HIPPA – Comes in 90s, it’s basically to save the privacy of the patient records and everything around the healthcare.

EMR/EHR/HID – Examples Electronical Medical Record

EMR Is the Centre of everything.

EMR application available in market:-

Allscripts : Best for large organization

Athenahealth : Best for reporting

Cerner : Best for Multiple site organization

NextGen : Best for population health management

DrChrono : Best for medical billing

Kareo : Best for small practices

ICD – International Classification Diseases :-

It is basically representation of a code of specific disease and this has been managed by WHO (World Health Organization).

CPT – Similarly for any procedure we do there is a code associated with it.

LIS – Lab Information System :-

1. It manages a lab operations and patient testing data.
2. LIS receives orders from the EHR and returns results.
3. Laboratory Professionals uses LIS to coordinate workflow and quality control of testing.
4. An LIS tracks clinical details and test results for patients. And then stores the info in its database.
5. These include conventional radiology, CT, and ultrasound, as well as images from patients MRIs, and pathology

Example :- Quest, Sunquest and Atlas lab

RIS – Radiology Information System :-

1. RIS is a software system that is used to efficiently manage the process of obtaining, storing and sharing medical imagery in a healthcare context.
2. A RIS is most commonly used in conjunction with PACS to track images, manage image archive.

Example :- GE PACS, Philips PACS (Picture Archiving and Communications System)

Payer VS Provider :–

Payer – Our Insurance or who pays our medical bill.

Provider – Who provide a healthcare like a Doctor, Nurse and Receptionist or Pharmacist all those people.

ADT, ORM & ORU –

Integration / Interface Engine - HL7 interface engine is a central hub that receives, routes, and sends HL7 messages to the appropriate healthcare information systems.

To Communicate two application which build on other code base or other application system.

Application 1 – XYZ.java

Application 2 – PQR.net

So, to communicate both application we follow common communication language which HL7.

For to communication we required a Integration Engine.

XYZ.java ------------- HL7 ------------- PQR.net

Tanmay Patil ------------- Mirth ------------- Patil Tanmay

Communication happens in many ways as bellow;

* TCP/IP, FTP, Web API’s

Examples - Available Integration Engines in Market

Mirth Connect, Rhapsody, Ensemble, Core point

What is an Integration Engine?

1. An integration engine is a software solution used in healthcare interoperability to enable exchange of health information between disparate healthcare systems, applications and data sources.
2. Integration Engine serves as central hub for healthcare data exchange, providing the necessary tools and capabilities to translate, transform, route and validate health data between different systems.
3. Integration engine can handle various healthcare data standards such as HL7, DICOM, X12 and FHIR, as well as custom file formats.
4. They can also supports different communication protocol such as TCP/IP, HTTP, FTP and SMTP to facilitate secure and reliable data transmission.

The Goal of an Integration Engine?

* The primary goal of using the integration engine in healthcare interoperability is to achieve seamless connectivity between different healthcare systems and application, which is crucial for delivering quality patient care.
* By using an integration engine, healthcare organizations can streamline their clinical workflows , automate data exchange and reduce manual error and redundancies, ultimately Improving patient outcomes and reducing costs.

Why do we need an Integration Engine?

NextGen Connect(Mirth) Integration Engine?

What is Inbound Message?

Any message coming from outside to inside of our mirth channel we name it as inbound message.

In Inbound Message we have also various types;

1. Raw :- The message we get it from data source we call it as raw inbound message.

Why raw because it’s in the original form of the message because we did not

do any modification, transformation on to this message that we receive from

the data source. So that we call it as raw inbound message.

What is Outbound Message?

If we are sending any message from our channel to outside of mirth to downstream applications.

Why we have Filters and transformers in destinations as well?

* Lets say we have different destinations connector types, one sending to TCP/IP connector with IP and PORT. Another is sending, creating a file and storing that file into a SMTP server of a different destination system. And at destination 3 we are storing that information into the database.
* For each Destination connector comes with its own dedicated filtration and transformation.
* For ex. For destination 1 we don’t want to send ADT messages they just want receive ORU messages even though we are doing on each type of messages at source. But here for destination 1 we don’t want to send ADT.

What is Encoded Data?

1. We receive Encoded data to the destination from the source connectors whenever message we gets after transformation to the destination is called as Encoded data.
2. The message we received after transformation from the source Outbound to the destination is called as Encoded data.
3. The transformation we will be doing on processed/filtered raw message, that transformed message it will be stored as transformed data and that outbound/transformed data is called as Encoded data.
4. And the Encoded data from the source connector is used by each destination connector as its raw data.

What is Response Data?

After sending data/message to the destination a response is received by the destination connector and stored it as response data.

What is METADATA?

Whenever we receive any message from the source system and from that message what values we need keep into our mirth channel, we call it as METADATA. Or we want to show the data on dashboard metadata table we call it as Custom METADATA.

Custom Metadata ?

To add the columns in metadata table in channel.

Custom metadata columns are user created columns that shows up in Custom METADATA table.

Durable Message Delivery ?

* Right now our channel is in active state but if something happened our channel went down due to connection issue or something. But it received 5 messages during that period, now this durable message delivery make sure that those 5 message we received are going to get delivered to that downstream application. The operation will not get halt in between those messages will not stuck on the channel.
* Our channel went down, there are 5 messages that got stuck, until you comes back and fix your issue. You enable the channel back again until you reprocess those 5 messages again manually they are going to just stuck on channel.
* It is enabled based on the selected message storage settings . If enabled, unfinished messages will automatically be recovered and processed if the channel gets halted or if the channel suddenly goes down.
* If it is set to **Reprocess only**, unfinished message will not automatically reprocessed, but we still have an option to manually reprocess those messages from message browser.

Prune metadata ?

* It means that we won’t able to fetch messages from channel after given period of days or months. Or we can’t pull the messages older than given months.
* Archiving is available for given period of days or months.
* It will be applicable for both Metadata and Content in Message Pruning section.

Edit Response ?

* Basically when you received a message and that message you have send to a destination.
* Now when you send a message to the destination.
* Once the message received at destination they will send a acknowledgement back to you.
* When you received that acknowledgement you can process that acknowledgement as well in the edit response tab.

Variable Maps:-

Response Map :- Continue from Edit Response.

* When you received acknowledgement from the destination, you can process that acknowledgement as well. For to process that response/acknowledgement we use Response Map.
* Widely it is not used. Below maps are used.

Connector Map :-

* Each destination is a connector, when you create a connector map the scope of the connector map is within that connector itself.
* We cannot access the map which is created by connector map in destination 1 to destination 2 or other destination as well.
* Let’s say I created a connector map in source connector, if I try to access that connector map I can’t be able to access connector map to the any of destination.
* This map is isolated to the current message.
* The Scope of the Connector Map is within the Current channel.
* If we can store a Connector map variable in Destination1, We will not be access that value in Destination2.

Channel Map :-

* Channel map scope is entirely to the channel itself in which channel map is created.
* When I created a channel map anywhere in the channel, it can be accessible within that same channel.
* Means we can access that channel anywhere in all destination and on the other transformer everywhere but within same channel.
* So the scope of channel map within the channel itself, we can’t be use that channel map in another channel.
* This map is isolated to the current message.
* The channel map is useful for anything that needs to be shared among multiple destinations.

Source Map :-

* Source map is similar to connector map, but basically a connector map you can create on destination as well, but source map is within the source connector.
* This map is isolated to the current message.
* This map is read only. For example File Reader will automatically inject the "Original File" variable.

Global Channel Map :-

* So the global channel map also the scope of the map within that same channel but another significance is that it can accessible to another message but within the same channel.
* But the difference is global channel map can be access within different message. It cannot be accessible to another channel.
* This Map is isolated to specific channel, But across multiple message.
* That means We can store value during message processing and it will be available during the lifecycle of the next message.
* But in case of above 3 maps message value can stored but not shared among other message over lifecycle.

Global Map :-

* Global map is global to all the channels and any map is created in global scripts or anywhere in the channel it is global to all the channels.
* If I change the value of global map in the channel it will be affected across multiple channel until I deploy that channel. So till time it will show same previous value.
* This map is available across your entire server, across all channels and all messages.
* That means we can store a value during message processing in one channel, and use that value from different channel.

Configuration Map :-

* This map we can add it in the settings, basically configuration maps are used for IP, PORTS, passwords.
* So for this configuration map there is no any deployment required, the values are affected across multiple channel where it is used.

Links :-

https://www.youtube.com/watch?v=zeMPry3ak6Y

https://www.youtube.com/playlist?list=PLd0LQoDCLunHTcHlz1CV19mSf29HMZJNF

Difference between Global Map and Configuration Map :-

Global Map :-

* After deploying channel, we changed global map value until and unless we again deploy that channel it will show same value when the channel was deployed.
* To get changed global map value affected on the channel you have to redeploy that channel.

Configuration Map :-

* But in configuration map that is not the case. We don’t need to deploy again that channel.
* Whenever we change the value it will be affected without deploying that channel this is the advantage of configuration map.

Difference between Global Map and Global Channel Map :-

Global Map :-

Global Map is accessible across all the channel.

Global Channel Map :-

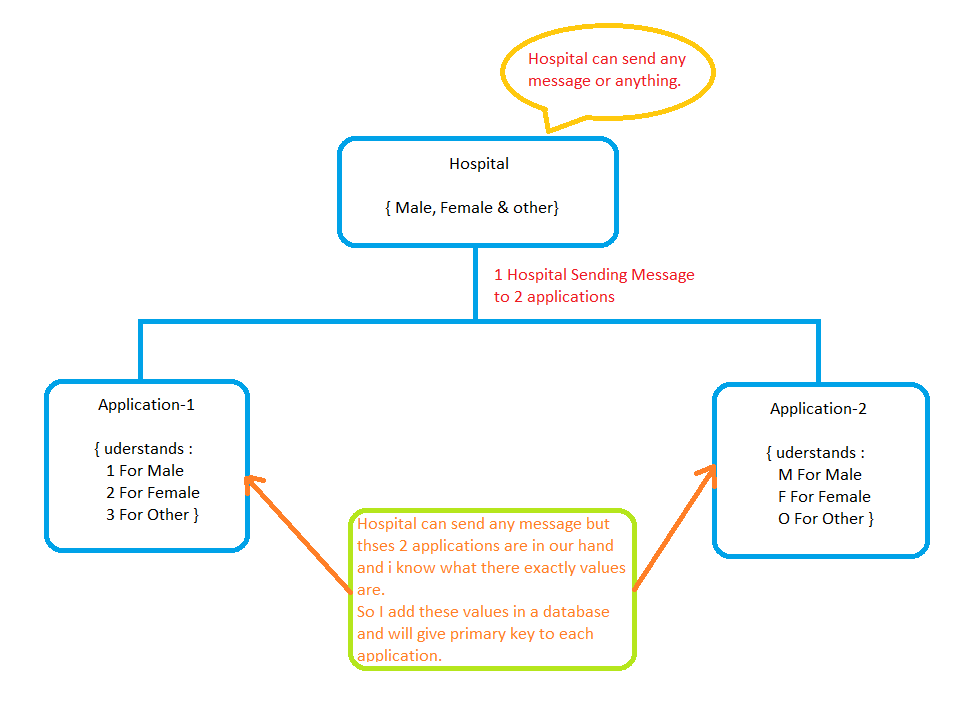
* Global Channel Map is applicable to the channel in which it created.
* We can’t access global channel map in another channel.
* The life span of that map is till you deploy the channel again.
* Let’s Say I added global channel map count variable and we are incrementing it.
* So whenever we deploy that channel it will start from beginning till you re deploy that channel.
* If you again deploy that channel it will start counting from beginning.

What is the purpose of Variable Maps?

* It is just as variable. Let’s say pid3 is the MRN, so you want that pid3 accessed in destination or anywhere else as well.
* Let’s say you change the tmp object and you have change the MRN value in the message and you want change to original value of that MRN you can store it in the channel map and we can access that channel map in the destinations.
* Suppose you want some variables to be defined, let’s say you created a channel for 10 sites and for each site has different name.
* And you want all site names to be there and you are writing a code depending on which site it is and you want to change that MSH3 value.
* So I will create a different channel maps and assign hardcoded MSH3 value In that channel maps before sending it each destination on particular channel have 3 different sites to send.
* So I created 3 channel maps in source transformer and I replace that value in each destination site before sending that message.
* Whatever variables are created in source that are available in destination transformer available variables tab.

Maps are used just advanced the filtration part in mirth and How widely we use these maps?

* A site is sending some set of values let’s say gender. For the different sites gender may have male, female and other things.
* But we sending values to a different application or someone else, they may not understand male female so for they might understand m, f some kind of things or some people can also use 1 for male, 2 for female, 3 for other.
* So for to convert that value we need to understand that what type of application is it and we have to defined some variables there to identify that application.



Code Templates

Functions :-

In Mirth we call functions as code templates.

Why we need code templates in Mirth ?

* A code template is a function of code that can be used across multiple channels
* A code template library is a group of code templates that is linked to one or more channel.
* When code template library linked to a channel, all code templates in the library will be available to the channel.

Interview questions

* Variable maps, Metadata, code templates
* What is the required segment for all HL7 messages?

Any type of message but one segment is required without that HL7 cannot be processed.

So without MSH HL7 message cannot be processed.

* For ADT required segments are MSH, PID and PV1.
* What is Accession Number?

Basically it is OBR3 we have

* Placer order number or filer order number ORU and ORM.
* OBR25 order status.
* ORC, OBR and OBX.
* Most common ADT message you see ADT08

What version worked on mirth?

Version 4.2

What version worked on hl7, what difference between v2.3 and v2.5.1?

Most notably - these changes are required for different certifications, government policies, etc. An example of this is having a specified field for CLIA certification information, bar code scanning fields for more granular tracking of laboratory samples, etc.

How to upload flat file and csv in mirth?

1. Create channel and give name to channel
2. Click on source tab, select connector type as File Reader
3. Give path into the directory in file reader settings.

How to make use of lookups in mirth?

Instead of calling out a specific map, use generic lookup function

var value = $(‘variableName’);

or we can reference a variable using Velocity Variable Replacement

$(variableName);

Why we do this, NextGen connect Integration Engine will automatically look that key up in all available maps. That may only be $cfg / $g or it may be all maps but in sequence wise

Location of filler id in HL7

OBR3

Location of place order id in HL7

OBR2

How to filter specific messages type in mirth and where to apply rules. How to code using rule builder and JavaScript?

1. Inside the source click on Edit Filter
2. After went Filter Add New Rule
3. We have types are available in type section as drop down
   1. For Rule Builder, There are three options are available such as Field, Condition and Values.
   2. For JavaScript, There blank page will open to code and apply condition in it.

How to update values using transformers?

1. Inside the source click on Edit Transformer
2. We get filtered processed data to transformer from filter
3. After went to Transformer Add New Step
4. After adding step we have drop down under type section
   1. For JavaScript, There blank page will open to code and apply condition in it.
   2. For Message Builder, There are some fields are available such as Message Segment, Mapping, default value and string replacement.

How to convert ORU to MDM message type?

MDM is Medical Document Management

How attachment works in HL7?

An attachment is a piece of data extracted from the raw incoming data messages and stored separately. It will be stored once, multiple atts will not be stored.

An extraction happens at the very beginning of the message life cycle, even before the pre-processor script runs.

When a destination connector dispatches data outbound, any attachment associated with the message will be automatically re-inserted in to the outgoing message.