FHIR Workflow

# Comments on the FHIR Ballot September 2016

Vassil Peytchev, Epic

## Current ballot content

The September 2016 FHIR ballot contains the new “Workflow module”, which attempts to specify workflow management within the FHIR specification. As noted in the ballot, “[i]nteroperable support for workflow execution is actually a more advanced FHIR activity because it requires a higher degree of standardization.”

The Workflow module, in my opinion, fails to achieve this higher degree of standardization. It mixes various concepts and the related resources and patterns. Please consider the following discussion as a proposed alternative approach for the workflow module.

### Definition of “Workflow”

The first problem with the module is the over-broad definition of “workflow”. The specification states that “a large proportion of the FHIR resources are devoted to the description of activities in one fashion or another and almost all of these fall into the realm of workflow - they describe things that can be done (definitions), are desired to be done (requests) or that have been done (events).”

I believe that events (event notifications), or synchronous requests (e.g. option A of the communication patterns) do not fall under “workflow”. By explicitly removing them from the definition of workflow in the module, the specification can be simplified.

### “Task”, “Request”, “Response”

I believe the mixing of Task, Request, and Response as resource patterns leads to unnecessary complication of the specification. The introduction of the Task resource is a very important development, and it needs to become the centerpiece of workflow execution. There is no need for other “actionable” resources.

This also pertains to real world implementations. A lab order or a prescription are not requests - they are specific acts on behalf of a care provider. They exist in the source system regardless of any actions taken on them. The workflow associated with these acts is what needs to be shared, and that should not be done by creating “\*Request” resources, but by sharing a Task, related to the order or prescription. Using the versioning of the Task Resurce, different versions can have references to the various requests and responses at the corresponding times.

### Do it “One Way”, please

A workflow can be represented as a series of states. The representation of these states can be implicit (as a result of specific transactions, which drive the state transition) or explicit, by modifying the resource which represents the state. I see the Task resource as the mechanism to represent the state of the workflow at a given moment. By requiring the use of Task for the purposes of workflow regardless of the interoperability paradigm, the specification will be more consistent, and therefore, easier to implement.

## Proposed Changes

### Domain specific \*Request resources

Restore the original DiagnosticOrder, etc. resources instead of the new \*Request ones, while keeping the common Task-related elements in Task

### Task resource

Modify the Task resource to contain the common actionable elements from the Request and the Response logical models (if there are any that are not already in Task). Remove the Request and Response logical models (unless there is workflow-independent reason to keep them).

Update the Task status codes to reflect common workflow states (e.g. initiated, declined, cancellation requested, cancelled, etc.)

Update the Task resource to allow statusReason to be a reference (to the event which resulted in the new status).

### Workflow description

Replace the current page with the following proposal

## **12.1 Workflow Description**

Workflow is an essential part of healthcare - orders, care protocols, referrals are the drivers of most activity within and across the care community (in-patient, outpatient, long-term care, nursing home, home care, etc.). FHIR is concerned with workflow when there's a need to share information about workflow state or relationships, when there's a need to coordinate or drive the execution of workflow across systems and when there's a need to define allowed actions, dependencies and conditions on behavior.

**Workflow state & relationships**

FHIR does not need to be used for the execution of workflow. Orders, care plans, lab results, hospital admissions, claim payments and other records can all be shared using FHIR without the process to actually solicit fulfillment of those orders or requesting payment of those claims being driven by a FHIR transaction. Interoperable support for workflow execution is actually a more advanced FHIR activity because it requires a higher degree of standardization. Rather than merely standardizing the data to exchange, interoperable workflow execution requires standardization of the processes, roles and activities of the different systems. (Note: the rest of the section is skipped, as it diverges from the topic of workflow).

**Workflow execution**

In addition to defining patterns for resources used in workflow processes, FHIR supports the execution of those processes as well. However, FHIR does not define a "one size fits all" solution for workflow architecture. FHIR supports a variety of interoperability paradigms and most of them ([REST](http://www.hl7.org/FHIR/2016Sep/http.html), [Messaging](http://www.hl7.org/FHIR/2016Sep/messaging.html) and [Services](http://www.hl7.org/FHIR/2016Sep/services.html) provide support for driving workflow execution. (The [Document](http://www.hl7.org/FHIR/2016Sep/documents.html) paradigm does not directly support driving behavior, though it can be combined with one of the other patterns to do so.) In addition, several of these paradigms may allow multiple approaches to supporting workflow, depending on the context and needs of the workflow process.

The Workflow Communication Patterns section of this page describes a number of options for workflow execution, summarizes their respective pros and cons and makes recommendations for the circumstances in which they might best be used.

**Workflow definition**

The definition of protocols, order sets, guidelines and other structures that define what sorts of activities should occur, what order they should occur on, what dependencies they have, in what circumstances they should start or end, etc. is handled by a pair of resources:

* ActivityDefinition defines an activity to be performed as a single step
* PlanDefinition defines the interrelationships of steps and the rules around their execution

The use of these two artifacts is documented TODO.

### **12.1.1 Workflow Resource Patterns**

Not all resources in FHIR are directly related to workflow - many are used to describe roles and entities (patients, medications, etc.) or infrastructure (structure definitions, value sets, etc.). However, a large proportion of the FHIR resources are devoted to the description of activities in one fashion or another and almost all of these fall into the realm of workflow - they describe things that can be done (definitions), are desired to be done or that have been done. The Task resource links these workflow-related resources together. The tables below summarize the list of workflow-relevant resources.

#### **12.1.1.1 The Task resource**

[TODO - description on how the Task resource represents the states of the workflow from initiation to completion]

#### **12.1.1.2 Workflow-related resources**

**Workflow Initiators** Resources that ask or express intention/desire for something to be done. Note that the creation/existence of such resources does not constitute initiation of the workflow in and of itself - it is the reference to one of these resources by a Task that indicates the existence of a workflow.

|  |  |  |  |
| --- | --- | --- | --- |
| Appointment\* | DeviceOrder(?) | ImmunizationRecommendation(?) | ProcessRequest(?)† |
| CarePlan | DiagnosticOrder | MedicationOrder | Referral |
| Claim | EligibilityInquiry | NutritionOrder | SupplyOrder |
| CommunicationRequest (?) | EnrollmentRequest(?) | ProcedureOrder | VisionPrescription |

***Ballot Note:*** *CommunicationRequest is a type of alert, I think it does not belong in Workflow; DeviceOrder (renamed to reflect reality) and EnrollmentRequest are not ready for publication; ImmunizationRecommendation does not seem to be workflow-related at all; ProcessRequest is to be retired. Renamed several “\*Request” to their previous name to properly reflect what they are.*

**Workflow Events** Resources that express that something has been done and which can potentially be done as a result of workflow execution. Note that these events can exist stand-alone, and only their reference from a Task resource indicates their participation in a workflow.

|  |  |  |  |
| --- | --- | --- | --- |
| AppointmentResponse\* | Coverage | FamilyMemberHistory | Observation |
| ClaimResponse | DeviceUseStatement | GuidanceResponse | PaymentNotice |
| ClinicalImpression | DiagnosticReport | ImagingStudy | Procedure |
| Communication | EligibilityResponse | Immunization | ProcessResponse† |
| Composition | Encounter | MeasureReport | QuestionnaireResponse |
| Condition (aka Problem) | EnrollmentResponse | MedicationAdministration | RiskAssessment |
| Consent | EpisodeOfCare | MedicationDispense | SupplyDelivery |
| Contract | ExplanationOfBenefit | MedicationStatement |  |

**Workflow Definitions** Resources that define something that can potentially happen in a patient and time-independent manner

|  |  |  |  |
| --- | --- | --- | --- |
| ActivityDefinition | HealthcareService | OperationDefinition | Questionnaire |
| DataElement | Measure | PlanDefinition |  |

\*  The Appointment and AppointmentResponse resources do not use the Task resource for workflow as the other resources. Their design is based on iCal conventions, so their model won't reflect the same alignment as most other resources. They are included here for completeness.

† ProcessRequest and ProcessResponse are candidates for retirement with their function subsumed by Task

Note that initiators, events and definitions don't exist in a 1:1:1 relationship. Some workflow initiators and events have obvious pairings. For example, a SupplyOrder which is referenced by the workflow Task, will generally always pair with a SupplyDelivery (also referenced by the Task). The same goes for EnrollmentRequest/EnrollmentResponse, etc. On the other hand, for other resources there isn't a strict pairing. A ReferralOrder as part of a Task might be responded to by an referencing an Encounter, DiagnosticReport, Procedure, RiskAssessment, etc. Similarly, a Procedure might be triggered by a DiagnosticOrder, ProcedureOrder, or ReferralOrder. The set of common linkages should be asserted in their respective resources. The specific types of responses for a given initiator will be governed by the Task.status and Task.code, any workflow definitions/protocols referenced and local convention.

#### **12.1.1.3 Workflow States Represented by Task**

[TODO]

### **12.1.2 Workflow Communication Patterns**

Workflow execution is supported in FHIR by several mechanisms. In considering how best to interoperate around workflow, there are a number of considerations:

* Which paradigm do you want to use (REST, messaging or services)?
* Is there infrastructure in place to support polling, push notifications via subscriptions or both?
* Is there a need for confirmation that the desired performer agrees to act, or can that be presumed?
* Is there a need to negotiate whether/how the requested action will be performed?
* Can the requesting and performing system communicate directly? Are they able to post to each other's servers (if using REST)?
* Is there an ability/need to have an intermediary to facilitate workflow execution?
* How many potential actors are involved?
* Will the workflow always be directed or is there a pool of potential performers who could choose to perform the requested action?

This section highlights some of the more common patterns and identifies their characteristics and limitations and provides recommendations on when each approach may be most useful or relevant. Please note that while this list of patterns is not exhaustive, initial focus on standardization will be on the ones listed. Patterns can be combined in various ways and there are likely some possibilities we haven't thought about yet (feel free to submit additional patterns using the 'submit a change' link at the bottom of the page). When possible, we have indicated the requirements for implementing a specific pattern. Note that tight interoperability around workflow execution (as with any other tight interoperability using FHIR) will depend on communicating participants doing some up-front negotiation around how they plan to support workflow execution or all communicating partners will need to adhere to an implementation guide that sets out clear interoperability expectations.

Prior to reviewing this list of options, readers are encouraged to be familiar with the following pages and resources: [REST](http://www.hl7.org/FHIR/2016Sep/http.html), [messaging](http://www.hl7.org/FHIR/2016Sep/messaging.html), [operations](http://www.hl7.org/FHIR/2016Sep/operations.html), [services](http://www.hl7.org/FHIR/2016Sep/services.html) and the [Subscription](http://www.hl7.org/FHIR/2016Sep/subscription.html) resource.

The scenarios below make use of a few conventions:

* The focus here is on the lifecycle of a "task" and the actions associated with it. Almost all workflows can be broken down to a sequence of these steps, though the responsibilities of the different parties may shift for each interaction and there can be more than two parties involved in the overall workflow
* Simple requests with single simple responses are the edge cases for the workflow module. While they could be implemented without the use of Task, using the patterns described here will likely make evolutionary improvements and extension of interoperability easier. The workflow initiation could be as simple as "please look at this information" and the response could be as simple as an implicit "it's been looked at" or the initiation could be for some more involved action that may include reporting back multiple interim and final steps
* The the participants are referred to as the "initiator" and as the "responder", which are often seen as request-response terms. However, in this context, the terms hold whether the request is expressed as a proposal, plan or full-blown order
* Each of the patterns defines the set of steps involved in processing the request, lists some of the benefits and limitations associated with the approach and then makes recommendations about when the pattern is most appropriate

#### **12.1.2.1 Communication Pattern Overview**

All the communication patterns serve a single purpose - sharing a Task resource between the initiator and responder. The list of patterns discussed here is as follows:

Option A: Sharing data via REST, Task via subscription, using the initiator’s FHIR server  
Option B: Sharing data via REST, Task via subscription, using the responder’s FHIR server  
Option C: Sharing data via REST, Task via subscription, using an intermediary’s FHIR server  
Option D: Sharing data and Task via messaging  
Option E: Sharing data and Task using a services request from initiator to responder

#### **12.1.2.2 Option A: Sharing data via REST, Task via subscription, using the initiator’s FHIR server**

##### **12.1.2.2.1 Steps**

1. The initiator creates a Task resource, available on its own server. The content resources (e.g. DiagnosticOrder) which are linked from the Task resource are also made available.
2. The responder receives a notification about the new resource via a subscription, retrieves Task resource and any linked resources as necessary, using simple GET.
3. The Responder takes appropriate action, and updates the Task resource.
4. Further steps, if necessary, can be carried out by both initiator and responder. The initiator is aware of any update to the task resource either via a subscription on its own, or by the fact that the resource is on its own server.

##### **12.1.2.2.2 Benefits**

* Provides the ability to handle complex workflows
* Can be used in cases where multiple possible responders “compete” for the Task.
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Can ask for things other than just fulfillment (e.g. please suspend)
* Explicit acknowledgement that the responder has received and agreed to act on the Task
* Allows for “lightweight” responders that only handle subscription notifications, without the need of a FHIR server on their own (provided they can create all necessary resources on the initiator's FHIR server)
* Uses only the REST API and subscription.

##### **12.1.2.2.3 Limitations**

* Requires subscription infrastructure.
* Requires that the responder has authority to update the Task directly to the initiator’s system (and possibly create other resources, like DiagnosticReport)

##### **12.1.2.2.4 Usage Recommendations**

This pattern enables the use of REST for all data exchanges, except Task, which is being shared via subscription. Allows for “lightweight” responders.

#### **12.1.2.3 Option B: Sharing data via REST, Task via subscription, using the responder’s FHIR server**

##### **12.1.2.3.1 Steps**

1. The initiator creates a Task resource on the responder’s server, and makes any linked resources available on its own server. The initiator also subscribes to the newly created resource
2. The responder is made aware of the new resource via the method of their choice, given that they control the FHIR server, and retrieves any linked resources as necessary, using simple GET.
3. The Responder takes appropriate action, and updates the Task resource. The initiator receives the notification of an updated Task via subscription.
4. Further steps, if necessary, can be carried out by both initiator and responder. The responder is aware of any update to the task resource either via a subscription on it’s own, or by the fact that the resource is on it’s own server.

##### **12.1.2.3.2 Benefits**

* Provides the ability to handle complex workflows
* Allows for “lightweight” initiators that only handle subscription notifications, without the need of a FHIR server on their own (provided all necessary resources are on other FHIR servers)
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Can ask for things other than just fulfillment (e.g. please suspend)
* Explicit acknowledgement that the responder has received and agreed to act on the Task
* Uses only the REST API and subscription.

##### **12.1.2.3.3 Limitations**

* Requires subscription infrastructure.
* Requires that the initiator has authority to create and update resources directly to the responder's FHIR server
* Requires the initiator to know in advance the responder for this task

##### **12.1.2.3.4 Usage Recommendations**

This pattern enables the use of REST for all data exchanges, except Task, which is being shared via subscription. Allows for “lightweight” initiators.

#### **12.1.2.4 Option C: Sharing data via REST, Task via subscription, using an intermediary’s FHIR server**

##### **12.1.2.4.1 Steps**

1. The initiator creates a Task resource on the intermediary’s server. The content resources (e.g. DiagnosticOrder) which are linked from the Task resource are also made available (either on the intermediary server, on a different server, or on the initator’s server).
2. The responder receives a notification about the new resource via a subscription, retrieves Task resource and any linked resources as necessary, using simple GET.
3. The Responder takes appropriate action, and updates the Task resource.
4. Further steps, if necessary, can be carried out by both initiator and responder. The initiator is aware of any updates to the Task resource via a subscription on its own

##### **12.1.2.4.2 Benefits**

* Provides the ability to handle complex workflows
* Allows for “lightweight” initiators that only handle subscription notifications, without the need of a FHIR server on their own (provided all necessary resources are on other FHIR servers) and/or “lightweight” responders without a need of a FHIR server on their own.
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Can ask for things other than just fulfillment (e.g. please suspend)
* Explicit acknowledgement that filler has received and agreed to act on the Task
* Can be used in cases where multiple possible responders “compete” for the Task.
* The intermediary can be a sophisticated workflow broker
* Uses only the REST API and subscription.

##### **12.1.2.4.3 Limitations**

* Requires subscription infrastructure.
* Requires a trusted intermediary

##### **12.1.2.4.4 Usage Recommendations**

This pattern enables the use of REST for all data exchanges, except Task, which is being shared via subscription. Allows for “lightweight” initiators and/or “lightweight” responders and/or workflow broker.

#### **12.1.2.5 Option D: Sharing data and Task via messaging**

##### **12.1.2.5.1 Steps**

1. The initiator sends message to responding system including Task resource (and other relevant resources) along with a MessageHeader with an "event" code saying "please act" and "data" element pointing to the Task resource as the item to act upon.
2. The responder system sends a response indicating receipt of the message and, optionally an indication of their intention to act on the Task. The response includes the updated Task.
3. The responder system may send incremental messages to the initiator using the Task resource to show progress (e.g. specimen collected, preliminary results, final results). This could include creating sub-tasks (e.g. additional tests performed based on the existing results). Similarly, the initiator may send additional messages to the responder with an updated Task resource.

##### **12.1.2.5.2 Benefits**

* Reduced number of communications
* All relevant data sent in one package
* Responses can be asynchronous and content may be routed
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Can request things other than just fulfillment (e.g. please suspend)
* Explicit acknowledgement that filler has received and agreed to act on the request (though no need for the placer to check)

##### **12.1.2.5.3 Limitations**

* Messaging is "heavy"
* Need to negotiate what allowed responses are and what data can be present in request and response messages
* Need message delivery infrastructure in place

##### **12.1.2.5.4 Usage Recommendations**

Existing messaging (likely v2) infrastructure and a need to stay consistent with that architecture

#### **12.1.2.6 Option E: Sharing data and Task using a services request from initiator to responder**

*This scenario needs work - there's not a lot of experience using FHIR services to manage the fulfillment process. It is possible that various flavors of the service pattern can be further developed to fit more special cases.*

##### **12.1.2.6.1 Steps**

1. The initiator may create and store a Task resource on their own system or an intermediary.
2. The initiator invokes a service on the responder system saying "please act on this task", including the content or a reference to the task resource and any other relevant data
3. The responder system responds (synchronously if using HTTP, but may be asynchronous if using SOAP or other transport mechanisms) with confirmation of receipt and, optionally indication of intention to act and/or completed response

##### **12.1.2.6.2 Benefits**

* ???

##### **12.1.2.6.3 Limitations**

* ???