‹ Home

Exported Sep 09, 2020 at 12:58 PM

HL7 FHIR eReferral Workstream

+103 others

Messages | Campfire | To-dos | Docs & Files | Check-ins | Schedule | Forwards

Added by Austin Irvine • Updated Jun 13, 2018 at 12:02 PM

/Communication

20% EARLY DRAFT

Overview:

Having the ability to for bi-directional communication between two systems can greatly increase the efficiency of the referral process. The Communication resource can be used to ask questions regarding a referral, request a referral be updated with missing information, or request additional information not covered by the specific referral dataset.

Constraints: The communication endpoints should be treated as a post/thread mechanism and not be used as a channel to transfer case related data (for further discussion). This should be done via the ServiceRequest resource. This will help ensure that changes to a referrals data are distributed properly for external systems.

Out of Scope: Patient communications, Notifications, Alerts

Links:

- https://simplifier.net/guide/eReferraldraftiGuide/Communication
-

Change Requests

- Add HealthcareService as a communication.recipient

Relevant chat.fhir.org Threads:

- Communications are tied to a specific referral with **.basedOn**. There are only two directions a communication can go, referrer > provider, and provider > referrer. How do we define the direction of the communication?
- Should the **recipient** have the option of healthcareService as a target resource?
- Changing payload.contentString --> payload.contentCodeableConcept

Notes:

- none

Profile

- Communication
  - **identifier [0..*] (Identifier):** Unique identifier
  - ~~**instantiatesCanonical [0..*] (canonical)**~~
  - ~~**instantiatesUri [0..*] (uri)**~~
  - **basedOn [0..*] (Reference):** Request fulfilled by this communication (i.e., related serviceRequest)
  - ~~**partOf [0..*] (Reference):**~~
  - **inResponseTo [0..*] (Reference(Communcation)):** Reply to another communication
  - **status [1..1] (code):** preparation | in-progress | not-done | suspended | aborted | completed | entered-in-error (always submit a communication as completed)
  - ~~**statusReason [0..1] (CodeableConcept):** Reason for current status~~
  - **category [0..*] [CodeableConcept]:** Message category
  - **priority [0..1] [code]:** Message urgency (Request priority valueset) (most messages expected to be "routine")
  - ~~**medium [0..*] (CodeableConcept):** a channel of communication~~
  - ~~**subject [0..1] (Reference(Patient)):** Focus of message~~
  - **topic [0..1] (CodeableConcept):** Description of the purpose/content
  - ~~**about [0..*] (Reference(Any)):** Resources that pertain to this communication~~
  - ~~**encounter [0..1] (Reference(Encounter))):**~~
  - **sent [0..1] (dateTime):** When the system sending the communication records the transmission time (sent with communication)
  - **received [0..1] (dateTime):** When the system receiving the communication records the transmission time (not sent with communication, updated by receiving system)
  - **recipient [0..*] (Reference(PractitionerRole))**: Message Recipient
  - **sender [0..1] (Reference(PractitionerRole)):** Message Sender
  - ~~**reasonCode [0..*] (CodeableConcept):** Indication for the message~~
  - ~~**reasonReference [0..*] (reasonReference):** Why was communication done?~~
  - payload [0..*] (BackboneElement): Message payload
    - content[x] [1..1]
    - contentString (string)
    - contentAttachment (Attachment)
    - ~~**contentReference (Reference(Any))**~~
  - ~~**note [0..*] (Annotation)** - Comments made about the communication~~

Important Decisions & Rationale:

- **Use Communication.basedOn to connect every communication to a ServiceRequest**. Communications do not exist independently of a ServiceRequest.
- **Always submit Communication.status as completed**, the other statuses are not relevant. Consider also using "entered-in-error" to revoke a message.
- **Default priority value is "routine"**
- **Don't use Communication.medium**, because it is intended for when using the Communication resource as a means to *create a record* a conversation/phone-call/etc... rather than using the Communication resource itself as the medium of communication.
- **Communication.subject is not used** because the Communication is about the ServiceRequest (via Communication.basedOn), not the patient.
- **Communication.topic ... usage is TBD**. It needs a custom valueset.
- **Communication.about not used** because it is redundant with .basedOn for eReferral purposes.
- **Communication.recipient** is not anticipated to be used in most cases, as the "ServiceRequest" is conceptually the recipient of the Communication. Communication.recipient would be used only when the sender wants to explicitly call out a specific participant in the referral process, when there are many participants. Note that we may need a way to generally specify the direction of the Communication, i.e. if it is intended for the referral SOURCE or the referral RECIPIENT.
- **Communication.sender** is mandatory for data attribution purposes (note that if the integration is a user-level integration rather than a system-level integration, this field is not required as the sender data in implied by the user authentication mechanism)
- **Communication.payload can only have 1 contentString, and multiple Attachments**:
  - when sending JUST a text message, use payload.contentString
  - when sending JUST an attachment, use payload.contentAttachment
  - when sending a message with an attachment(s), use multiple payloads; one for the text message (payload.contentString) and additional .payload entries for each attachment (payload.contentAttachment).
  - Do *not* include multiple text messages (payload.ContentString) in one Communication
- The following fields are anticipated to have **variable support between systems** and are included more for future-proofing reasons, it is not recommended to rely strongly on them at this time:
  - inResponseTo
  - priority
  - status
  - topic
  - recipient
- All messages posted to a ServiceRequest should be visible by all people who have access to view the ServiceRequest itself. i.e., entire Circle of Care is able see all messages

Special Discussion: Request for Information Process

In important workflow in managing referrals is the RFI (Request for Information) process, where the referral target asks the referral source for more information about the referral (such as an assessment document, a medical history, etc...).

There are two steps to this process, the RFI request, and the RFI response.

Consider the following approaches for RFI requests:

1. **Informal RFI via Communication**: Do not treat RFIs any differently than a regular communication process. A communication can be sent with a message (in Communication.payload.contentString) such as "Please send me the most recent AssessmentX for this patient". The recipient of this message can respond with another message that provides the necessary information.

   PRO: This method simply leverages a core communication process, with no need to program additional functionality specifically for RFI
   CON: It cannot handle enforcement or tracking of responses to RFIs.
   BOTTOM LINE: This simple method can handle a basic implementation of RFIs and will be compatible with any system that can handle the eReferral Communication process. However, it does not have enough structure for more robust RFI implementations needed in the marketplace.

2. **Formal RFI via Communication:** Use the standard communications process, but add a indicator/flag that the communication is specifically intended for a RFI process. There are a number of potential ways to indicate this:
   - Set MessageHeader.eventCoding = "Request for Information"
   - Set MessageHeader.reason ="Request for Information"
   - Use Communication.topic = "Request for Information
   - Add an extension to Communication called "responseRequired"

   PRO: Simple to implement on top of standard communication process, while adding structure that would can work towards more workflow like tracking of RFIs & requiring responses.
   CON: Somewhat "hack-y" approach, where Task is designed specifically for use cases such as this. Some systems process RFIs completely different than communications. How to track in FHIR that an RFI was "complete"?
   BOTTOM LINE: This is a simple-to-implement solution that would likely cover all current use cases, but is not quite standard "FHIR-y" in nature, and may not future proof against more advanced uses of Task resource in the future.

3. **Formal RFI Via CommunicationRequest:** Use the "CommunicationRequest" resource to request an RFO.

BOTTOM LINE: While this may seem like an intuitive use of CommunicationRequest, it is actually fairly outside of the intended purpose of CommunicationRequest which is meant to make a record of when you ask somebody to make a Communication on your behalf to another entity. It is recommended to discard this alternative.

4. **Formal RFI via Task:** Create and send a task resource with Task.code = "Request for Information". The Task.description would include any narrative for the request. The Task would connect to the referral via Task.basedOn. This task would be associated with a new message event also called "request-for-information", with the referral requester (PractitionerRole) assigned as the Task.owner.

   PRO: Most robust approach that fit with the different approaches to RFIs that different systems have. Leverages advanced FHIR "Task" functionality. Should be future proof as more potential "tasks" could be assigned to a referral with future workflows. Can be implemented independent of a communications functionality.
   CON: More complex to implement than other options.
   BOTTOM LINE: This is the most functionally whole solution, but also the most complex to implement.

Consider the following approaches to RESPOND to requests for information (note that the request and response methods can to some degree be mixed and matched):

1. **Respond with an information Communication** - Respond simply with a communication. Based on the content, the reader is to understand whether or not the tasks was completed in a non-structured way.

   PRO/CON/BOTTOM LINE: Same as "Informal RFI via Communication"

2. **Response with a formal Communication**: similar to "Formal RFI via Communication", respond with a Communication and add a flag of (using similar alternatives as "Formal RFI via Communication") to indicate that it is fulfilling an RFI. Use "Communication.inResponseTo" to indicate which RFI is being fulfilled.

   PRO/CON/BOTTOM LINE: Same as "Format RFI via Communication"

3. **Respond by updating the ServiceRequest:** The expected response to an RFI could be for the request to update the initial ServiceRequest.

   PRO: Well aligned with the existing workflow of some vendor systems. Required updated data can slot directly into its most relevant spot in the ServiceRequest.
   CON: Assumes that the request can always be fulfilled by modifying the serviceRequest, what about other scenarios like: a) information is provided offline, b) information can be provided via a communication. Also, note that in eReferral specification there is not *yet* functionality to allow the sender to update their ServiceRequest after initial submission.

BOTTOM LINE: This approach is fairly robust for many use cases, but leaves out use cases that can be handled offline or via communications, and requires the fairly advanced and not-yet-included-in-the-specification functionality of the requester updating the ServiceRequest after initial submission.

4. **Respond by updating the Task:** When the RFI has been completed, the person who completes it can update the task with a completed status and send a ". This can be independent of any other action, thus allow the method used to fulfill the request to take any form (i.e., send a communication, update service request, respond with a phone call, send a letter, conversation, mail documents, etc...).

   The completed Task resource would be sent with MessageHeader.eventCoding = "Response to Request for Information".
   Variation: if the tasks is completed in a communication or update Service Request message, include the completed task in those messages (however, this would not cover the case of when a response is provided offline).

   PRO: Robust, flexible, aligned with FHIR philosophies, strong process/workflow tracking potential, can be implemented independent of Communications functionality
   CON: Two messages instead of one would need to be sent when responding to an RFI (e.g., "send Communication" message + "Response to Request for Information"), more complicated than other options to implemented if communications is already implemented.
   BOTTOM LINE: This method handles all use cases fully, but is somewhat more complicated to implement than other options.

Recommendation:

- Use the "Task" method for both sending the RFI, and for indicating that the RFI has been fulfilled. Though somewhat more complicated to implement than other alternatives, this is the most robust, flexible and scaleable approach.
- Some systems may choose to use "Informal RFI via Communication" methods for both request and response. This is not in conflict with the recommended "Task" method, rather it is a simpler approach that uses the existing functionality of the Communications process.

Comments & Events

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
Here is a preliminary version of the communication profile as discussed on the working group meeting:

https://www.screencast.com/t/ObFzlcj6
Aug 21, 2018 at 1:06 PM · Notified 92 people

Alfred Wong, VP of Engineering, Think Research

My Thoughts

Here are some of my thoughts about how we should be treating the communication resource. My lens on it, is definitely more from an acute care and long term care lens. It's has taken me much longer to think through a lot of the use cases thus the other resources are taking longer. I feel there could be other ones as well but I will add to this as I go

There are other things we need to take into consideration as well, especially when you look at the landscape of what is happening to HIS systems in Ontario. Also if you look at BC where they have majority of their systems in Cerner and Meditech where they could have a more integrated system as well.

The use cases below are based off of current day systems.

Please feel free to comment or make suggestions. I know there already has been a first stab at this resource but below are my thoughts.

Based on the current definition of what a Communication resource is suppose to represent ("This resource is a record of a communication.") We need to make sure we're not removing too much out of the resource to support the creation of the record.

Use Case 1 - Hospital to Internal Clinic

Patient A is admitted to hospital A for a fracture in their pinky. Hospital A also has an internal physiotherapy clinic (Clinic B) to help patients recover from their injuries. The system Clinic B uses is a separate system but has a tight integration between hospital A's system and has the ability to retrieve information about a patient and the encounter.

Patient A is sent to the x-ray department to verify the broken pinky and is then given a finger splint to treat the injury. Patient A is discharged and is then referred to Clinic B to rehab the pinky. The physiotherapist had some questions about the diagnostic images and needed answers from the original physician who treated Patient A. Furthermore, nothing in the discharge report was able to answer physiotherapist's question. Through Clinic B's system, a communication request is sent to Hospital A's system to ask for more information about the x-ray for the pinky. The original physician answers the question and sends the communication back to the physiotherapist through Hospital A's system.

Use Case 2 - Hospital to LTC Site

Hospital A is part of a larger network of hospitals and long term care sites. More specifically, 3 acute care centers and 2 long term care sites. They all share the same instance of a HIS / EMR but is segregated by facility (i.e. they can't see each other's patient information, encounters, etc...) and often refer acute care patients to their 2 long term care sites.

An elder aged man has come into Hospital A with a severe medical condition that impaired his day-to-day lower body movement. Hospital A treated his condition but recognized that he will need a short term stay at one of their long term care facility to help with his rehabilitation before going home. Hospital A then proceeds to discharge him and refers him to one of their long term care facilities. LTC Site B receives the referral but before admitting him they wanted to understand more of his medication history, his diagnostic imagery that came with the referral bundle. Once that communication occurs and satisfied with the MRP's explanation to their question they admit the elder man into the facility.

Attributes

- **identifier** - We should keep this as it falls into the same problems of how does a system know which communication resource they are talking about. For example, Hospital A's system has an internal identifier for that communication resource of 1234. When we send this resource over to Clinic B's system, it will be important to keep track of what the identifier is to update Hospital A's communication resource of 1234.
- **partOf / about** - We should keep one of them because this will be useful for use case 1. Being able to specify the DiagnosticReport as what this communication is about is important as that is different from the concept of the Communication.context attribute. There can be other uses here not just for the DiagnosticReport but other important resources as well such as Observation, MedicationStatement.
- **subject** - We should keep this to support the ability to provide more context about the patient, especially in a more integrated settings like a hospital or long term care site.
- **context** - We should keep this to support the ability to provide more context about the patient's encounter or care plan in a more integrated setting like a hospital or long term care site. Being able to go back to the original encounter as a reference point is important for healthcare professionals.
- **reasonReference / reasonCode -** We should keep this as another purpose of this resource is to keep a record of the communication and this will help create a record of why the communication was created.

Aug 25, 2018 at 8:54 AM · Notified 91 people

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
My thoughts on these communication elements:

.identifier: Possibly, however there are a number of other ways to identify it as well. In particular, the resource .id element, and the .basedOn to connect it to the serviceRequest

.partOf: I struggle with the difference between this field and "**.basedOn**", as they seem to be pretty close in intention of what they mean. Might be worth asking on chat.fhir.org. Also, in the use case you describe, is the x-ray inherently part of the ServiceRequest (referenced via .basedOn) anyway?

.subject: Subject is already referenced in the ServiceRequest under .basedOn, which is why it was left out on the initial pass on this profile. To include it here too would be duplication (in the context of eReferrals). There may still be a reason to include it though for clarity purposes... I'm unsure.

.context: I don't believe is currently supported by any of the eReferral vendors. May be better to include in the "for future use" bucket?

.reasonCode: The valueSet is very very specific and clinical. In the .basedOn > ServiceRequest we capture reason, but chose to disregard the coding due to over-specificity. In reality, people tend to freehand their "reason for Referral". **ReasonReference** feels like it runs into the same problem as basedOn vs. PartOf vs. ReasonReference. Do we really need all 3 of these, what is the fundamental difference in meaning between them (as it relates to eReferrals)?
Sep 04, 2018 at 9:28 AM · Notified 91 people

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
We have been doing some investigation on how to send "Communications" (i.e., messages between eReferral requesters and providers about the referral, e.g., a "Request for more information") back and forth over the API. There is a fundamental architecture question we are debating that we would like some feedback from the community on:

When sending a communication, there are two methods we are considering:
1) Send ONLY the newest communication
2) Send EVERY message whenever a new message is added

#1 is a lighter and cleaner implementation
#2 is a bit more robust, as it allows to still follow the full conversation in the payload, allows to check for communications that may have been dropped, but requires a bit more implementation work (i.e., to check for matching communication IDs, pull out the most

recent)

I'm curious what other systems/approaches are used in the world of messaging for this, and what the preferences out there are?
Nov 07, 2018 at 11:01 AM · Notified 91 people

ion moraru
I am in favor of #1 option. In response to an RFI request the recipient could provide an update to the eReferral or send a document. The RFI request could thus be fulfilled out-of-band with the conversation initiated by the RFI request. Having the full conversation with every message is helpful if the request is expected to be resolved in the conversation. I also assumed that the "communication" message in this case is not expected to change the eReferral status (i.e. the Source would like to cancel the initial referral).
Nov 07, 2018 at 6:01 PM · Notified 91 people

Ted Jin, Architect 2, Integration and Solutions Architecture
My preference would be #1 with the assumption that referral messages are exchanged in a federated API environment. I would also assume that "basedOn" and/or "partOf" elements can always reference to the originating system so that a message recipient will be able to view the full referral history when needed. #2 option, on the other hand, may result in bulky message communication and unnecessary data being transmitted over wires.
Nov 08, 2018 at 2:46 PM · Notified 91 people

Alex Goel
I agree that option #1 is easier if we're assuming that communication should be like mail, but I think it really depends on the use case. Right now referral messages seem like they are isolated. Is there value in doing option #2? I think there may be if we want these messages to work more like email, because it can provide context to whoever is getting the message. Regarding Ted's point I think it depends on how unnecessary/necessary this information would be for patient care.

We don't want to overwhelm users with information, but we don't want to persist existing problems of not providing information readily.
Nov 08, 2018 at 3:11 PM · Notified 91 people

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
There is a bit of a email vs messaging paradigm at play here.

A few notes:

- the communication includes a reference (basedOn) back to the original referral always.
- This scenario is for pushing communications to other systems. Presumably the application would be responsible for identifying the relevant messages to show in option 2
- a GET Communication (for all Communications related to a specific referral ID) can also be available to “catch up”

It sounds like the general consensus seems to be #1, with just the most current message.
Nov 08, 2018 at 3:19 PM · Notified 91 people

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
Note that even in case 1, I think it is expected that each system keep/persist the historical messages that were sent, and can choose to present that historical context as per their own interfaces.

The main benefit I think of option 2 (I believe) is more on the question of data redundancy and robustness, in the case of a previous message not getting through it can still all be seen from the most recently retrieved message. This may be overdoing it though and there may be better options to deal with this particular angle if the problem (e.g. resending messages, queues, pub/sub, etc...)

-Tim Berezny
CTO & Co-founder, Caredove
Tim@caredove.com
705-717-6359
Nov 08, 2018 at 3:32 PM · Notified 91 people

BK

Branko Koprivica, Architect 2
If we implement the e-* initiative like this one by using block chain, the communication would pretty much be OOTB, i.e. in order to retrieve a communication a local API call would suffice. The block chain would distribute all communication already, across all peer nodes' ledgers and in an orderly fashion. HL block chain, for example, uses Kafka for ordering transactions.
Nov 08, 2018 at 3:56 PM · Notified 91 people

AW

Alfred Wong, VP of Engineering, Think Research
I agree Option 1 is easier from a technical perspective and keeps the size of the payloads smaller and more manageable.

I agree with Tim that each system should be keeping a record of the Communication resource as they receive it. At that point, the history should be available locally so then

what is the point of option 2?

The questions that I've been asking myself around this issue is:

1. Is the problem we are trying to solve here is so that external systems that have access to the communication need a historic view of what was exchanged between the two parties? If it is, then how would this external system get that information? Would they just query all Communications using .based_on using the ServiceRequest as a filter?
2. We're using a pretty simple example on how a communication request would be used but I wonder if this would break down for more complicated one?

For example, a use case that often happens in a mental health settings for seniors is the following:

- A family doctor sends a referral to mental health.
- The central intake system receives it but does not get enough information and they often ask for a full history for the patient. They need the patient's (3 things):
  - best possible medication history (BPMH),
  - lab work
  - Any diagnostic information (DI)
- Within the central intake system they send a Communication resource for this information
- The family doctor receives this ask and then responds with all 3 pieces of documentation for the mental health worker.
- Other things to consider, the mental health worker may have more questions around the medication history or a question around some lab work.

Maybe we can use this as an example to work through our thoughts on how best to fulfill this using our current assumptions. I know we don't want to complicate things but I worry we may not support some of these other cases.

My initial thoughts / questions are:

- Introduce a MessageHeader type for communication-request, communication-response. Where the response type will hold information on which communication it refers to.
- Would it be three different communication resources for each piece of information or is it just one?

One Communication Resource Sent

- How does the requesting system know that all three of the asks have been completed?
  - The responding system would create three Communication resources for each ask (BPMH, Labs, DI) referencing the original ServiceRequest with the .based_on field and using the .part_of for the original Communication request.
  - The requesting system would then receive all three Communication resources and use the .part_of to know which Communication it is tying back to.

Three Separate Communication Resources Sent

- Each BPMH, Labs, and DI would get their own Communication resource
- The responding system would respond individually
- The requesting system would receive all three Communication resources and know which asks are being fulfilled.

- Do we use the CommunicationRequest resource to kick off the process to help group and focus the Communication resources around a specific ask.
  - One CommunicationRequest is sent for BPMH, Labs, DI and is .based_on
  - The responding system will create a three communications based_on the ServiceRequest but is part_of the CommunicationRequest.
  - Similar to the One Communication Resource Sent, the requesting system would receive the individual response but know which CommunicationRequest it is for.
  - Currently, the FHIR spec says this is referenced by other things but not the ServiceRequest which makes me wonder why or maybe I don't understand the full scope of this CommunicationRequest resource?
  - I totally see this overcomplicating things but this may help my question 1 above as an external system could get all communication resources around the initial request

I recognize that I could be mixing up the initial intent of what we are trying achieve here with a simple Communication workflow. I'm just wondering how do we add a bit more robustness to the flow to ensure certain communication resources are being responded to.

Definitely longer then I wanted and I'm not saying any of these are the answer. I just want to understand what we are trying to solve and start that discussion. Hopefully I can be on the next call to discuss.

AG 👏

Nov 10, 2018 at 12:38 PM · Notified 91 people

Matt Atwood
Attaching my rough notes regarding the Communication resource based on our discussions up to this point.

CommunicationResourceNotes.docx 124 KB • Download

Jun 03, 2019 at 10:47 AM · Notified 91 people

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
I have done a deep dive on the communication resources and completed the following:

1. I have fully profiled the Communications Resource
2. I have completed a detailed analysis of all potential methods (that i could think of) to do a request-for-information process.

Still todo is to create a sample Communication resource, and sample "send communication" message.

My analysis on the RFI process summarizes all of the approaches that we have talked around on the calls over the past month, and more. After analysis, I'm pretty confident that using the Task resource is the way to go (with a caveat that leveraging the proposed informal communication process can optionally be used as well).

You can find the analysis on the Communication Resource Basecamp page, near the bottom titled "Special Discussion: Request for Information Process".

GR Terrific!

Jul 11, 2019 at 6:10 PM · Notified 88 people

Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group
I have posted the question from our discussion today about Communication.payload.contentString change to Communication.payload.contentCodeableConcecpt on chat.fhir.org, here:

https://chat.fhir.org/#narrow/stream/179166-implementers/topic/Communication.2Epayload.2EcontentString.20change.20to.20CodeableConcep

Jul 23, 2019 at 1:57 PM · Notified 89 people