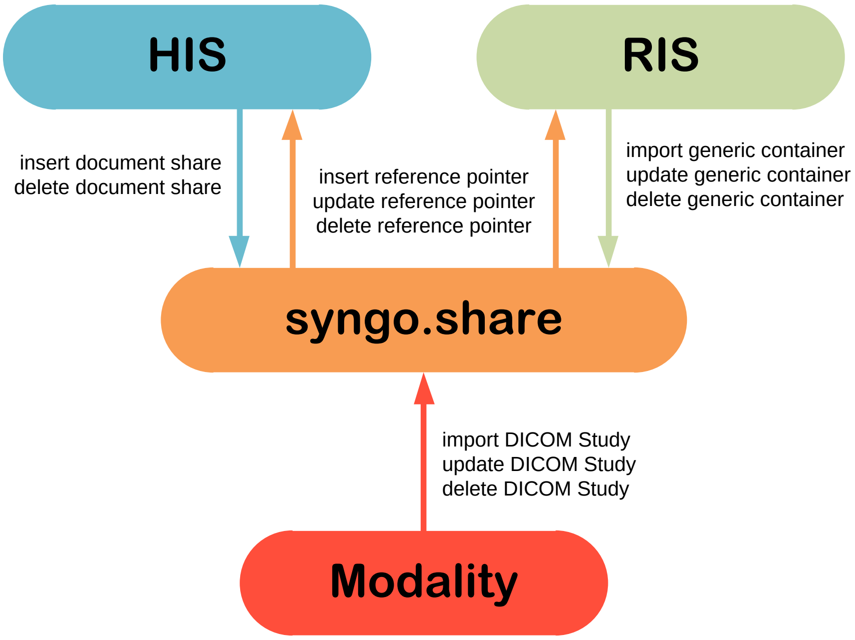

Figure 3: Reference Pointer Workflow

specified by Example 28. If the report is update, an additional HL7 ORU RO01 as illustrated by Example 29 is sent to the HIS and RIS to update the inserted reference pointer. After that the HIS shares the DICOM Study and report to inform the attending doctor about the availability of the two documents as well as to ensure that the attending doctor can access the provided information. To this end the HIS sends two HL7 ORU RO1 messages to *syngo*.share (see Examples 30 and 31). As soon as the medication of the patient is finished, the HIS might revoke the created shares by sending HL7 ORU R01 messages as defined in Examples 32 and 33. Finally, to delete the DICOM Study and report after a certain amount of time, the RIS sends specific HL7 ORU R01 messages like the ones shown in Examples 34 and 35. Since the two documents are not longer accessible, *syngo*.share instructs the HIS and RIS to delete the associated reference pointers by sending HL7 ORU messages similar to the ones illustrated by Examples 36 and 37
Example 26. The following HL7 ORU R01 message inserts a reference pointer pointing to a DICOM Study with Study Instance UID `1.2.3.4.5.6.7.8.9`. The referenced DICOM Study belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`.

```
MSH|^~\&|EventServer|organizationalUnit^configuredSendingFacility^L
   |configuredReceivingApplication|configuredReceivingFacility
   |20000102030405.1234||ORU^O01^ORU_O01|1|P|2.8.1|1|||||UNICODE UTF-8
PID|1||patientId^^^assigningAuthority^PI~patientSsn^^^assigningAuthority
   ^SS||lastName^firstName^middleName^suffix^prefix^^^A|
   |20000102030405.1234|M||race^raceDescription|patientStreet^
   ^patientCity^patientState^1234^patientCountry|||||||accountNumber^^
   ^assigningAuthority||||||||A^austrian
PV1|1|I^inpatient|||||||||||||||||visitId^^^assigningAuthority|||||||||||
   ||||||||||||||||||||alternateVisitId^^^assigningAuthority|V
ORC|OP|placerOrderNumber|fillerOrderNumber||CM
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription||20000102030405.1234||||||||||
   |accessionNumber||||20000102030405.1234|||F
OBX|1|RP|1.2.3.4.5.6.7.8.9^dicomStudyDescription^^dicomStudyId|1
   |RP:2:TENANT_NAME:DICOM_STUDY:1.2.3.4.5.6.7.8.9^^^^100|||CT|||F||
   |20000102030405.1234|organizationalUnit^organizationalUnitDescription
```

Example 27. The following HL7 ORU R01 message updates a reference pointer pointing to a DICOM Study with Study Instance UID `1.2.3.4.5.6.7.8.9`. The referenced DICOM Study belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`.

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription|||20000102030405.1234|||||||||
   |updatedAccessionNumber||||20000102030405.1234|||F
OBX|1|RP|1.2.3.4.5.6.7.8.9^updatedDicomStudyDescription^^dicomStudyId|1
   |RP:2:TENANT_NAME:DICOM_STUDY:1.2.3.4.5.6.7.8.9^^^^100|||MR|||F||
   |20000102030405.1234|organizationalUnit^organizationalUnitDescription
```

Note that the HL7 MSH, HL7 PID, HL7 PV1 and HL7 ORC segments are identical to the ones defined in Example 26.

Example 28. The following HL7 ORU R01 message inserts a reference pointer pointing to a generic container with generic container UID `1.2.3.4.5`. The referenced generic container belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`.

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|...
OBX|1|RP|1.2.3.4.5^genericContainerDescription^^genericContainerId|1
   |RP:2:TENANT_NAME:GENERIC_CONTAINER:1.2.3.4.5^^genericFileId^^1|||.pdf
   |||F|||20000102030405.1234|organizationalUnit
   ^organizationalUnitDescription
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, HL7 ORC, and HL7 OBR segments are identical to the ones defined in Example 26.

Example 29. The following HL7 ORU R01 message updates a reference pointer pointing to a generic container with generic container UID `1.2.3.4.5`. The referenced generic container belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`.

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|...
OBX|1|RP|1.2.3.4.5^updatedGenericContainerDescription^^genericContainerId
   |1|RP:2:TENANT_NAME:GENERIC_CONTAINER:1.2.3.4.5^^genericFileId^^1||
   |.pdf|||F|||20000102030405.1234|organizationalUnit
   ^organizationalUnitDescription
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, HL7 ORC, and HL7 OBR segments are identical to the ones defined in Example 27.

Example 30. The following HL7 ORU R01 message creates a share for a DICOM Study with Study Instance UID `1.2.3.4.5.6.7.8.9` so that it can be accessed by a user with login name `user`.

```
MSH|^~\&|sendingApplication|organizationalUnit|receivingApplication
   |receivingFacility|20000102030405.1234||ORU^O01^ORU_O01|1|P|2.8.1|||||
   |UNICODE UTF-8
OBX|1|RP||1|RP:2:TENANT_NAME:DICOM_STUDY:1.2.3.4.5.6.7.8.9||||||F||
   |20000102030405.1234
PRT|user|LI||PI|user^userLastName^userFirstName^userMiddleName^userSuffix
   ^userPrefix^^^^^^^^^A|||organizationName^organizationNameTypeCode|^
   room^^facility|deviceId
```

Example 31. The following HL7 ORU R01 message creates a share for a generic container with generic container UID `1.2.3.4.5` so that it can be accessed by a user with login name `user`.

```
MSH|...
OBX|1|RP||1|RP:2:TENANT_NAME:GENERIC_CONTAINER:1.2.3.4.5||||||F||
   |20000102030405.1234
PRT|...
```

Note that the HL7 MSH and HL7 PRT segments are identical to the ones defined in Example 30.

Example 32. The following HL7 ORU R01 message deletes a share for a DICOM Study with Study Instance UID `1.2.3.4.5.6.7.8.9` so that it can not longer be accessed by a user with login name `user`.

```
MSH|...
OBX|...
PRT|user|UN||PI|user^userLastName^userFirstName^userMiddleName^userSuffix
   ^userPrefix^^^^^^^^^A|||organizationName^organizationNameTypeCode|
   ^room^^facility|deviceId
```

Note that the HL7 MSH and HL7 OBX segments are identical to the ones defined in Example 30.

Example 33. The following HL7 ORU R01 message deletes a share for a generic container with generic container UID `1.2.3.4.5` so that it can not longer be accessed by a user with login name `user`.

```
MSH|...
OBX|1|RP||1|RP:2:TENANT_NAME:GENERIC_CONTAINER:1.2.3.4.5||||||F||
   |20000102030405.1234
PRT|...
```

Note that the HL7 MSH and HL7 PRT segments are identical to the ones defined in Example 32.

Example 34. The following HL7 ORU R01 message deletes a DICOM Study with Study Instance UID `1.2.3.4.5.6.7.8.9`.

```
MSH|...
OBX|1|RP||1|RP:2:TENANT_NAME:DICOM_STUDY:1.2.3.4.5.6.7.8.9||||||D||
   |20000102030405.1234
```

Note that the HL7 MSH segment is identical to the one defined in Example 30.

Example 35. The following HL7 ORU R01 message deletes a generic container with generic container UID `1.2.3.4.5`.

```
MSH|...
OBX|1|RP||1|RP:2:TENANT_NAME:GENERIC_CONTAINER:1.2.3.4.5||||||D||
   |20000102030405.1234
```

Note that the HL7 MSH segment is identical to the one defined in Example 30.

Example 36. The following HL7 ORU R01 message deletes a reference pointer pointing to a DICOM Study with Study Instance UID `1.2.3.4.5.6.7.8.9`. The referenced DICOM Study belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`.

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|1|placerOrderNumber|fillerOrderNumber|universalServiceId
   ^universalServiceDescription|||20000102030405.1234||||||||||
   |updatedAccessionNumber||||20000102030405.1234|||D
OBX|1|RP|1.2.3.4.5.6.7.8.9^updatedDicomStudyDescription^^dicomStudyId|1
   |RP:2:TENANT_NAME:DICOM_STUDY:1.2.3.4.5.6.7.8.9^^^^100|||MR|||D||
   |20000102030405.1234|organizationalUnit^organizationalUnitDescription
```

Note that the HL7 MSH, HL7 PID, HL7 PV1 and HL7 ORC segments are identical to the ones defined in Example 26.

Example 37. The following HL7 ORU R01 message deletes a reference pointer pointing to a generic container with generic container UID `1.2.3.4.5`. The referenced generic container belongs to the organizational unit `organizationalUnit` and is associated with a visit with visit ID `visitId` of a patient with patient ID `patientId`.

```
MSH|...
PID|...
PV1|...
ORC|...
OBR|...
OBX|1|RP|1.2.3.4.5^updatedGenericContainerDescription^^genericContainerId
   |1|RP:2:TENANT_NAME:GENERIC_CONTAINER:1.2.3.4.5^^genericFileId^^1||
   |.pdf|||D|||20000102030405.1234|organizationalUnit
   ^organizationalUnitDescription
```

Note that the HL7 MSH, HL7 PID, HL7 PV1, HL7 ORC, and HL7 OBR segments are identical to the ones defined in Example 36.

Distributed by
Siemens Healthcare GmbH
Henkestr. 127
91052 Erlangen
Germany
Phone: +49 9131 84-0
siemens.com/healthcare

Legal Manufacturer
ITH icoserve technology for healthcare GmbH
Innrain 98
6020 Innsbruck
Austria
Phone: +43 512 89059

www.siemens.com/syngo.share

Made in Austria