Communication protocols and norms in healthcare

Introduction to the HL7 Norm

Course: Biomedical Informatics

Lecturer: Assoc. Ph.D. Miroslav Končar

Email: miroslav.koncar2@fer.hr

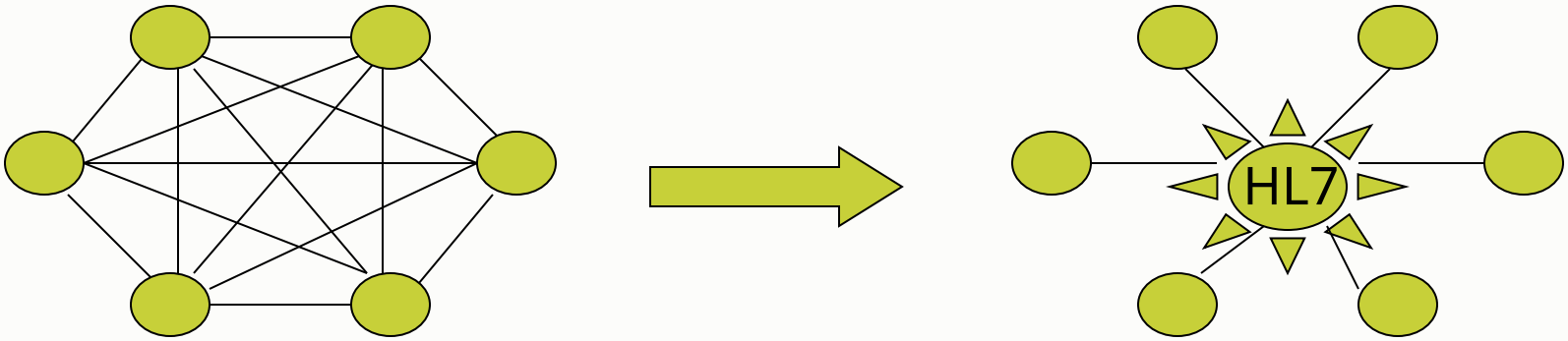
Content

- Basic information about HL7 organization and recommendations
- HL7 Norms
  - HL7v2.x
  - HL7v3
  - HL7 CDA
  - HL7 FHIR
- Conclusion and further motivation

HL7 organization

- HL7 represents the world's leading initiative in the field of interoperability of ICT solutions in medicine
- Mission – "*To provide standards for the exchange, management and integration of data that support clinical patient care and the management, delivery and evaluation of healthcare services*."
- Association founded in 1987 in the USA
- In the meantime, the organization has become global (HL7 Inc), with headquarters in Ann Arbor, USA, and 30+ official branches around the world (including HL7 Croatia)
- www.hl7.org

HL7 motivation



- Reducing the number of required interfaces
  - Point-to-point - N nodes include n*(n-1)/2 interfaces
- Network environment - "HL7 bus"

Communication node (application) in the system

HL7 set of norms

- The most common implementations
  - – Application protocol for electronic data exchange in healthcare systems – HL7v2, v3
  - – Standardization of clinical document architecture (Clinical Document Architecture) – HL7 CDA
  - – HL7 FHIR – API economy, mobile applications, cloud computing
- Other – for your information…
  - - Context Management Component Specifications - CCOW
  - – Normization of knowledge representation – Arden Syntax
  - – Vocabularies and coding systems
  - - …

HL7 - basic information

- HL7 standard – enables the exchange of clinical and administrative data between distributed application
- Application layer
  - Responsible for exchanging information between two network applications.
  - Functions such as security checks, identification, availability check, and the most important instantiation of the information change itself

7. Application layer	***HL7***
6. Layer presentations	***XML***
5. Session layer	***Web Services/ebXML (profiles, security)…***
4. Transfer layer	***Network functions (transfer information)***
3. Network layer	
2. Data layer	
1. Physical layer	

OSI Composition

HL7v2.x

- The first significant advances towards the goals set by the mission
- Simple and efficient application in hospital environments, especially when it comes to administrative functions
- Implementations
  - ADT (Admission, Discharge, Transfer)
  - Ordering laboratory tests and associated processes
- Significantly accepted by the market and industrial suppliers - today by far the most widespread norm in the medical ICT technology segment in application
- Latest version – v2.9 (September 2019)
  - IMPORTANT - unless otherwise stated, we use the specifications in the rest of the material**HL7v2.5.1**

HL7 messages

- **HL7 Message**is a key concept for understanding the HL7v2 standard
- An HL7 message is a rounded set of information that is exchanged between applications and systems
- The HL7v2 message is described using the syntax found in the norm in the so-called**Abstract Message Syntax**boards
  - Part of the standard specification itself
  - Defines the construction of the message, segments, their sequence and formatting
  - HL7v2 message = n(Segments) = n(N*(message fields - Message Fields))
  - Fields in the message (message fields) are defined by data types, which can be simple with a single value, or complex with multiple components

HL7v2 message syntax

- Sending a message is related to **Trigger Event (TE)** – an event in the real world that results in some communication in the form of messages. Examples:
  - Patient admission or visit (A01)
  - Patient transfer (A02)
  - Patient discharge (A03)
- Each message has its own **Message Type (MT),** which defines the category to which the message belongs.

  For example:

  - ADT (Admission, Discharge, Transfer) - a group of messages related to the admission, discharge and transfer of a patient
  - ORU (Unsolicited transmission of an observation message) - a group of messages related to sending the results of observations about the patient's condition
  - OML (Laboratory Order Message) – a group of messages related to sending the processing of patient samples

HL7v2 message syntax

- The combination of TE and MT uniquely defines the syntax of the message we send. We write the name of the message MT^TE
  - For example, ADT Â01 is a unique code for the patient admission message, and as such we will find it defined in the standard in the corresponding Abstract Syntax Table
- TEs are specific for each MT, which means that the connection between MT and TE is n:1.
  - One TE has a connection with only one MT
  - MT can have a connection with one or more TEs

HL7v2.x - Segments

• An HL7 message consists of a series of segments, which are actually a logical group of fields (Message Fields).

• Each segment in the HL7 standard is identified with a unique identifier (ID of 3 characters)

– For example, an ADT message can contain the following segments: Message Header (MSH), Event Type (EVN), Patient ID (PID), and Patient Visit (PV1)

• Segments in the message:

– Mandatory or optional (optional are marked with [ ] brackets)

– Repeating, one or more (marked with { } brackets)

– A segment can be both optional and repetitive – in that case it is marked with {[ ]} The order of brackets is not important

• The segment attribute table describes the fields in the segment and their use. The fields are described by the following attributes:

– SEQ : Position of the field in the segment

– LEN : Normative maximum field length (C.LEN : field length for checking compatibility (conformance))

– DT : Field data type

– OPT: optionality

– RP/# : repeating field, i.e. the number of allowed repetitions

– TBL# : table identifier with field values

– ITEM# : internal identifier in the HL7 standard

– Element Name : human readable name of the field

HL7v2.x - Segments

- Segment groups – one or more segments can be organized into logical groups
  - Each group of segments has its own name (but not a code!), as an identifier that does not change
  - A group of segments, similarly to individual segments, can be optional/mandatory, and can be repeated.
  - As of v2.5, the first segment in a group is always mandatory
  - Segments within a group can be nested
- A message can also have a case where the segment that follows *one of the options offered*. Then the syntax is marked with brackets and delimiters as follows (example): <OBR|RQD|RQ1|RXO|ODS|ODT>

Possible values for field optionality in segment (HL7v2.5.1)

Code	Definition	Explanation
R	*Required*	Mandatory field
ON	*Optional*	Optional
C	*Conditional on TE, or some other field*	Depending on TE itself, or some other field. The dependency must be clearly described in the table in the standard
X	*Not used with this TE*	Not supported with the specified TE
B	*Backwards compatibility*	Left for compatibility with previous versions
W	*Withdrawn*	Withdrawn from use

Example - MSH segment definition (English Segment definition table)

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	1	ST	R			00001	Field Separator
2	4	ST	R			00002	Encoding Characters
3	180	HD	O		0361	00003	Sending Application
4	180	HD	O		0362	00004	Sending Facility
5	180	HD	O		0361	00005	Receiving Application
6	180	HD	O		0362	00006	Receiving Facility
7	26	TS	R			00007	Date/Time Of Message
8	40	ST	O			00008	Security
9	13	CM	R		0076/ 0003	00009	Message Type
10	20	ST	R			00010	Message Control ID
11	3	PT	R			00011	Processing ID
12	60	VID	R		0104	00012	Version ID
13	15	NM	O			00013	Sequence Number
14	180	ST	O			00014	Continuation Pointer
15	2	ID	O		0155	00015	Accept Acknowledgment Type
16	2	ID	O		0155	00016	Application Acknowledgment Type
17	3	ID	O		0399	00017	Country Code
18	16	ID	O	Y	0211	00692	Character Set
19	250	CE	O			00693	Principal Language Of Message
20	20	ID	O		0356	01317	Alternate Character Set Handling Scheme
21	10	ID	O	Y	0449	01598	Conformance Statement ID

MSH segment (v2.5.1)

HL7v2.x

Message Fields

- A field is actually a string of characters
- A field in a message can exist in one of three states
  - Populated – contains information/content
  - Not Populated – does not contain information/content
  - Null – explicitly communicates the value zero (" ")
- Special characters – according to the table below

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 fields where allowed.
Repetition Separator	~	2	Separates multiple occurrences of a field where allowed.
Escape Character	\	3	Escape character for use with any field, component, or sub-component represented by an ST, TX or FT data type.
Subcomponent Separator	&	4	Separates adjacent subcomponents of data fields where allowed.

HL7v2.x data types

- 89 different data types (DT)
- Basic division – simple and complex types
- Simple data types are those types that carry a unique value, while complex types are a set of sub-elements where each has its own data type and definition.

- Examples of simple data types
  - DT - Date in YYYYMMDD format
  - DTM - Date and time including time zone
  - ST – text data up to 200 characters
  - TX – text data up to 64k data

HL7v2.x data types

- Complex data elements are associations of data that logically belong together
  - Codes and identifiers
  - Names and addresses
  - Other complex types
- Example: HD - Hierarchic Designator – Identifies an administrative object (system or application) that has the possibility of assigning unique identifiers.
  - It is used to define the sender and receiver of the message.
  - If the first field is present, the second and third are optional. If the third component is present, then the second must also be present. The second and third components together must be present, or both NULL
  - As an example, described onhttps://hl7-definition.caristix.com/v2/HL7v2.5.1/DataTypes/HD

Field	Length	DT	Optionality	Repeatability	Tables
HD.1 - Namespace Id	20	IS (coded value for user defined tables)	ON	-	0300
HD.2 - Universal Id	199	ST (string)	C	-	
HD.3 - Universal Id Type	6	ID (coded value for HL7 defined tables)	C	-	0301

Example of HL7v2.x Abstract Message Table

- Patient admission for hospital care - Patient Admit (ADT^A01)
  - ADT - message type
  - A01 – trigger event
- Segments appear in the order listed in the table
- Mandatory segments
  - MSH - Message Header (all messages start with MSH!)
  - EVN – Event Segment - additional information about TE
  - PID – Patient Identification Segment
  - PV1 – Patient Visit – information related to the patient's visit (assigned doctor, type of visit, similar)
- Several groups of segments
  - Search request
    - Timing and quantity
    - Search results
    - Sample
  - Insurance information

ADT^A01^ADT_A01 ADT Message	Status	Chapter
MSH	Message Header	2
[{ SFT }]	Software Segment	2
EVN	Event Type	3
PID	Patient Identification	3
[PD1]	Additional Demographics	3
[{ ROL }]	Role	15
[{ NK1 }]	Next of Kin / Associated Parties	3
PV1	Patient Visit	3
[PV2]	Patient Visit - Additional Info.	3
[{ ROL }]	Role	15
[{ DB1 }]	Disability Information	3
[{ OBX }]	Observation/Result	7
[{ AL1 }]	Allergy Information	3
[{ DG1 }]	Diagnosis Information	6
[DRG]	Diagnosis Related Group	6
[{ --- PROCEDURE begin		
PR1	Procedures	6
[{ ROL }]	Role	15
}] --- PROCEDURE end		
[{ GT1 }]	Guarantor	6
[{ --- INSURANCE begin		
IN1	Insurance	6
[IN2]	Insurance Additional Info.	6
[{ IN3 }]	Insurance Additional Info - Cert.	6
[{ ROL }]	Role	15
}] --- INSURANCE end		
[ACC] Accident Information 6		
[UB1] Universal Bill Information 6		
[UB2] Universal Bill 92 Information 6		
[PDA] Patient Death and Autopsy 3		

Legend

Required segments

Segment Groups

Example of an HL7v2 message

MSH|^~\&|GHH LAB|ELAB-3|GHH OE|BLDG4|200202150930||ORU
R01|CNTRL-3456|P|2.4<cr>

PID|||555-44-4444||EVERYWOMAN EVE E^^^ L|JONES|19620320|F|||153
FERNWOOD DR.^ STATESVILLE OH 35292||(206)3345232|(206)752- 121|| ||
AC555444444||67-A4335^OH^20030520<cr>

OBR|1|845439^GHH OE|1045813 GHH
LAB|15545 GLUCOSE|||200202150730|||||||| 555-55-
5555^PRIMARY^PATRICIA P^^^ MD^^|||||||||F|||||444-44-
4444^HIPPOCRATES^HOWARD H^^^^MD<cr>

OBX|1|SN|1554-5^GLUCOSE^FAST 12H CFST:MCNC:PT:SER/PLAS:QN||

182|mg/dl|70_105|H|||F<cr>

HL7v2.x communication - transfer confirmations messages

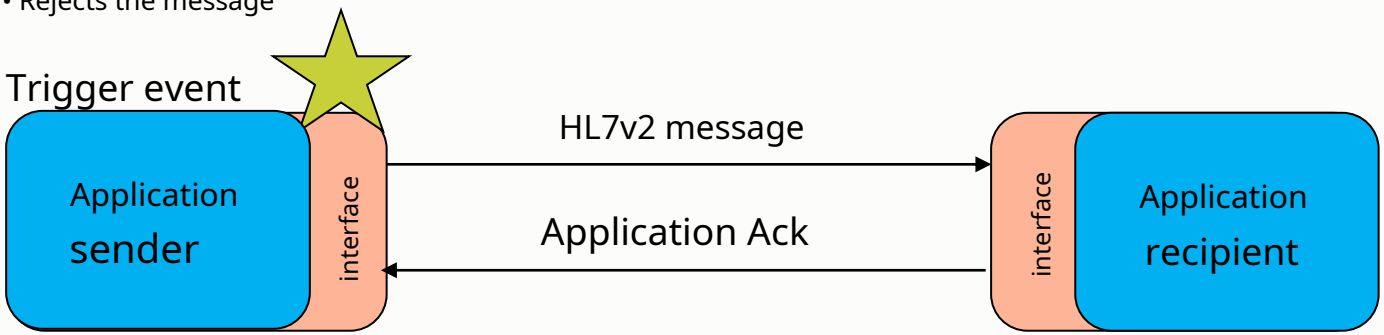
- The dynamics of information transmission, and the handshake between the interface of the sender and the receiver - the way in which they mutually confirm the receipt of the message
- Two levels of confirmation
  - Application Acknowledgment – confirmation at the application level that the message has been received, and that the process can continue (MSH-16 field)
  - Accept Acknowledgment – confirmation that the message has been saved in the database, and that the sender does not need to send the message again (MHS-15 field)
- MSH-15 and MSH-16 fields can have values from table 0155
- Based on these messages and specifications, we are talking about two types of certificates - original (eng.*Original)* and reinforced (eng.*enhanced)* way

HL7 Table 0155

Value	Description	Comment
AL	Always	
NE	Never	
ER	Error/reject conditions only	
SU	Successful completion only	

Original Mode

- **Original mode: MSH-15 and MSH16 fields are either NULL or not present**
- After sending the message, actions on the receiver's side are as follows
  - The destination interface checks:
    - Which TE and MT are we talking about, and whether the interface supports the same – MSH-9
    - Processing ID is acceptable – MSH-11
    - Message version is acceptable – MSH-12
  - If any of the above conditions are not met, the system rejects the message
  - If all conditions are met, the message is transferred to the application, which then processes the message. The result of the process is
    - Sends a confirmation that everything is fine
    - Sends confirmation that an error has occurred
    - Rejects the message



HL7v2.x Confirmation Levels – cont'd

Enhanced Mode

- **At least one of MSH-15 and MSH-16 is not NULL**

• The concept of two levels of confirmation is introduced:

– Accept Acknowledgment (MSH-15 field)

• The message has been received and saved in a secure repository

• Interface and storage status, syntax are checked

• MSH-9, MSH-11 and MSH-12 are validated as in Original mode

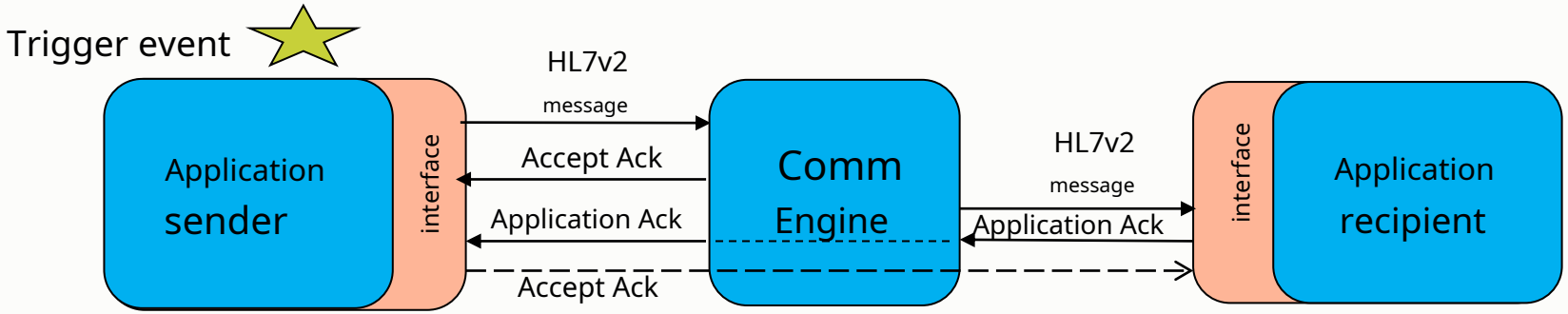
• In practice, this means that some other system has downloaded the message, and that the original application that generated the message should not send it again.

– Application Acknowledgment (MHS-16 field)

• The message is processed by the destination and logically taken over by the same.

• Acknowledgment is sent as the start of a new process, which may again require its own Accept Acknowledgment (MSH-15). Application Ack (MSH-16) is however always NULL or NOT (Application Ack cannot have its own App Ack)

• Original and Enhanced modes are practically the same if the MSH-15 field has the value NE(VER) and the MSH-16 field has the value AL(WAYS)



Confirmation of receipt

• To confirm receipt, the HL7v2 protocol defines**Acknowledgment message (ACK)**

• When the application does not have a specific response to the basic message, that is, when it is an error, we use the so-called**general**ACK messages. Otherwise, for each message (MT^TE), the norm will specify exactly what the response looks like

• The segments of the general ACK message are listed in the table below

Segment	Optionality	Possibility repetitions	Definition
MSc	R	-	Message header
SFT – Software Segment	ON	∞	Additional definition of the SW that sends the message
MSA – Message Acknowledgment	R	-	Information related to message confirmation
Error - Error	ON	∞	Additional information in case of error

MSA Segment

- Contains key information related to message confirmation, and is the only mandatory segment along with MSH
- Ack Code is the first field in the confirmation, and we use codes from the table (Table 008) to transfer information to the sender

MSA Segment

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	2	ID	R		0008	00018	Acknowledgment Code
2	20	ST	R			00010	Message Control ID
3	80	ST	B			00020	Text Message
4	15	NM	O			00021	Expected Sequence Number
5			W			00022	Delayed Acknowledgment Type
6	250	CE	B		0357	00023	Error Condition

Board 008

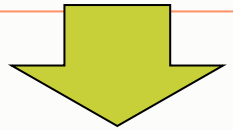
Value	Description	Comment
AA	Original mode: Application Accept - Enhanced mode: Application acknowledgment: Accept	
AE	Original mode: Application Error - Enhanced mode: Application acknowledgment: Error	
AR	Original mode: Application Reject - Enhanced mode: Application acknowledgment: Reject	
CA	Enhanced mode: Accept acknowledgment: Commit Accept	
CE	Enhanced mode: Accept acknowledgment: Commit Error	
CR	Enhanced mode: Accept acknowledgment: Commit Reject	

Example – Original way

MSH| ~\&|LABxxx|ClinLAB|ICU||199109180605 44||MFN M03^MFN_M03 | MSGID002|P| 2.5 <cr>

MFI|LABxxx^Lab Test Dictionary^ L|UPD|||AL

MFE|MUP|199109051000|199110010000|12345^WBC^L OM1|... MFE| MUP|199109051015|199110010000|6789^RBC L^OM1|...



MSH| ~\&|ICU||LABxxx|ClinLAB|199109180605 45||MFK M03^MFK_M01 | MSGID99002|P|2.5 <cr>

MSA|AA|MSGID002

MFI|LABxxx^Lab Test Dictionary^ L|UPD|||MFAA MFA|MUP|

199110010000|199110010040|S|12345^WBC^L MFA|MUP|

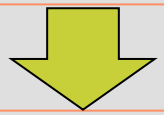
199110010000|199110010041|S|6789 RBC L

Example – enhanced mode

MSH|^~\&|LABxxx|ClinLAB|ICU||199109180605 44||MFN M^03|MSGID002|P|2.5|||AL|AL MFI|
LABxxx L^ab Test Dictionary^L|UPD|||AL <cr>
MFE|MUP|199109051000|199110010000|12345^WBC L^
OM1|...
MFE|MUP|199109051015|199110010000|6789 R^BC^ L
OM1|...

MSH|^~\&|ICU||LABxxx|ClinLAB|199109180605 45||MSA|MSGID99002|P|2.5 <cr> MSA|
WHAT|MSGID002

MSH|^~\&|ICU||LABxxx|ClinLAB|199110010805 04||MFK|MSGID5002|P|2.5|||AL <cr> MSA|AA
|MSGID002
MFI|LABxxx L^ab Test Dictionary^L|UPD|||MFAA MFA|MUP|
199109051000|199110010040|S|12345 ^WBC L^MFA|MUP|
199109051015|199110010041|S|6789 R^BC^ L



MSH|^~\&|LABxxx|ClinLAB|ICU||199110010805 07||ACK|MSGID444|P|2.5 <cr> MSA|
CA|MSGID5002

Localizations and extensions

- Messages and information that are not covered by the global norm, and address local requirements - popularly called**With messages/ segments/TE**
- Local Z messages
  - Users can extend and define new local Z HL7v2 messages that are not covered by the standard
    - It is recommended to use existing segments wherever possible
  - A local Z message may consist of all Z segments, except that all messages must use the first MSH segment as prescribed by the standard, including local Z Acknowledgment messages
  - Users can develop Z segments and add them to Z messages
  - Users can develop Z segments and add them to HL7 messages. The TE can remain the same if the basic intent of the message is the same
- Adding existing HL7 segments to existing HL7 messages is not recommended

Localizations and expansions – cont'd

- Users have the ability to develop their own Z TE trigger events
- Groups of Segments - localization in the form of creation of local groups from existing individual segments, i.e. "de-grouping" of existing groups is not allowed
- Segments and individual fields
  - Users may not modify existing segments
  - Local fields may be used in local segments, although using existing fields is strongly recommended
  - Extending existing segments with local fields is not prohibited, but not recommended
- Extending data types
  - Locally defined data types can be used in local fields, although using existing data types is strongly recommended
  - Pre-defining existing fields is not allowed
  - DTs can be locally extended if necessary, resulting in the creation of a local Z data type

HL7v2.x rules

- Message processing
  - Segments, fields, components and sub-components and additional repetitions of fields that are not expected are ignored
  - Segments that are expected and not present are interpreted as containing all empty fields
  - Fields and components that are expected within the segment and are not included are interpreted as not being present
- Complexity of communication
  - Delayed confirmations
  - Sequential sending of messages (for database synchronization)
  - Fragmented sending when dealing with large messages
  - Sending messages in a batch
- General rules
  - The HL7 standard does not assume that the data is stored in the database in any of the applications
  - The HL7 standard does not set any requirements on data ownership

HL7 version 2.x - problems

- The process of creating 2.x messages is completely ad hoc
  - There is no explicit methodology
  - There are no formal instructions for the construction of messages
  - Most fields in the message are optional
- Dynamic model – application responsibilities, confirmations of receipt
- Result => interoperability of different HL7v2.x implementations not guaranteed and practically impossible!!!
- Consequence => Development of other versions of the norm (in future lectures)

Conclusion

- HL7 is the world's leading standardization initiative in the field of medical informatics and ICT technologies in that industry

- HL7v2 is by far the most common standard in the industry
  - Ease of access
  - Wide availability of implementation tools
  - History and number of installations
- In the following lectures:
  - HL7 v3
  - HL7 CDA
  - HL7 FHIR

Resources and Literature

- All information is available at www.hl7.org
- Norms are available free of charge, with prior registration
- HL7 Croatia – www.hl7.hr
- HL7v2.5.1. specifications:
  - https://hl7-definition.caristix.com/v2/HL7v2.5.1
- Open source or freeware HL7 SW implementations
  - HL7 HAPI - https://hapifhir.github.io/hapi-hl7v2/
  - NextGen Connect (ex Mirth) - https://www.nextgen.com/productsand-services/integration-engine .