FHIR R4 / R5 / R6

Norway

30-Aug 2023

Grahame Grieve

R5 Overview

- What's in R5
- Dealing with versions in FHIR
- R6 plans

Overview of changes in R5

- Complete Rework of Subscriptions framework
- Rework the type framework
- Move extensions to a new package
- RESTful API and Search clarifications
- New, renamed, and deleted resources & types (R4 & R4B)
- Moved many code systems and value sets to terminology.hl7.org icon
- Added Operations for Large Resources
- Added the ability to define additional bindings on elements

http://hl7.org/fhir/history.html

Rework of Subscriptions framework

- Publish & Subscribe Pattern in FHIR
  - Like WebSub, Pub/Sub, etc.
- Reusable topics across systems
- Based on resource changes or even
- Servers choose what to support
- Optimized implementations
- Server-Driven workflows

Rework the type framework

- Formal Definitions for Base types
- Formal Definitions for CanonicalResource / MetadataResource
  - As interfaces
- Mostly only relevant for code generators

Move extensions to a new package

- Frequency of updates to Extensions was starting to become a real problem
- http://hl7.org/fhir/extensions / hl7.fhir.uv.extensions
- Still integrated into the navigation structure of the standard
- But will be published more often
  - 3x a year?
- Also includes the translations between R4/R5 – so we can fix them

RESTful API and Search clarifications

- Lots of small changes to the RESTful API and search pages
  - Normative pages (no breaking changes)
- Adding clarification language
- Making conformance expectations clearer (“SHALL…”)
- These should mostly be regarded as applying to R4 implementations
- Are not expected to be breaking (but why did we have to clarify?)
- You should review them (difference, difference)

New, renamed, and deleted resources & types

- ActorDefinition, ArtifactAssessment, BiologicallyDerivedProductDispense, ConditionDefinition,DeviceAssociation, DeviceDispense, EncounterHistory, FormularyItem, GenomicStudy, ImagingSelection,InventoryItem, InventoryReport, NutritionIntake, NutritionProduct, Permission, TestPlan and Transport
- New types: integer64, Availability and ExtendedContactDetail
- Renamed resources: DeviceUseStatement -> DeviceUsage and RequestGroup -> RequestOrchestration
- Removed resources: CatalogEntry, DocumentManifest (use List), Media, ResearchDefinition, ResearchElementDefinition, RiskEvidenceSynthesis, and the type Contributor
- For the Medication Definition resource… lots of changes

http://hl7.org/fhir/history.html

Code systems and value sets to terminology.hl7.org

- Moving most value sets and code systems out to http://terminology.hl7.org (THO / UTG)
- Higher update frequency (change proposal
- More reuse in v2/CDA etc
- Didn’t move the vs/cs used in required bindings
- The package hl7.terminology.r{X} is always in scope

Operations for Large Resources

- Group and List can get very big (>100k entries) (ConceptMap?)
- Resources >1MB in size (largest sighting for me: 100MB)
- Resources that size are engineering challenges
  - May become a functional challenge (processing time approaches update frequency)
- Define operations for
  - Is entry in set
  - Add entry to set
  - Remove entry from set

Additional Bindings

- Coded elements have one binding
- Sometimes, one binding is not enough – various use cases
  - Required bindings for restricted use contexts
  - Document current binding / components of value sets
  - Provide useful UI subsets (e.g. UCUM)
  - Reduce the need for slicing (hard work for everyone)
- R5 allows you to add additional bindings
- Backported to R4 etc using extensions
- IPS Example

Other changes in R5

- 4157 change requests
- 1896 substantive changes
- Most of the changes are in response to implementation feedback

Converting between versions

- R4 Diff: http://hl7.org/fhir/diff.html

Changes from both R4 and R4B

Observation	
Observation.instantiates[x]	• Added Element
Observation.triggeredBy	• Added Element
Observation.triggeredBy.observation	• **Added Mandatory Element**
Observation.triggeredBy.type	• **Added Mandatory Element**
Observation.triggeredBy.reason	• Added Element
Observation.partOf	• Type Reference: Added Target Type GenomicStudy
Observation.subject	• Type Reference: Added Target Types Organization, Procedure, Practitioner, Medication, Substance, BiologicallyDerivedProduct, NutritionProduct

Converting between versions

- R4/R5 Transforms (using FHIR Mapping Language)

Maps to Observation

FML Conversion for Observation: R5 to R4 (Ready for Use/Trial Use)

```
/// url = 'http://hl7.org/fhir/StructureMap/Observation5to4'
/// name = 'Observation5to4'
/// title = 'FML Conversion for Observation: R5 to R4'
/// status = 'active'

conceptmap "ObservationStatus" {
  prefix s = "http://hl7.org/fhir/observation-status"
  prefix t = "http://hl7.org/fhir/4.0/observation-status"

  s:registered - t:registered
  s:preliminary - t:preliminary
```

- Not always completely successful (e.g. change was too significant)

Multiple releases of FHIR

- There have been five major releases of FHIR (Starting to work to R6)
- Over time, we change things – make breaking changes
  - It would be better if we got it right first time
  - But it's better to fix things when we don't
  - Maturity rating reflects our process – change slows down over time
- No changes to normative content (some in R4)
- In the mean time, we support multiple versions…

Supported Version Releases

- Release 1: No functional support
- Release 2: Oct 2015. Being phased out (USA)
  - Release 2B: May 2016 draft, a couple of large commercial. Java support only
- Release 3: Feb 2017. Not used much now (Europe)
- Release 4: Dec 2018. Main focus of implementation
  - Release 4B: May 2022: Reworked some aspects of R4
- Release 5: Mar 2023. Gradual Adoption

Release History

- http://hl7.org/fhir/directory.html / http://hl7.org/fhir/package-list.json


```
{
  "package-id" : "hl7.fhir.core",
  "title" : "FHIR Specification",
  "canonical" : "http://hl7.org/fhir",
  "introduction" : "This table provides a list of all the versions of FHIR (Fast
  "footnote" : "Note: Subsequent to Sept 2013, the FHIR version policy was change
  "list" : [ {
    "version" : "current",
    "date" : "n/a",
    "desc" : "Current Development build (about 30min behind version control, may
    "path" : "http://build.fhir.org",
    "status" : "ci-build",
    "current" : true
  },{
    "version" : "4.0.0",
    "date" : "2018-12-27",
    "desc" : "FHIR Release #4: First Normative Content",
    "path" : "http://hl7.org/fhir/R4",
    "status" : "normative+trial-use",
    "sequence" : "R4",
    "current" : true
  },{
```

Versioning is expensive

- Historically, version changes have been very expensive
  - Or, profitable for some, but bad for health
- Much argument about everything to do with versions….

What version is this resource?

- Not explicit in every resource
  - Version is a property of the channel/context, not the data
  - Resources might be correct in multiple versions
- Explicit in the CapabilityStatement
  - Not necessarily the same version! e.g. a cross-version repository of end-points
- Explicit in Profiles & Implementation Guides
  - These may be properties of resources or properties of the context
  - Resources can (/should) conform to many profiles (+versions)

Stamping version inside a resource

- Use Resource.meta.profile:

```
{
  "resourceType" : "Patient",
  "meta" : {
    "profile" : [
      "http://hl7.org/fhir/4.0/StructureDefinition/Patient",
      "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient|5.0"
    ]
  },
  ...
}
```

Version Numbering Strategy (Standards)

- Publication.Minor.Patch
  - Differ to SemVer in nature of changes allowed with minor revisions
- R3: 3.0.2 – 2 patch on R3
  - Intermediate versions 3.x.0 by release and then 4.0.0 once finalized
  - R4B is 4.3 because 4.1 & 4.2 were used by R5 drafts
- As of R5 – move to using label: 5.0.0-snapshot3
  - Can’t sort purely alphabetically
  - R6 currently is 6.0.0-cibuild (rolling build: cibuild = unstable)
- HL7 Implementation Guides follow the same pattern

Versioning Profiles, Value sets, Code Systems

- Policy: Use Semver
  - Not all sources do (see hl7.org/fhir/codesystem-version-algorithm.html)
  - Genuinely breaking changes: new artifact
- Can version independently, but we encourage you to version by package
  - IG publisher can impose common version
  - What's easiest for tracking changes?
- Unversioned references in packages are 'package versioned'
  - Use versioned package dependencies to resolve references
  - Easy for authors, harder for tools / implementers – but being done

Versioning Extensions

- References in Extension.url are not versioned
- Breaking changes in extension definitions are not supported
  - Not quite the same as not possible
- References to extensions in profiles can be versioned (same rules)
- Validation might be based on older version

Cross-Version extensions

- Adopt an element in an earlier version
- See http://hl7.org/fhir/versions.html#extensions
- http://hl7.org/fhir/[*version*]/StructureDefinition/extension-[*Path*]
  - New Elements
  - New types on choice elements
  - New resource types (use Basic)

```
{
  "resourceType" : "Basic",
  "id" : "uscdi",
  "text" : {
    "status" : "generated",
    "div" : "<div xmlns=\"http://www.w3.org/1999/xhtml\"><table class=\"grid\"><tr><td colspan=
  },
  "extension" : [{
    "url" : "http://hl7.org/fhir/5.0/StructureDefinition/extension-SubscriptionTopic.url",
    "valueUri" : "http://hl7.org/fhir/us/core/SubscriptionTopic/uscdi"
  },
  {
    "url" : "http://hl7.org/fhir/5.0/StructureDefinition/extension-SubscriptionTopic.version",
    "valueString" : "6.0.0"
  },
  {
    "url" : "http://hl7.org/fhir/5.0/StructureDefinition/extension-SubscriptionTopic.name",
    "valueString" : "my topic name"
  },
  {
    "url" : "http://hl7.org/fhir/5.0/StructureDefinition/extension-SubscriptionTopic.title",
    "valueString" : "example"
  },
  {
    "url" : "http://hl7.org/fhir/5.0/StructureDefinition/extension-SubscriptionTopic.status",
```

Cross-Version extensions: Limitations

- Differences in available data types
- Differences in narrative constraints on data values
- Differences in terminology bindings
- Extensions already exist (not uncommon)

- No resolution at this time (ongoing discussion)

Versioning Software Tools

- You do you
  - We like semver (but vary from it)
- FHIR internal tooling stack is mature (~1 decade old)
  - tx.fhir.org, Java HAPI Core, Java validator, IG Publisher, Core publisher
  - No particular product roadmap (i18n!)
  - Effectively a stream of regular releases
  - Major version numbers are somewhat arbitrary
  - Minor/patch maintained

Supporting Multiple Versions

- Converting between versions
- API Strategy
- Persistence Strategy
- Documentation Strategy
- IG Publisher support

Converting between versions

- R4 Diff: http://hl7.org/fhir/diff.html

Changes from both R4 and R4B

Observation	
Observation.instantiates[x]	• Added Element
Observation.triggeredBy	• Added Element
Observation.triggeredBy.observation	• **Added Mandatory Element**
Observation.triggeredBy.type	• **Added Mandatory Element**
Observation.triggeredBy.reason	• Added Element
Observation.partOf	• Type Reference: Added Target Type GenomicStudy
Observation.subject	• Type Reference: Added Target Types Organization, Procedure, Practitioner, Medication, Substance, BiologicallyDerivedProduct, NutritionProduct

Converting between versions

- R4/R5 Transforms (using FHIR Mapping Language)

Maps to Observation

FML Conversion for Observation: R5 to R4 (Ready for Use/Trial Use)

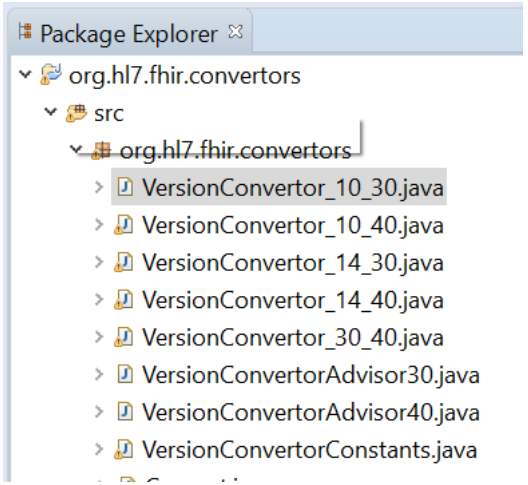
```
/// url = 'http://hl7.org/fhir/StructureMap/Observation5to4'
/// name = 'Observation5to4'
/// title = 'FML Conversion for Observation: R5 to R4'
/// status = 'active'

conceptmap "ObservationStatus" {
  prefix s = "http://hl7.org/fhir/observation-status"
  prefix t = "http://hl7.org/fhir/4.0/observation-status"

  s:registered - t:registered
  s:preliminary - t:preliminary
```

- Not always completely successful (e.g. change was too significant)

Java Converter




```
21904
21905
21906  public static org.hl7.fhir.r4.model.Resource convertResource(org.hl7.fhir.dstu3.model.Resource src, boo
21907    if (src == null)
21908      return null;
21909    if (src instanceof org.hl7.fhir.dstu3.model.Parameters)
21910      return convertParameters((org.hl7.fhir.dstu3.model.Parameters) src);
21911    if (src instanceof org.hl7.fhir.dstu3.model.ActivityDefinition)
21912      return convertActivityDefinition((org.hl7.fhir.dstu3.model.ActivityDefinition) src);
```

- Only supported fully for conformance resources
- Contributions for other resources are welcome

Version independent logic

- Use a façade in front of versions e.g.

```
IPatient = interface (IDomainResource) {
  IHumanName getNameI();
}

R3.Patient = class (DomainResource, IPatient) {
 public HumanName getName() {…}
 public IHumanName getNameI() {…}
}
```

- This is a lot of work, but partially done in some reference implementations – you can contribute

Resource Conversion isn't everything

- GET [base]/Patient/[id]?(params)
  Accept: [content-type]

- HTTP 200 OK
  Content-Type: [content-type]

  { … body …}

- The entire exchange has a version (can't mix with one exception)

Simplest Approach: multiple end-points

- http://test.fhir.org/r3
- http://test.fhir.org/r4
- http://test.fhir.org/r5

- fhirVersion element in the applicable CapabilityStatement applies
- Pro: Simple
- Con: Logical records get multiple URLs

Single end-point, multiple versions

- http://test.fhir.org/rX
- The fhirVersion parameter on the MIME-type that applies to the resource (but fixes the whole exchange)

```
GET [base]/metadata
Accept: application/fhir+json; fhirVersion=4.0
```

Single end-point, multiple versions

- Server specifies what versions it supports, with a default
- Client chooses a version using the fhirVersion parameter
- Fixes the whole exchange

- Conversion information for resource names and search parameters: https://github.com/FHIR/interversion/tree/master/package

Determining Server Versions

```
GET [base]/$versions
Accept: application/json
[other headers]
```

```
{
  "versions": ["3.0", "4.0"],
  "default" : "4.0"
}
```

$convert

- Ask server to convert versions

```
POST /base/$convert
Accept: application/fhir+json; fhirVersion=3.0
Content-Type: application/fhir+json; fhirVersion=4.0
```

Non-API Exchange

- There's almost always a mime type:

```
application/fhir+json; fhirVersion=4.0
```

- If that's not possible (local files, no context policy):

```
"meta" : {
  "profile" : ["http://hl7.org/fhir/4.0/StructureDefinition/Patient"]
}
```

Persisting Multiple versions

- Store Resources with known version (implicit, or explicit)
- Use the profile marker if you really need to

Persistence and Conversion

In general 3 options:

- Store resources as you get them (and convert on the fly if needed)
- Store resources in your preferred version (and convert if needed)
- Extract information from resources and store in (relational?) database

Or… Do all 3 things at once:

- Store resources as you first received them (for audit trail)
- Store resources in your preferred version (for flexibility)
- Build specific tables for particular indexing (for performance)

Documentation Strategy

- Simple: Different documentation for different versions
  - Multiple repetitions of narrative
  - Implementers have to compare between versions
- Complex: One set of documentation, with different profiles/examples
  - One combined narrative with explicit differentiation
  - Implementers explicitly deal in multiple versions

- Which is better depends on the implementers
- Do the business rules differ? What about documentation versions?

IG Publisher Support

- IG Publisher is R5 internally
  - All profiles, value sets, code systems etc are converted to R5
- Some IGs are multi-version
  - Multi-version output – terminology e.g. THO: .r4, .r4, .r5 – R{X} versions of resources
  - Multi-version output - some profiles e.g. Subscription Backport (.r4, .r4b)
  - Use R5 resources internally (but no implementation support for R5 resources)
  - No plans for fully cross-version IGs (break the backbone!)

Managing Multiple Versions

- Versions are expensive and painful
- There’s some fantasy land where they don’t happen

Should you move to R5?

- Do you need the new things in R5
  - Have they been / can they be backported?
  - How much will you benefit otherwise?
- How much will it cost?
  - How big is your eco-system, what's your change overhead?
- For most existing trading systems, cost/benefit says don't change
- But don't ignore R5: there's lots of clarifications and fixes
  - Important advice for implementing R4

FHIR Manifesto

- Focus on Implementers
- Target support for common scenarios
- Leverage cross-industry web technologies
- Require human readability as base level of interoperability
- Make content freely available
- Support multiple paradigms & architectures
- Demonstrate best practice governance

Tuckmans Stages of Group Development

- **Forming**
- **Storming**
- **Norming**
- **Conforming** (performing?)
- Adjourning

Our goal for R6

- **Make the patient core ‘Normative’**

Foundation

Conformance	Terminology	Security	Documents	Other
• CapabilityStatement **N** • StructureDefinition **N** • ImplementationGuide 4 • SearchParameter 5 • MessageDefinition 1 • OperationDefinition **N** • CompartmentDefinition 3 • StructureMap 4 • GraphDefinition 2	• CodeSystem **N** • ValueSet **N** • ConceptMap 3 • NamingSystem 4 • TerminologyCapabilities 1	• Provenance 4 • AuditEvent 4 • Permission 0 • Consent 2	• Composition 4 • DocumentReference 4	• Basic 3 • Binary **N** • Bundle **N** • Linkage 0 • MessageHeader 4 • OperationOutcome **N** • Parameters **N** • Subscription 3 • SubscriptionStatus 2 • SubscriptionTopic 2

Base

Individuals	Entities #1	Entities #2	Workflow	Management
• Patient **N** • Practitioner 5 • PractitionerRole 4 • RelatedPerson 5 • Person 4 • Group 3	• Organization 5 • OrganizationAffiliation 1 • HealthcareService 4 • Endpoint 2 • Location 5	• Substance 2 • BiologicallyDerivedProduct 2 • Device 2 • DeviceMetric 1 • NutritionProduct 1	• Task 3 • Transport 1 • Appointment 3 • AppointmentResponse 3 • Schedule 3 • Slot 3 • VerificationResult 1	• Encounter 4 • EncounterHistory 0 • EpisodeOfCare 2 • Flag 1 • List 4 • Library 4

Clinical

Summary	Diagnostics	Medications	Care Provision	Request & Response
• AllergyIntolerance 3 • AdverseEvent 2 • Condition (Problem) 5 • Procedure 4 • FamilyMemberHistory 2 • ClinicalImpression 1 • DetectedIssue 2	• Observation **N** • DocumentReference 4 • DiagnosticReport 3 • Specimen 2 • BodyStructure 1 • ImagingSelection 1 • ImagingStudy 4 • QuestionnaireResponse 5 • MolecularSequence 1 • GenomicStudy 0	• MedicationRequest 4 • MedicationAdministration 2 • MedicationDispense 2 • MedicationStatement 4 • Medication 4 • MedicationKnowledge 1 • Immunization 5 • ImmunizationEvaluation 1 • ImmunizationRecommendation 1 • FormularyItem 0	• CarePlan 2 • CareTeam 2 • Goal 2 • ServiceRequest 4 • NutritionOrder 2 • NutritionIntake 1 • VisionPrescription 3 • RiskAssessment 2 • RequestOrchestration 4	• Communication 2 • CommunicationRequest 2 • DeviceRequest 1 • DeviceDispense 0 • DeviceAssociation 0 • DeviceUsage 1 • BiologicallyDerivedProductDispense 0 • GuidanceResponse 2 • SupplyRequest 1 • SupplyDelivery 1 • InventoryItem 0 • InventoryReport 0

Support	Billing	Payment	General
• Coverage 4	• Claim 2	• PaymentNotice 4	• Account 2

Normative

“No breaking change such that previous implementations that were correct become incorrect”

- In a bi-directional interface, this is rather difficult to define, and to decide

- But it’s pretty much happening by default with R4
  - Our feedback is gradually getting clearer

Normative Tasks

- Get implementer feedback / market survey to confirm our decision
- Change our existing processes so that we can succeed in getting ‘normative’
- Improve the definitions around breaking changes and make implementer expectations clearer
- Decide which resources will be made normative, and which won’t
  - FOMO for some resources
- Some resources are not candidates (too early in the Tuckman cycle)

Other R6 Agendas

- Finish with Obligations / Additional Bindings
- Improve quality processes/QA on supporting collateral
- Invest more in version Migration Assistance
- Can we do something about on-ramps? (Complexity)
  - Determine common on-ramps for implementers

Supporting implementers

- Support scaling the ecosystem internationally
  - 88 countries and growing quickly
- Continue to improve the tools that support the eco-system
  - Validator, simplifier, terminology servers(!!!), sushi, code generators, publishing tools
- Grapple with the IG/profile explosion
- Improve testing / conformance eco-system
- More work on mapping / transformation
- Work with regulators – building relationships and trust