CitiusTech

Introduction to HL7 Messaging Specifications

Session 1

Version 1.0
May 2020

Agenda

- **Why HL7?**
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Why HL7?

- In all hospitals or healthcare provider organizations, hospital activities such as billing records or patient tracking are usually handled by various computer systems
- Ideally, these systems should be communicating / interfacing with each other on receiving information
- This sharing and receiving of information should be based on a standard-based messaging
- HL7 helps healthcare systems to communicate through a set of HL7-defined but flexible standard using its guidelines and methodologies
- These guidelines are a set of rules that enable information sharing and processing in a manner which is standard and consistent, enabling smooth sharing of clinical information between healthcare organizations
- Ideally, this information sharing should help curb the occurrence of geographically-isolated and highly-variable medical care

Agenda

- Why HL7?
- **Information Systems**
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Information Systems (1/4)

Various information systems in the healthcare enterprise

- Appointment schedulers
- Patient Registration Systems
- Service “order filler” systems, e.g.
  - Laboratory
  - Radiology
  - Surgery
- Billing Systems
- Others

Information Systems (2/4)

What is shared by these information systems?

- Patient demographic data
- Patient healthcare history
- Practitioner Information
- Hospital and clinical information
- Other common data

Parameters	Patients (n =
Demographics	
Age (yrs)	65 ± 15
Male sex	62 (50%)
Atherogenic risk factors	
Arterial hypertension	81 (65%)
Hypercholesterolemia	43 (35%)
Diabetes mellitus	13 (10%)
Family history	32 (26%)
Smoking	37 (30%)

Information Systems (3/4)

How do Information Systems Communicate?

- Almost all healthcare organizations rely on messaging standards for information exchange between systems and applications
- Every message in a healthcare workflow works as a vital link
- If messages generate errors then details related to patient admission, billing, laboratory results, and other vital data cannot be processed from one system to another

Modes of communication involve two types of processing:

- Batch Processing
  - One or more message can be transferred through batch files (e.g. patient data) to be sent in a single file
  - Batch processing is useful for systems that are not connected via real-time transmission protocols (such as those systems that do not transmit via TCP/IP connections)
- Real-time Processing
  - Messages are transferred in real-time between systems

Information Systems (4/4)

Messages needs to be in a format for carrying the data

- Format can be defined by the enterprise
  - Messages can travel only between the enterprise boundary
- Format can be defined by the system manufacturer
  - Messages can travel only between the systems which are defined by the manufacturer

Or

- Format can be standards based
  - Messages can be shared between any provider or any healthcare entity
  - HL7 being a standard can be used for exchanging information across providers

Agenda

- Why HL7?
- Information Systems
- **Introduction**
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Introduction (1/3)

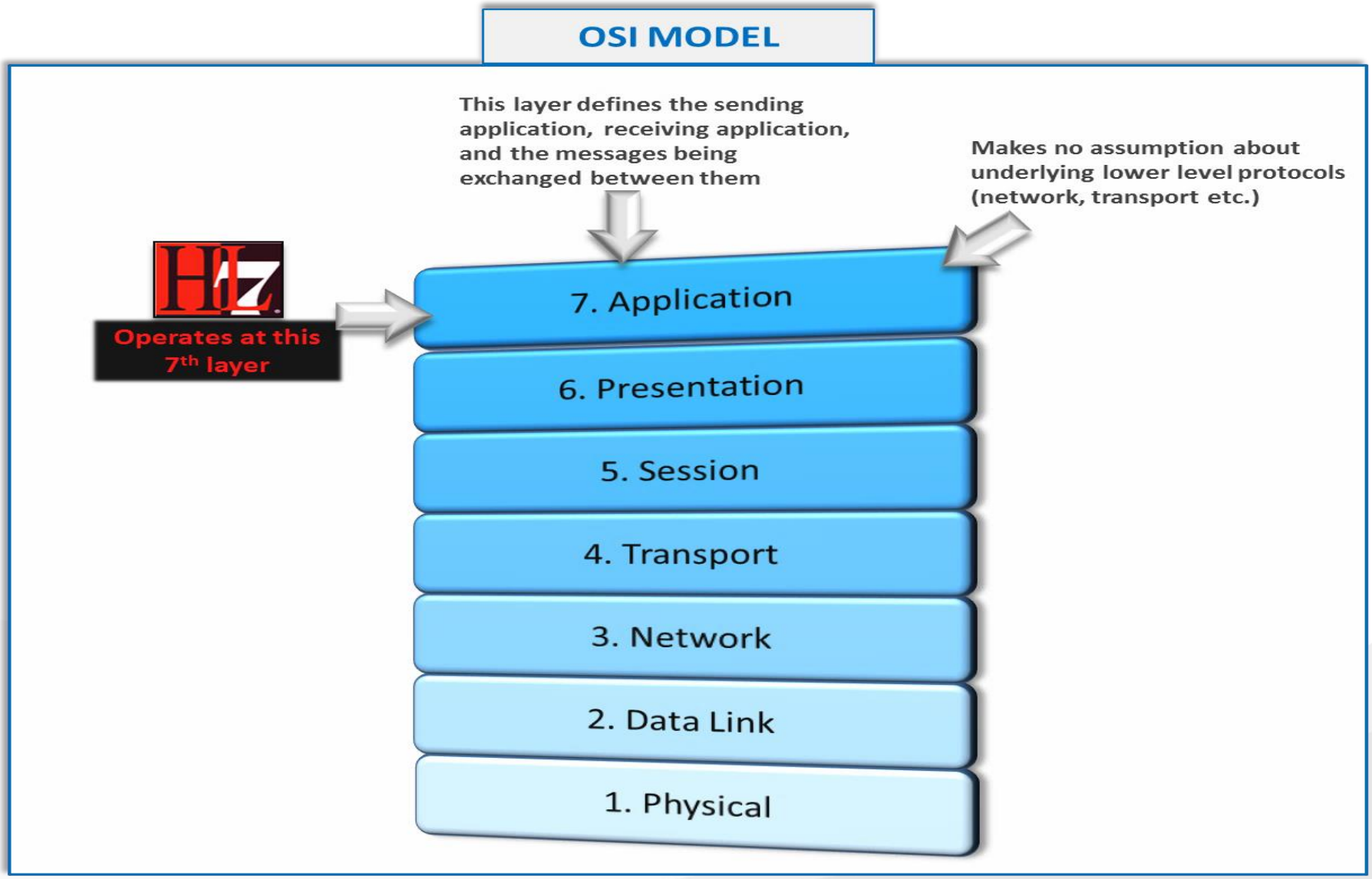
What does HL7 provide?

- A set of predefined logical formats for creating messages that defines the elements of the message
- Assists in physical message formatting by:
  - Providing HL7 default formatting also known as Encoding Rules/7 or ER/7
  - Providing guidance for using other formats such as XML
- Guidelines for communicating:
  - Information on the lower layer protocols
  - Acknowledgment protocols

What is HL7?

HL7 offers a standard for data exchange among healthcare applications, thus enabling communication in healthcare settings.

Introduction (2/3)



Introduction (3/3)

Where does HL7 Operate?

- Patient Administration
- Order Entry
- Financial Management
- Observation Reporting
- Master files and Indexes
- Medical Records / Information Management
- Scheduling and Logistics
- Patient Care
- Patient Referral

Agenda

- Why HL7?
- Information Systems
- Introduction
- **HL7 Versions**
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

HL7 Versions (1/4)

- HL7 has got multiple versions like HL7 V2.x (V2.1, V2.2, V2.3 etc.), HL7 V3.0, HL7 CDA, FHIR
- HL7 V2.x is a messaging protocol which is ASCII text-based which had lot of optionality defined in the standard. It is the most popular one which is used till date for communicating between healthcare applications
- HL7 V3.0 was based on the RIM Model which covered object-oriented classes to capture content that needed to be shared among systems with reduced level of optionality
- Based on RIM, CDA (Clinical Document Architecture) was developed. Unlike HL7 V2.8, HL7 V3, HL7 CDA messaging is entirely XML based.
- FHIR is based on the concept of Resources which can be exchanged in JSON, XML or RDF format

HL7 Versions (2/4)


```
MSH|^~\&|GHH LAB|ELAB-3|GHH OE|BLDG4|200202150930||ORU^R01|CNTRL-3456|P|2.8<cr>
PID|||555-44-4444||EVERYWOMAN^EVE^E^^^^L|GREEN|19620320|F|||153 ROSEWOOD DR.^
^STATESVILLE^OH^32292||(206)3445232||||AC555444444||67-A4335^OH^20030520<cr>
OBR|1|845439^GHH OE|1045813^GHH LAB|15545^GLUCOSE|||200202150730|||||||||
555-55-5555^PRIMARY^JANE P^^^^PHD^^|||||||||F||||||444-44-4444^HIPPOCRATES^HOWARD H^^^^MD<cr>
OBX|1|SN|1554-5^GLUCOSE^POST 12H CFST:MCNC:PT:SER/PLAS:QN||^180|mg/dl|70_105|H|||F<cr>
```


```
    <content styleCode="Bold">Walter White, the 7<sup>th</sup>
            </content> is a 60 year old male referred for further diabetes management. Onset of diabetes in his
            <content revised="delete">twenties</content>
            <content revised="insert">teens</content>
            His blood pressure has been on the rise for the last two months.
            </text>
    </section>
</component>
```

HL7 Versions (3/4)


```
<HL7Interaction xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <id root=' ' extension=' '/>
 <creationTime value=' '/>
 <interactionId extension='HL7Interaction' root='2.16.840.1.113883'/>
 <processingCode code='D|P|T'/>
 <processingModeCode code='A|T|I|R'/>
 <acceptAckCode code='AL|ER|NE'/>
 <receiver typeCode="RCV">
   <device determinerCode="INSTANCE">
     <id/>
     <name/>
     <desc/>
     <existenceTime><low value=' '/></existenceTime>
     <telecom value=' ' />
     <manufacturerModelName/>
     <softwareName/>
   </device>
 </receiver>
 <sender typeCode="SND">
   <device determinerCode="INSTANCE">
     <id/>
     <name/>
     <desc/>
     <existenceTime><low value=' '/></existenceTime>
     <telecom value=' ' />
     <manufacturerModelName/>
     <softwareName/>
   </device>
 </sender>
 <controlActProcess>
 </controlActProcess>
</HL7Interaction>
```

HL7 Versions (4/4)


```
{
  "resourceType": "Patient",
  "identifier": [
    {
      "system": "abc.com/values",
      "value": "9000"
    }
  ],
  "name": [
    {
      "family": "Dsouza",
      "given": [
        "Fiona",
        "A"
      ]
    }
  ],
  "gender": "female",
  "birthDate": "1988-08-06",
}
```

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- **Conceptual Approach**
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

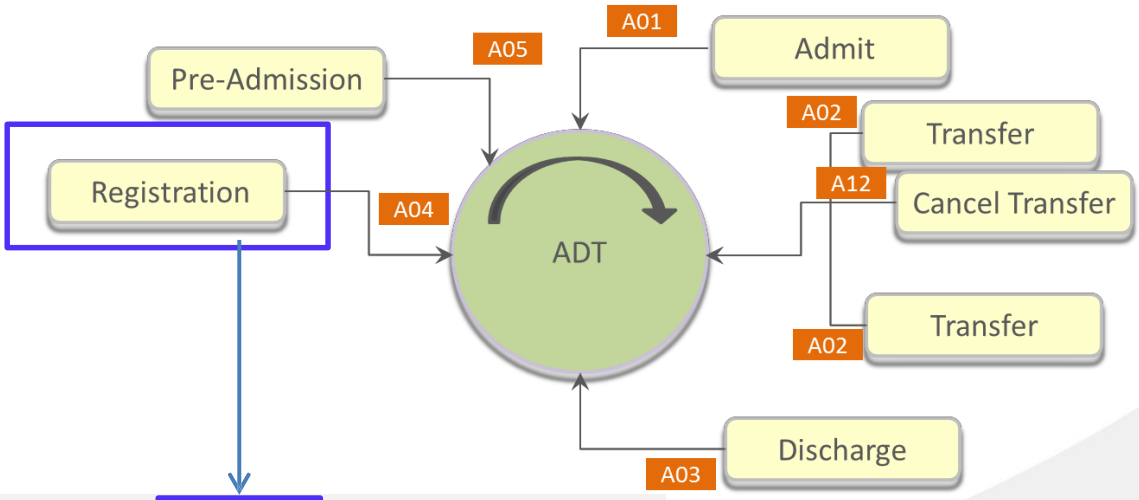
Conceptual Approach (1/3)

Trigger event

- An event in the real world of healthcare creates the need for data to flow among systems.

Rules for trigger events

- Each trigger event is associated with only one message type (For ex. ADT)
- A message type may have multiple trigger events associated with it. (For ex. A01, A02, A03…)



MSH|^~\&|EPIC|EPICADT|SMS|SMSADT|199912271408|CHARRIS|ADT^A04|1817457|D|2.5|
PID||0493575^^^2^ID 1|454721||DOE^JOHN^^^^|DOE^JOHN^^^^|19480203|M||B|254 MYSTREET AVE^^MYTOWN^OH^44123^USA||(216)123-4567|||M|NON|4000034403~1129086

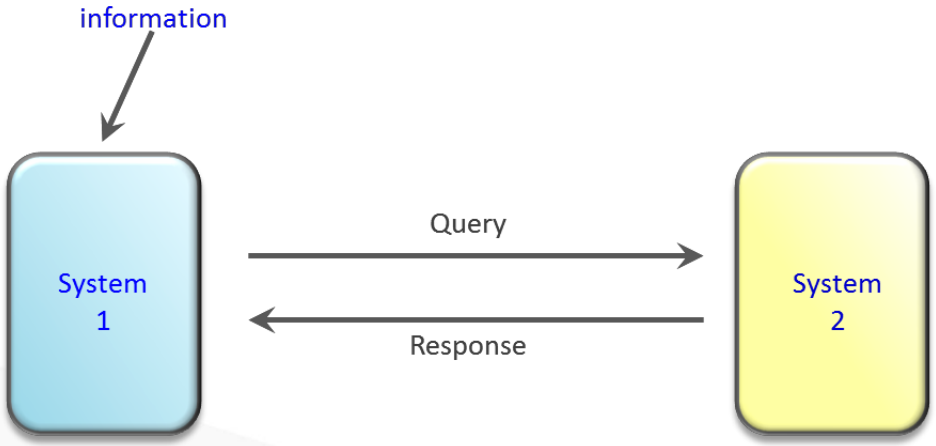
Conceptual Approach (2/3)

Models of data flow for HL7 are:

- Declarative Model: Unsolicited update/acknowledgment
- Interrogative Model: Query/Response

Query/Response is used for collecting:

- Data regarding a single patient, for e.g., send all lab results for patient #123456
- Data regarding multiple patients, for e.g., send the list of patients whose attending physician is Dr. #123
- Data that is not patient related, for e.g., send the age specific normal values for serum protein



Conceptual Approach (3/3)

Unsolicited update/acknowledgment

- When transfer of information is initiated by the application system that deals with the triggering event, the transaction is termed as an **unsolicited update**.
- For e.g. ADT/ACK – admit/visit notification (event A01). An A01 event can be used to notify:
  - Nursing system that the patient has been admitted and needs a care plan prepared
  - Pharmacy system that the patient has been admitted and will need drugs to be prescribed.
  - Finance system for starting a billing period for the admitted patient
  - Dietary system that the patient has been admitted and requires dietary services

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- **Communications Environment**
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Communication Environment (1/2)

Universe of environments of interest to HL7 includes:

- Ad hoc environments that do not provide even basic transport reliability.
  - For e.g. point-to-point RS-232 links, modems
- Environments that support a robust transport level, but do not meet the high-level requirements.
  - For e.g. TCP/IP, DECNET, and SNA
- ISO and proprietary networks that implement up to presentation and other high-level services
  - For e.g. IBM's SNA LU6.2 and SUN Microsystems's NFS
- Two or more applications running on the same physical and/or logical machine that are not tightly integrated
  - For e.g. Pipes in a UNIX System

Communication Environment (2/2)

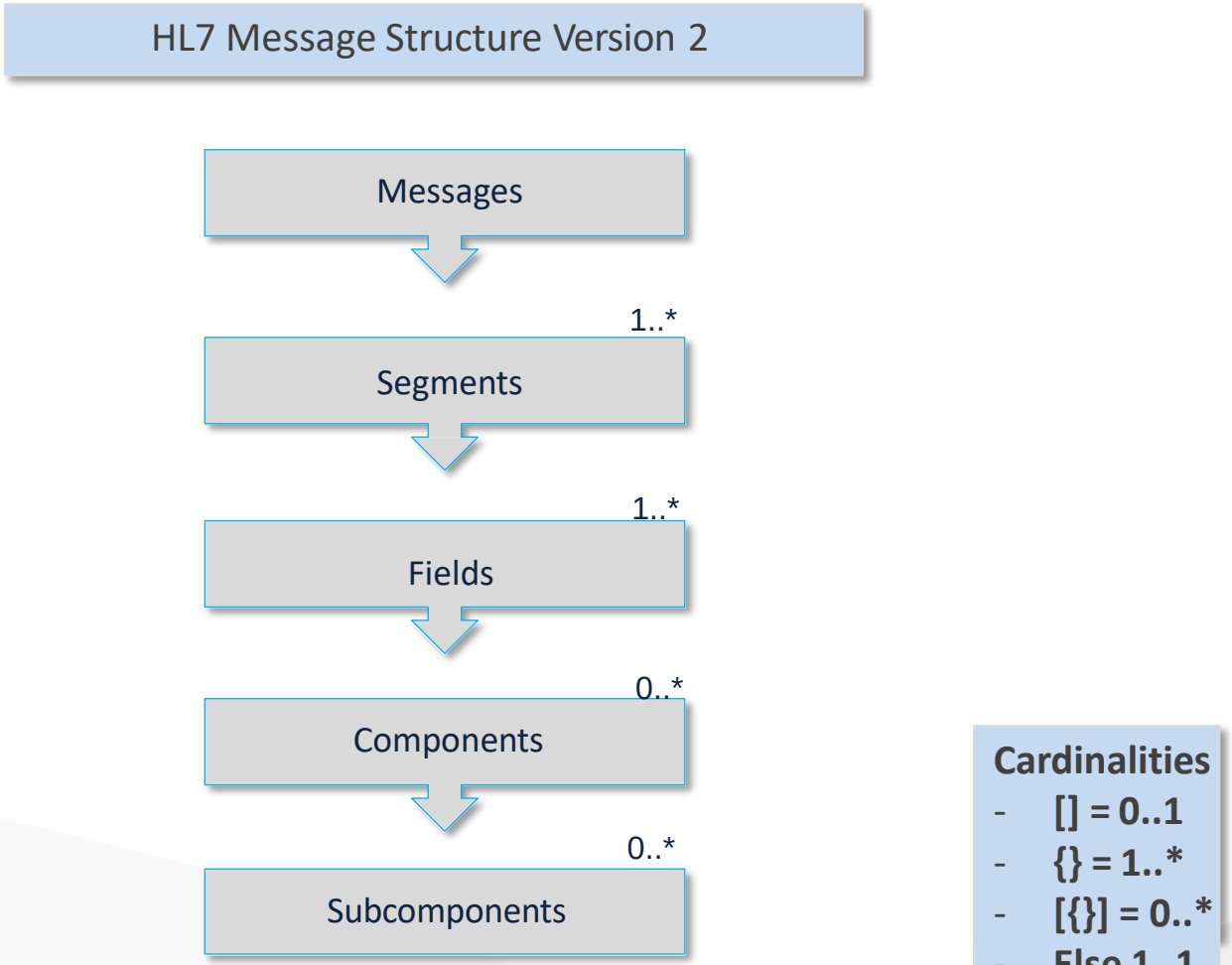
HL7 standard assumes that the communications environment will provide the following capabilities:

Error free transmission	Applications can assume that they correctly received all of the transmitted bytes in the order in which they were sent. However, it is recommended that sending applications may not assume that the message was actually received without receiving an acknowledgment message.
Character conversion	If two machines exchanging data use different representations of the same character set, the communication environment will be responsible to convert the data from one representation to the other.
Message length	HL7 sets no limits on the maximum size of HL7 messages. Standard assumes that the communication environment can transport messages of any length that might be necessary. In practice, sites may agree to place some upper bound on the size of messages along with the usage of message continuation protocol.

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- **Message Framework**
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Message Framework (1/4)



Cardinalities
- **[] = 0..1**
- **{} = 1..***
- **[{}] = 0..***
- **Else 1..1**

Message Framework (2/4)

Message

- Message is the atomic unit of data transferred between systems
- Each message has a message type that defines its purpose
- For example, the ADT Message type is used to transmit portions of a patient's Patient Administration (ADT) data from one system to another
- There exists one-to-many relationship between message types and trigger event codes
- Message type may be associated with more than one trigger event
- Message is comprised of a group of segments in a defined sequence

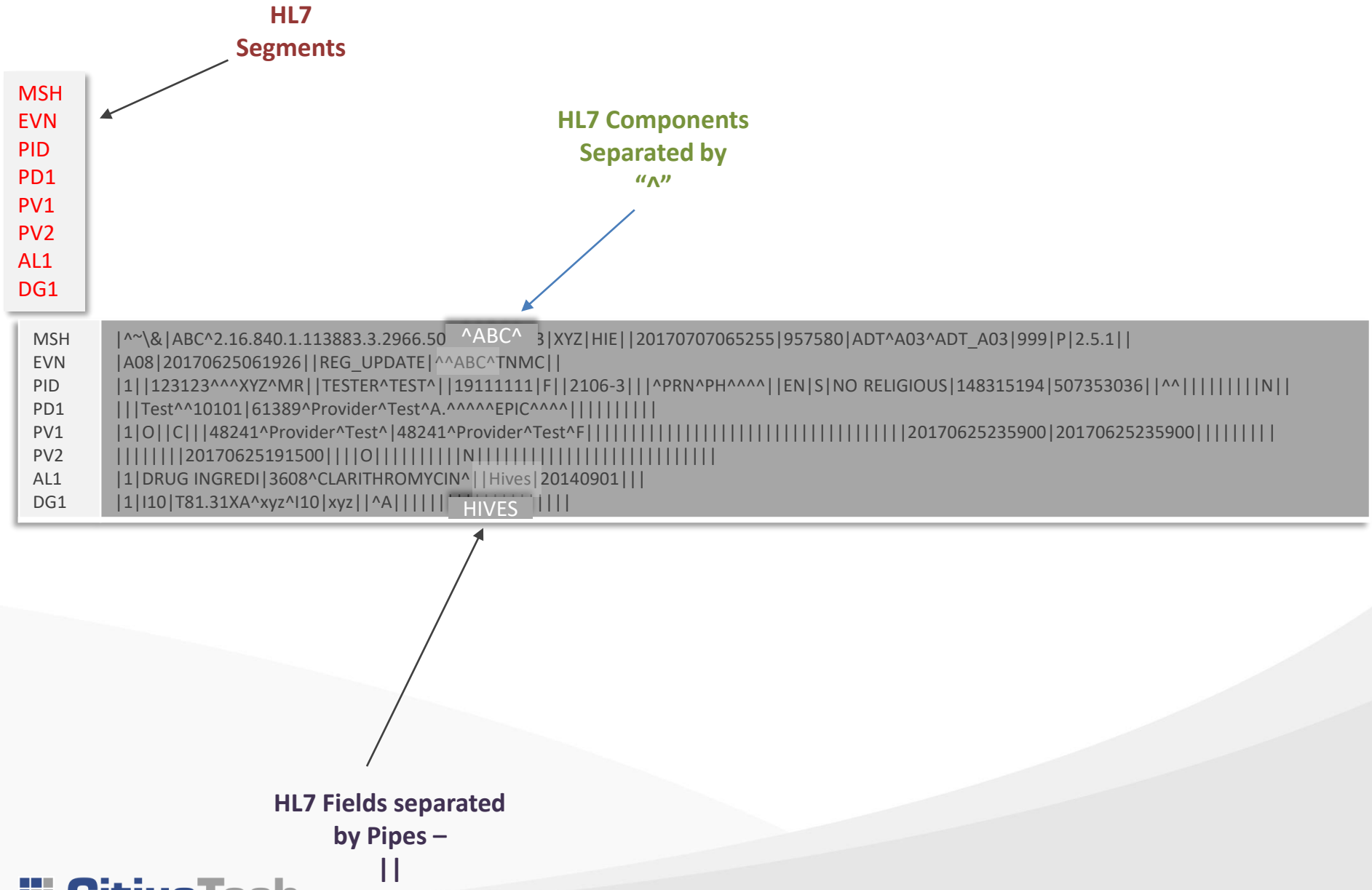
Segments and Segment Groups

- Segment is a logical grouping of data fields. Segments of a message may be required or optional
- They may occur only once in a message or they may be allowed to repeat
- Each segment is given a name
- Each segment is identified by a unique three-character code known as the Segment ID
- All segment ID codes beginning with the letter Z are reserved for locally defined segments
- Two or more segments may be organized as a logical unit called a segment group

Fields

- A field is a string of characters
- Fields for use within HL7 segments are defined by HL7
- Fields consists of components and subcomponents

Message Framework (3/4)



Message Framework (4/4)

- Message type: Indication that the data in the HL7 message belongs to a particular set of actions.
  - ADT : Admission, Discharge, Transfer
  - ACK : Acknowledgment
- Triggers : Creates the need for data to flow between healthcare systems
  - A01 : Inpatient Admission
  - A04 : Outpatient Registration
  - A05 : Pre-Admission
  - A11 : Cancel Admission/Registration

HL7 ADT Messages

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- **Message Construction Rules**
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Message Construction Rules (1/15)

Message Delimiter

Delimiter	Suggested Value	Encoding Character Position	Usage
Segment Terminator	<cr>	-	Terminates a segment record. This value cannot be changed by implementers
Field Separator	|	-	Separates two adjacent data fields within a segment. It also separates the segment ID from the first data field in each segment.
Component Separator	^	1	Separates adjacent components of data field where allowed.
Subcomponent Separator	&	4	Separates adjacent components of data field where allowed. If there are no subcomponents, this character may be omitted.
Repetition Separator	~	2	Separates multiple occurrences of a field where allowed.
Escape Character	\	3	Escape character for use with any field represented by a string data type. If no escape characters are used in a message, this character may be omitted. However, it must be present if subcomponents are used in the message.
Truncation	#	-	"#" denotes that the value of the field may be truncated. The truncation character "#" should be used at the end of a truncated value to indicate it was truncated. Conformant applications shall not truncate a value that is shorter than the length specified.

Message Construction Rules (2/15)

Syntax - Using message delimiters

```
MSH|^~\&| …..
XXX|field1|component1^component2^subcomponent3.1&subcomponent3.2^component4| …..
YYY|repetition1~repetition2| …..
ZZZ|data includes escaped \|\~ special characters| …..
```

HL7 Message Structure

Message Construction Rules (3/15)

- In the example, after the “MSH” segment-identifier code, the Message Header segment uses the field separator, “|” to establish which character serves as the field separator throughout the message
- In the next field, the four characters “^~\&” respectively establish the component separator character, the repetition character, the escape character, and the sub- component separator character that will apply throughout the message
- The hypothetical “XXX” segment includes field1 with no internal structure, but the next field has several components separated by “^”, and the third of these is made up of two sub-components separated by “&”
- Repetition is allowed in the first field of the hypothetical “YYY” segment (in the above example, the values “repetition1” and “repetition2”). The field of the hypothetical “ZZZ” segment has a text value that includes the characters “|~”, which are escaped to prevent their normal structural interpretation

HL7 Message Structure

Message Construction Rules (4/15)

Fields, Components, and Sub-Components

```
PID|Field1|Compnent1^Component2|Component1^Sub-component1&Sub-component2^Component3|Repeat1~Repeat2
```

Segment - *eg. PID*
- Field1 - *Simple*
- Field2 - *Has Components*
  - Component1
  - Component2
- Field3 - *Components and Sub-Components*
  - Component1
  - Component2
    - Sub-Component1
    - Sub-Component2
  - Component3
- Field4 - *Repeating field value*
  - Repeat1
  - Repeat2

Message Construction Rules (5/15)

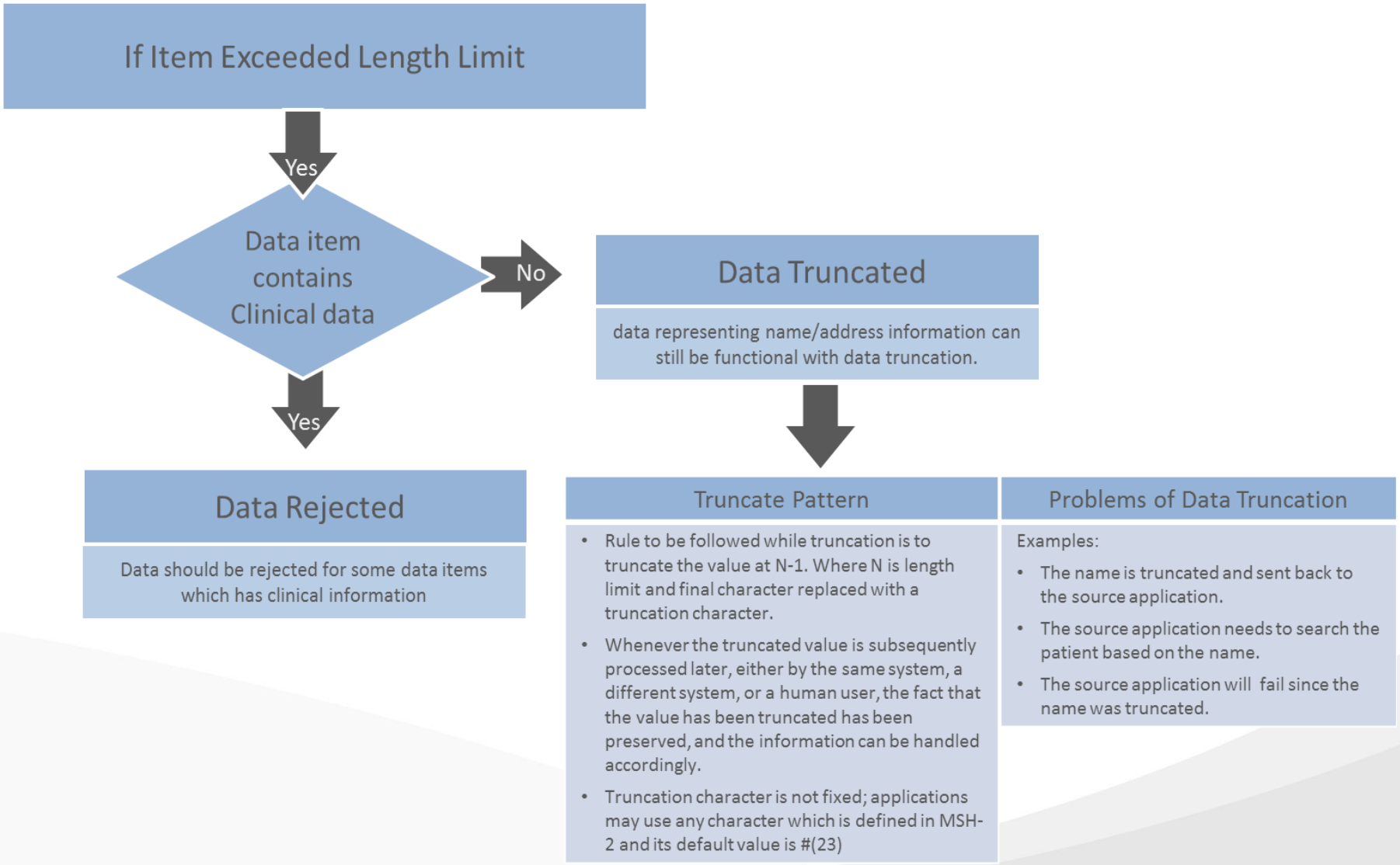
MSH – Message Header Segment

MHS|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3|||||||

SEQ	LEN	DT	OPT	Element Name	MSH Segment Example
1	1	DT	R	Field Separator	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
2	4	ST	R	Encoding Character	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
3	180	HD	O	Sending Application	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
4	180	HD	O	Sending Facility	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
5	180	HD	O	Receiving Application	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
6	180	HD	O	Receiving Facility	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
7	26	TS	O	Date/ Time of Message	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
8	40	ST	O	Security	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
9	7	CM	R	Message Type	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
10	20	ST	R	Message Control ID	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
11	3	PT	R	Processing ID	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||
12	8	ID	R	Version ID	MSH|^~\&|SEM||PYX||20040301192350||ADT^A04|ADT757452230|P|2.3||||||||

How to Read an HL7 Message

Message Construction Rules (6/15)



Message Construction Rules (7/15)

Example: Truncate Pattern

- **Source message**

```
MSH|^~\&#|LCS|LCA|LIS|TEST9999|199807311532||ORU^R01|3630|P|2.2
PID|3|2161348473|20923085580|01572633|20923085580^TESTPAT||19730204|
M|||^^^^00000-0000||||||86427531^^^03|SSN# HERE
PV1||I|^802^1||||8625^Physician^Paul P|86-7468^||xxx||||||||V1001
NTE|1|L|Result: NEGATIVE as per the blood sample and no HIV-1 detected , CD4 Percentage: CD4 test normal no of white blood cells count normal
```

In the preceding sample message, the length for the comments field in the NTE segment is defined as 50 characters. After the message transformation, as shown in the following message, the truncation character “#” will be added after the 49th character, which indicates that the message is truncated from there on.

Message Construction Rules (8/15)

Example: Truncate Pattern

- **Transformed message**

```
MSH|^~\&#|LCS|LCA|LIS|TEST9999|199807311532||ORU^R01|3630|P|2.2
PID|3|2161348473|20923085580|01572633|20923085580^TESTPAT||19730204|
M|||^^^^00000-0000||||||86427531^^^03|SSN# HERE
PV1||I|^802^1||||8625^Physician^Paul P|86-7468^||xxx|||||||||V1001
NTE|1|L|Result: NEGATIVE as per the blood sample and no H#
```

Message Construction Rules (9/15)

Abstract Message Syntax

- Schema of the message
- A definition table that indicates the usage of all the segments that may appear in the message legally in a standard HL7 message.
- Reading the syntax :
  - - [] → 0..1
  - - {} → 1..*
- Example
  - - MSH MSA [ERR]
  - - MSH PID PV1 {[ORC]{OBX}}

Message Construction Rules (10/15)

Given the following abstract message definition:

MSH Message Header

MSA Message Acknowledgment

```
[ { PID Patient Identification
    [ { WDN Widget Description
        { WPN Widget Portion }
        [ WPD Widget Portion Detail ]
      }
    ]
  }
]
```

No brackets around it - **Required**
[ ] - **Optional**
{ } - **Repeating**
[{ }] - **Optional Repeating**

Which of the following ordered segments in a message would be ILLEGAL?

A. MSH MSA PID

B. MSH MSA PID WDN WPN WPD PID PID PID

C. MSH MSA PID WDN WPN WDN WPN WPN WPD

D. MSH MSA PID WDN WPN WPD WPD WDN WPN

E. MSH MSA PID WDN WPN WDN WPN WPD

Message Construction Rules (11/15)

Message construction rules

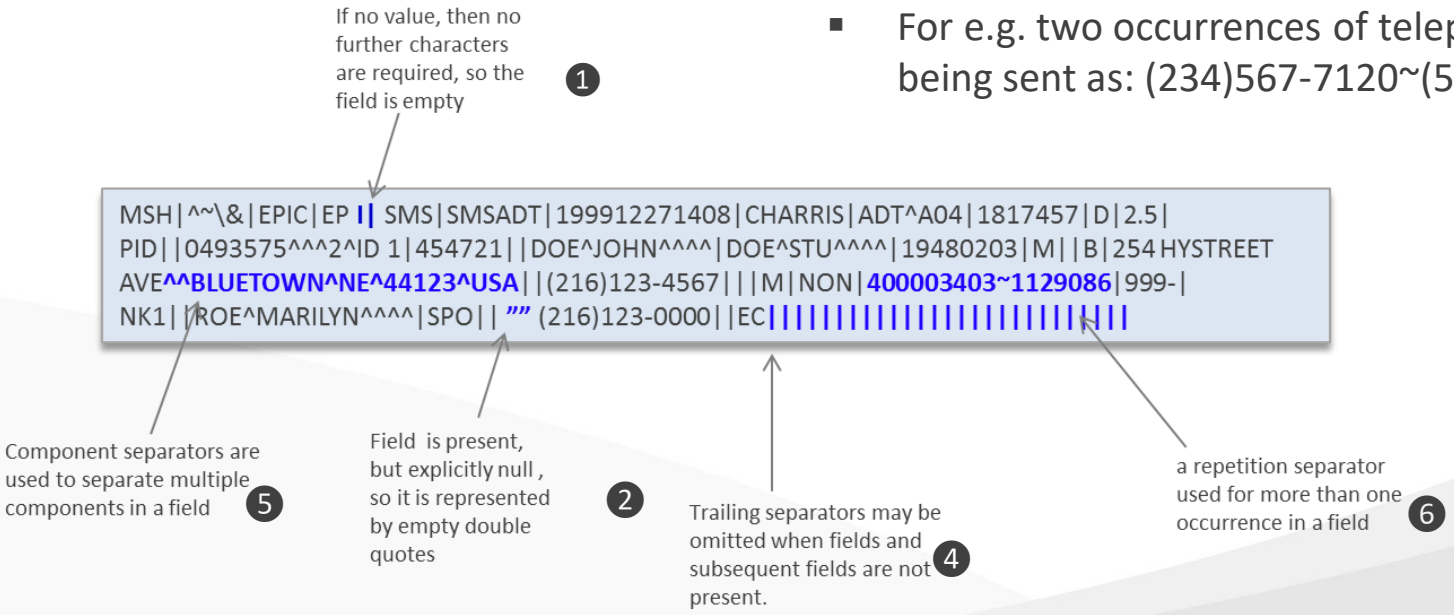
- A message consists of multiple segments.
- Segment construction rules are:
  - Each line in a HL7 message is known as a Segment. It starts with a 3-character textual segment identifier which is always upper case.
  - Ex. "MSH" or "PID" or "PV1", it ends at a ASCII 13 (cursor return) character. ◀
  - The three-letter abbreviation is usually a mnemonic for its purpose. "MSH" stands for "Message Header" and PID for "Patient Identification".
  - Messages should terminate with an ASCII 13 character, ◀ (carriage return character).
  - Precede each field with the data field separator (“|”).
  - Use HL7 recommended encoding characters (“^~\&”).
  - Encode the data fields in the order given in the table defining segment structure.
  - Encode the data field according to its HL7 data type format.

```
MSH|^~\&|EPIC|EPICADT|SMS|SMSADT|199912271408|CHARRIS|ADT^A04|1817457|D|2.5| ◀
PID||0493575^^^2^ID 1|454721||DOE^JOHN^^^^|DOE^STU^^^^|19480203|M||B|254 HYSTREET ◀
AVE^^BLUETOWN^NE^44123^USA||(216)123-4567|||M|NON|400003403~1129086|999-| ◀
NK1||ROE^MARILYN^^^^|SPO|| ”” (216)123-0000||EC|||||||||||||||||||||||||| ◀
```

Message Construction Rules (12/15)

Insertion of fields in a segment

- If the value is not present, no further characters are required
- Data fields that are present but explicitly null are represented by empty double quotes “”
- It is not necessary, and is undesirable, to pad fields to fixed lengths. Padding to fixed lengths is permitted.
- Trailing separators may optionally be omitted. For example, |field1|field2|||| is equivalent to |field1|field2, when field3 and subsequent fields are not present.
- If a field has more than one component, then components are separated with component separators.
- For more than one occurrence of the field, a repetition separator is used.
- For e.g. two occurrences of telephone number are being sent as: (234)567-7120~(599)128-1234|



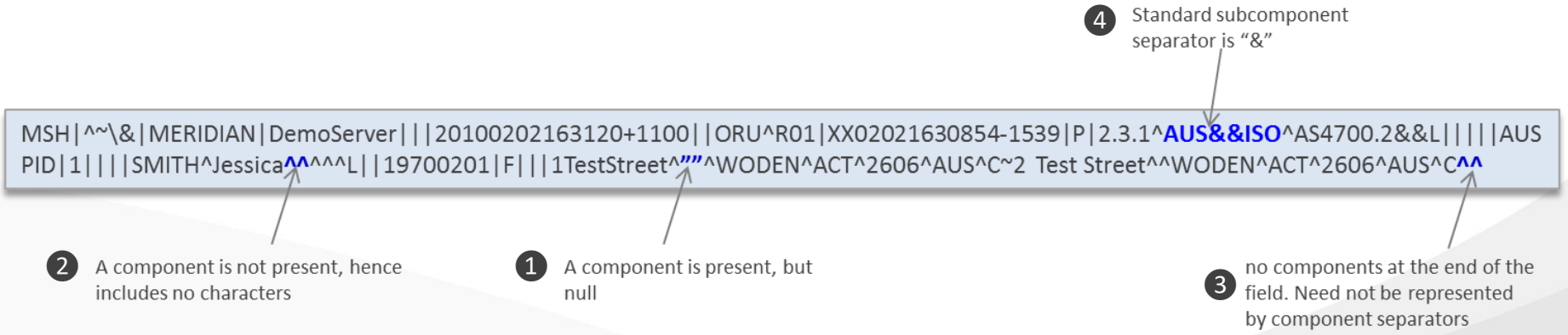
Message Construction Rules (13/15)

Components in a field

- Components that are present, but null are represented by the characters ""
- Components that are not present are treated by including no characters in the component
- Components that are not present at the end of a field need not be represented by component separators. For example, the two data fields are equivalent:

  |ABC^DEF^^| and |ABC^DEF|

- Certain component definition calls for components to be broken into subcomponent, similar rules apply for subcomponents
- Standard subcomponent separator is “&”



Message Construction Rules (14/15)

Rules for receiving HL7 messages and converting their contents to data values

- Ignore segments, fields, components, subcomponents, and extra repetitions of a field that are present but were not expected
- Treat segments that were expected but are not present, consisting entirely of fields that are not present
- Treat fields and components that are expected but were not included in a segment as not present

Message Construction Rules (15/15)

Messages may be locally extended as follows:

- Users may develop local Z messages to cover areas that are not already covered by existing HL7 messages. These must consist of HL7 segments, where possible
- A local Z message may consist of entire Z segments, except that it must begin with an MSH segment
- A local Z acknowledgment message must begin with an MSH segment, followed by an MSA segment, an optional SFT segment and a conditional ERR segment
- Users may develop Z segments and add them to Z messages.
- For example, a ZPD segment could be created to contain customized patient demographics information

```
MSH|^~\&|ADG|HIS|||200701220156||ADT^A08||P|2.3|||
EVN|A04|200501060156|||
PID|||XX3050||ROCHELLE^NEW^BABY||20050106|F|BRIANNA A|W|||||S||718567||XX9050
PV1||O|N57^5703^02|4|||319419^GONZALEZ^JENNY^JE
FFRIES|||PGL||||N||||A||023|||||||||||||||||||A04| |10|||200501060144
ZD1||98.9|160||||X^^^^|^^^X^|^X^|X^^^|X^^^^|120^80 |X^^^^^|X^^^|||||||||||||||||||||||||
ZNI||TEST^VALUE|XXX||||X^^|^X^|^X^|X^^|X^^^|120^80 |X^^|X^^^|||||||||||||||||||||||||
ZOI||TEST|100||||X^^|^X^|^X^|X^|X^^|120^80|X^^|X^^ ^|||||||||||||||||||||||||
```

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- **Use of Escape Sequences**
- Version Compatibility Definition
- Acknowledgment Messages
- Message Control Segments

Use of Escape Sequences (1/2)

- HL7 defines character sequences to represent ‘special’ characters that are not otherwise acceptable in HL7 messages
- These sequences begin and end with the message Escape character (usually ‘\’), and contain an identifying character, followed by 0 or more characters
- Most common use of these escape sequences is to escape the HL7 defined delimiter characters. These delimiters or separator characters are defined in MSH-1 and MSH-2 of the HL7 message

Escape Character	Usage
\F\	Sends the field separator
\S\	Sends the component separator
\T\	Sends the subcomponent separator
\R\	Sends the repetition separator
\E\	Sends the escape character
\H\	Start highlighting
\N\	Normal text (end highlighting)
\P\	Sends the truncation character

Use of Escape Sequences (2/2)

- During the parsing of HL7 messages, special escape sequences should be correctly recognized and converted by interface engines and HL7-enabled applications.
- A simple example of this can be seen in the following OBX segments:

```
OBX||10|||Current Medications
OBX||11|||DILANTIN & NORVASC
```

- The OBX-5 field is defined by HL7 to contain data conforming to an ST data type. What is evident in the message above is the value DILANTIN & NORVASC. The '&' character is typically used as the subcomponent separator in an HL7 message. As per HL7 rules, this message is not appropriately formatted. Since the '&' character in this case is meant to be a part of the resulting text, it needs to be escaped. The correct representation of this message should be as follows:

```
OBX||10|||Current Medications
OBX||11|||DILANTIN \T\ NORVASC
```

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- **Version Compatibility Definition**
- Acknowledgment Messages
- Message Control Segments

Version Compatibility Definition (1/2)

- The encoding rules (for receiving HL7 messages and converting their contents to data values) allow the following definition of a backward compatibility requirement between V2.x versions of HL7
- HL7 standard may introduce new messages in each succeeding version
- Standard may introduce new segments to an existing message in each succeeding version. In general, new segments will be introduced at the end of a message, but they may be introduced elsewhere within the message if the segment hierarchy makes this necessary
- In each succeeding version, the Standard may add new fields at the end of a segment, new components at the end of a field, and new subcomponents at the end of a component; and a non-repeating field may be made repeating

Version Compatibility Definition (1/2)

- If a non-repeating field is made repeating, the first instance of that repeating field must have the same meaning as the non-repeating field had in the prior version of HL7
- For existing fields in existing segments, data types may be changed by the rule mentioned above, if the leftmost (prior version) part of the field has the same meaning as it had in the prior version of HL7
- In other words, if new parts of the field (those that are part of the new data type) are ignored, what remains is the old field (defined by the old data type), which has the same meaning as it had in the prior version of HL7

Version Compatibility Definition (2/2)

MSH|^~\&|EPIC|EPICADT|SMS|SMSADT|199912271408|CHARRIS|ADT^A04|1817457|D|2.5|
PID||0493575^^^2^ID 1|454721||DOE^JOHN^^^^|DOE^STU^^^^|19480203|M||**B|254 HYSTREET AVE^^BLUETOWN^NE^44123^USA**||(216)123-4567|||M|NON|400003403~1129086|999-|
NK1||ROE^MARILYN^^^^|SPO|| ”” (216)123-0000||EC|||||||||||||||||||||||||

The sample message here is HL7 V2.5 message with an address filed (permanent address) which cannot be repeated.

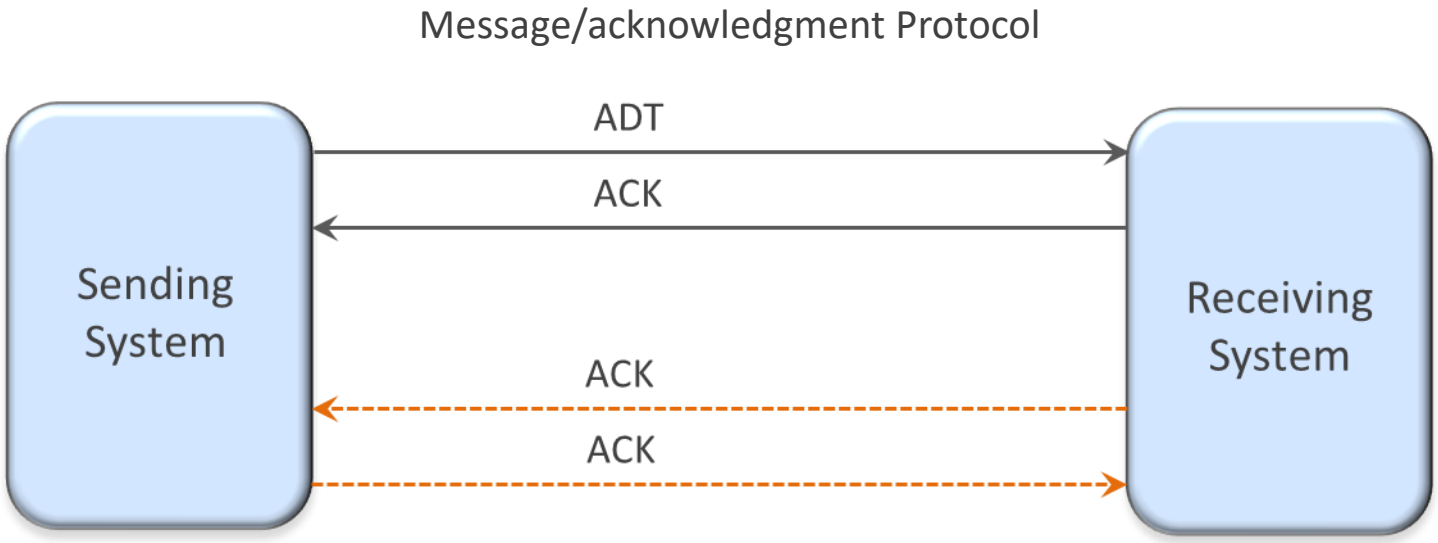
HL7 V2.7 version may introduce a repeating address fields like current address, mailing address, billing address etc.

MSH|^~\&|EPIC|EPICADT|SMS|SMSADT|199912271408|CHARRIS|ADT^A04|1817457|D|2.7|
PID||0493575^^^2^ID 1|454721||DOE^JOHN^^^^|DOE^STU^^^^|19480203|M||**B|254 HYSTREET AVE^^BLUETOWN^NE^44123^USA ~ 100 MAIN ST PO BOX 1022 SEATTLE WA 98104**||(216)123-4567|||M|NON|400003403~1129086|999-|
NK1||ROE^MARILYN^^^^|SPO|| ”” (216)123-0000||EC|||||||||||||||||||||||||

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- **Acknowledgment Messages**
- Message Control Segments

Acknowledgment Messages (1/13)



- Receiver informs the sender whether the message was received and processed successfully
- Acknowledgment are broadly of 2 types
  - Original Mode ACK : Point to point setting
  - Enhanced Mode ACK: Network settings

Acknowledgment Messages (2/13)

```
MSH|^~\&|ImgOrdMgr|UABImgCtr|MegaReg|UABHospC|20070529090142-0500||ACK^A01 |3944441|P|2.5
MSA|AA|01052901
MSA|AA|01052901
```

- MSA segment contains acknowledgment information such as
  - The acknowledgment code (indicating success or failure)
  - The message control ID of the message being acknowledged
  - Optional error segment

Fields of interest in the MSA segment:

Sequence	Name
1	Acknowledgment Code
2	Message Control ID
3	Text Message
6	Error Condition

Acknowledgment Messages (3/13)

HL7 defines acknowledgment rules that allow the receiving system and/or application to give the following information to the sending system and/or application:

Whether the message was received properly.

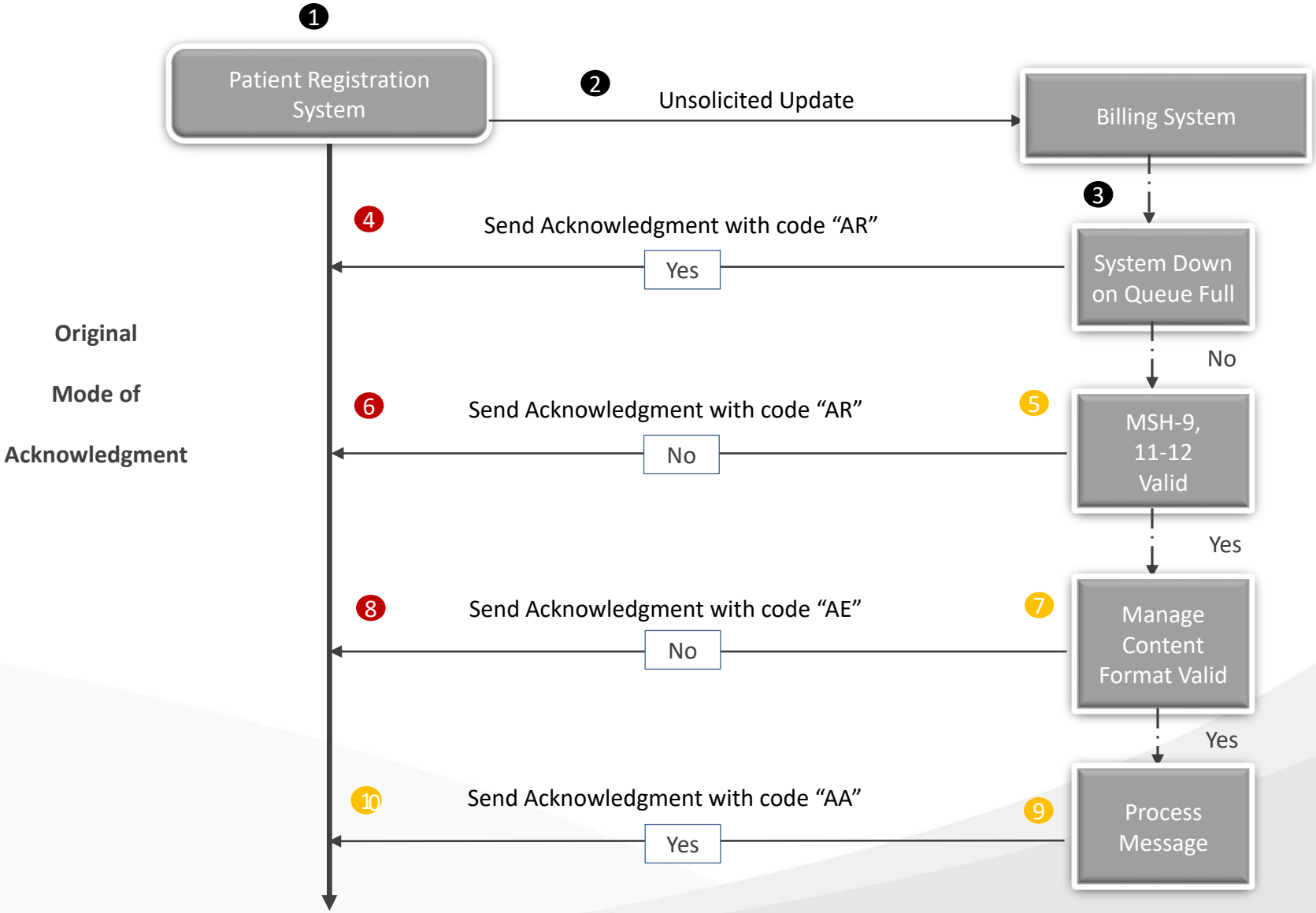
Whether the message was processed properly.

- **Original Mode ACK**: Point to point setting

  When the unsolicited update is sent from one system to another, this acknowledgment mode specifies that it should be acknowledged at the application level.

- **Enhanced Mode ACK**: Network settings
  - Receiving system may send back an immediate accept acknowledgment indicating whether it was able to receive and take custody of the message, without respect to the status of further processing
  - Later on, the receiving application may send back an application acknowledgment indicating whether the message could be processed

Acknowledgment Messages (4/13)



Acknowledgment Messages (5/13)

Following is the processing flow for original mode acknowledgment:

- Originating message should be null, empty for MSH-15 and MSH-16 fields to trigger original level of acknowledgment
- If the originating message has an invalid value for message type (MSH-9), version ID (MSH-12), or processing ID (MSH-11), the receiver returns an acknowledgment message with an acknowledgment code of AR (application reject), and processing is complete
- If the originating message is not processed successfully, the receiver returns an acknowledgment message with an acknowledgment code as follows:

Acknowledgment Messages (6/13)

- AE (application error), if the content or format of the message were invalid (e.g., missing segments, invalid order code). In this case an ERR segment may also be sent

```
MSH|^~\&|RICHARD.SONS
LAB||ADT|767543|20090824828731||ACK|XX3657|P|2.5
MSA|AE|ZZ9380|MSH parsing or validation error: MSH.3.Sending
Application is null|1
```

- AR, if the message processing has failed for some other reason (e.g., system down, queue full)

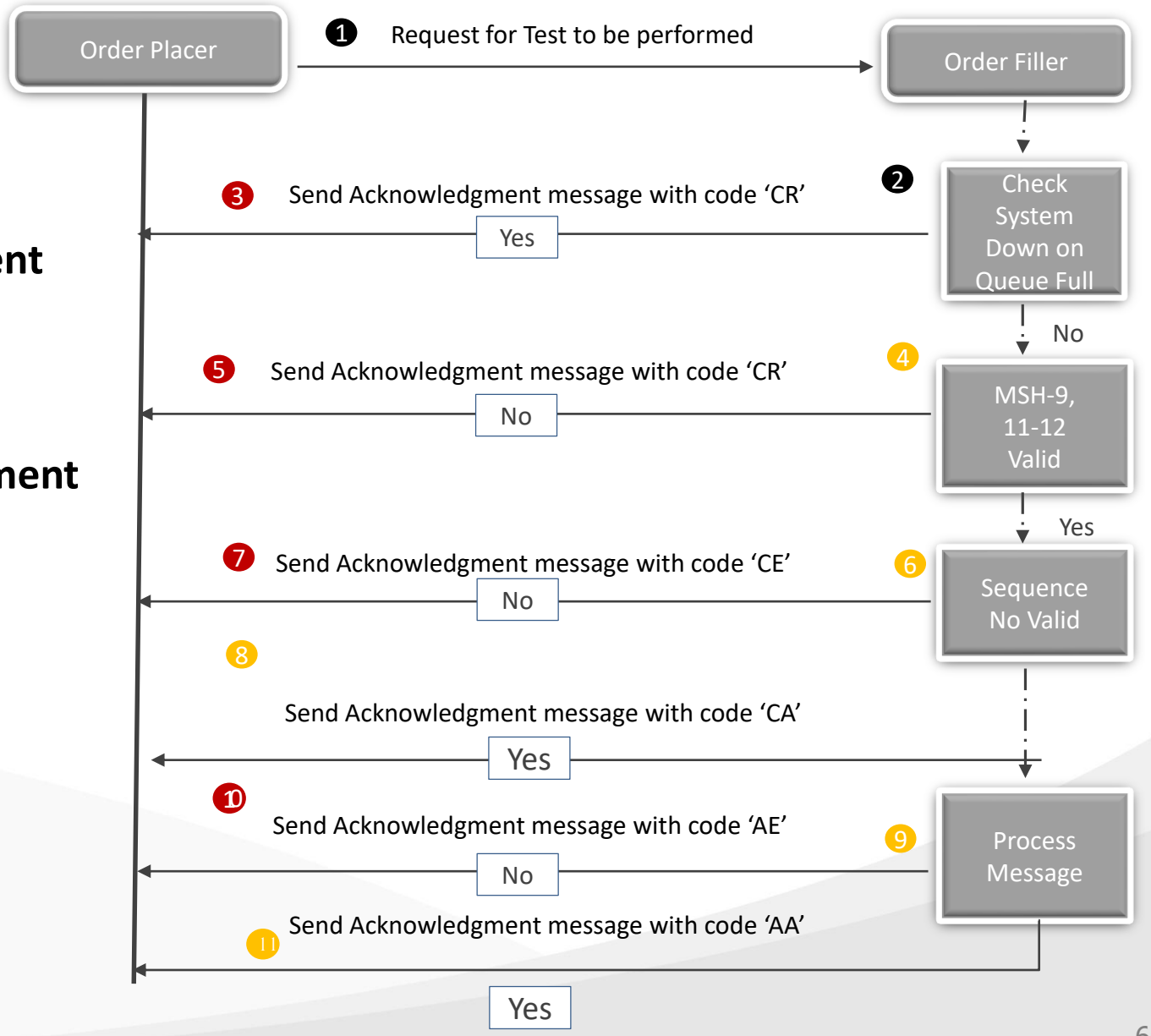
```
MSH|^~\&|RICHARD.SONS
LAB||ADT|767543|20090824828731||ACK|XX3657|P|2.5
MSA|AR|ZZ9380|Error while committing a message into
HCA_HL7.TO.CIM.IN queue|1
```

- Otherwise, if the originating message is successfully processed, the receiver returns an acknowledgment message with an acknowledgment code of AA (application accept), and the processing is complete

```
MSH|^~\&|RICHARD.SONS
LAB||ADT|767543|20090824828731||ACK|XX3657|P|2.5
MSA|AA|ZZ9380||1
```

Acknowledgment Messages (7/13)

Enhancement Mode of Acknowledgment



Acknowledgment Messages (8/13)

The following is the processing flow for enhanced mode acknowledgment:

- Originating message should NOT have null, empty in MSH-15 and MSH-16 to trigger enhanced mode of acknowledgment
- If the originating message has an invalid value for message type (MSH-9), version ID (MSH-12), or processing ID (MSH-11), the receiver returns an acknowledgment message with an acknowledgment code of CR (Commit reject), and processing is complete

```
MSH|^~\&|RICHARD.SONS LAB||ADT|767543|20090824828731||ACK|XX3657|P|2.5
MSA|CR|ZZ9380|Error while committing a message into HCA_HL7.TO.CIM.IN queue|1
```

Acknowledgment Messages (9/13)

The following is the processing flow for enhanced mode acknowledgment:

- If the originating message cannot be accepted for some other reason (e.g., the value of sequence number is invalid), the receiver returns an accept acknowledgment message with an acknowledgment code of CE (commit error) and further description of the error in the text message

```
MSH|^~\&|RICHARD.SONS LAB||ADT|767543|20090824828731||ACK|XX3657|P|2.5
MSA|CE|ZZ9380|MSH parsing or validation error: MSH.3.Sending Application is null|1
```

- Otherwise, the originating message is accepted successfully, the receiver returns an acknowledgment message with an acknowledgment code of CA (commit accept), and processing is complete

```
MSH|^~\&|RICHARD.SONS LAB||ADT|767543|20090824828731||ACK|XX3657|P|2.5
MSA|CA|ZZ9380||1
```

Acknowledgment Messages (10/13)

Acknowledgment Choreography

- Acknowledgment Choreography is defined as the definition of the acknowledgments to be expected for a given trigger event. It is required to document the expected acknowledgment based on the values in MSH.15 and MSH.16
- At the transport level only an ACK should be returned, we should never send an application response message
- The combination of MSH.15=Valued and MSH.16=Blank disallows/does not permit the use of an WPR associated with a prior WPP since the WPR is a response message to an WPP and therefore cannot stand on its own. It needs to be communicated that the exchange paradigm in some profiles where WPP (or equivalent) is sent with MSH.15=Always, MSH.16=Never but an ORP (or equivalent) is used.

WPP - Windows Software Trace Preprocessor message

WPR - Windows Software Trace Preprocessor Response message

Acknowledgment Messages (11/13)

Example of Acknowledgment Choreography

Field name	Field Value: Original mode	Field value: Enhanced mode		
MSH.15	Blank	Valued	Blank	Valued
MSH.16	Blank	Blank	Valued	Valued
Immediate Ack	-	ACK	-	ACK
Application Ack	WPP	-	WPP	WPP

Acknowledgment Messages (12/13)

Points to remember:

- For original mode: Both MSH-15-accept acknowledgment type and MSH-16-application acknowledgment type are null or not present
- For enhanced mode: At least one of MSH-15-accept acknowledgment type or MSH-16-application acknowledgment type is not null
- The original acknowledgment protocol is equivalent to the enhanced acknowledgment protocol with MSH-15-accept acknowledgment type = NE and MSH-16-application acknowledgment type = AL

Note:

AL – Always,

NE – Never

Acknowledgment Messages (13/13)

Field #	Acknowledgment condition values
MSH.15	AL. Select this option if you always want to send accept acknowledgments NE. Select this option if you never want to send accept acknowledgments SU. Select this option if you want to send accept acknowledgments after successful transmission of a message ER. Select this option if you want to send accept acknowledgments only in the event of an error
MSH.16	AL. Select this option if you always want to send application acknowledgments NE. Select this option if you never want to send application acknowledgments SU. Select this option if you want to send application acknowledgments after successful transmission of a message ER. Select this option if you want to send application acknowledgments only in the event of an error

Note:

AL – Always,

NE – Never

Agenda

- Why HL7?
- Information Systems
- Introduction
- HL7 Versions
- Conceptual Approach
- Communications Environment
- Message Framework
- Message Construction Rules
- Use of Escape Sequences
- Version Compatibility Definition
- Acknowledgment Messages
- **Message Control Segments**

Message Control Segment

Segment Category	Segment Name	Description	Purpose
Control	ADD	Addendum segment	Used to define the continuation of the prior segment
	BHS	Batch header segment	Defines the start of a batch
	BTS	Batch trailer segment	Defines the end of a batch
	DSC	Continuation pointer segment	Used in the continuation protocol
	ERR	Error segment	Used to add error comments to acknowledge messages
	FHS	File header segment	Used to head a file (group of batches)
	FTS	File trailer segment	Defines the end of file
	MSA	Message acknowledgment segment	Contains information sent while acknowledging another message
	MSH	Message header segment	Defines the intent, source, destination and some specifics of the syntax of a message
	SGH	Segment Group Header	Indicates that the new segment group begins
	SGT	Segment Group Trailer	Indicates the end of segment group

Message Control Segment

Segment Category	Segment Name	Description	Purpose
General Purpose	NTE	Notes and comments segment	Used for sending notes and comments
	OVR	Override segment	Segment allows sender to override specific receiving application's business rules to allow for processing of a message that would normally be rejected or ignored
	SFT	Software segment	Segment provides additional information about the software product(s) used as a Sending Application. The primary purpose of this segment is for diagnostic use.

Thank You

CitiusTech Markets

CitiusTech Services

CitiusTech Platforms

Accelerating Innovation

CitiusTech Contacts

Email ct-univerct@citiustech.com

www.citiustech.com