

FHIR US Conformance: Specification Outline

US Realm Specifications

-- overview

- US Core
- Da Vinci
- ...

Underlying specifications

- FHIR R4+
- CDS Hooks
- SMART on FHIR Launch
- FHIRcast - future

Problem

- Need consistent FHIR conformance across US Implementation Guides
- Need to constrain and extend US-realm profiles in a consistent way
- Need to declare the responsibilities of a systems in a 1st generation FHIR API used to query an EHR systems; the sender (e.g. server) or receiver (e.g. client)
- Need to prepare the implementer for issues related to versioning, searching resources using FHIR,
- Need best practices
 - How to reuse specifications
 - API
 - search
 - hooks
 - create/update
 - custom operation
 - Data Structures
 - US profiles reuse
 - US profile testing
 - "Mandatory" element use to specify "supported" and "must be specified"
 - Use of null flavor (i.e. "dataAbsentReason")
 - Allowing and prohibiting modifiers extensions
 - what is the default behavior? Allo
 - adding local extension (e.g. Z-Segments)
 - grouping extension similar to Z-segments
 - specifying which resource can use an extension (e.g. Patient-specific extensions, Observation-specific, Global extension)
 - Extensions
 - IG specific extensions - adding mapping to external terminology (concept domains) to avoid duplication?
 - modifier extensions - are they allowed for "mandatory" data elements (e.g. USCDI)?
 - data type extensions - **new to FHIR, not present in V2 or CDA !!!**

Data Validation rules for US Realm

- Applicable for all specification that use FHIR resources

Transition Table from Base Standard to Profile

What are the rules for organizations (like VA) when reusing/containing/extending a US-realm IG/profile?

This section will describe the rules relate to mandatory/cardinality/extensibility:

- Add transition table base standard -vs profile

Conformance Usage Indicator to specify Support and Presence Rules

This discussion summarizes how the V2 and V3 conformance usage indicator could be used to enhance the clarity of FHIR profile and designate which are the Mandatory, Required, Required but may be empty, or in-scope/optional USCDI.

Extensions. valueCode	Min Cardi nality	Max Cardi nality	Data Absent Reason (null indicator)	Local Extensions (not in IG)	Coded Data Elements	Terminology binding	Testing (preliminary discussion)	Client responsibilities	Server responsibilities
--------------------------	------------------------	------------------------	---	------------------------------------	------------------------	------------------------	--	-------------------------	----------------------------

Mandatory	1	>0	not allowed	allowed?	"unknown" or equivalent is not allowed as a value	required (value set mandatory)	All test cases will mandate the presence of this data element. Some of these elements may be mandatory in the base standard (e.g. Observation.status).	must be able to process the data consistent with the IG requirement	Must produce a clear exception if the data element is missing
Required	1	>0	allowed	allowed?	any valid code allowed	extensible (value set mandatory)	Data absent must be used when the value is not available: <ul style="list-style-type: none"> pass: test for absence with null indicator pass: test for presence fail: absence without null indicator 	must be able to process the data consistent with the IG requirement	
RequiredButEmpty	0	>0	allowed	allowed	any valid code allowed	extensible (value set mandatory)	Test cases may validate: <ul style="list-style-type: none"> pass: absence of value (e.g. data doesn't appear in the EHR) pass: absence of value + null (e.g. patient refused to answer) pass: in some cases /pre-conditions, the value will be expected (e.g. gender at birth was entered) 	must be able to process the data consistent with the IG requirement	
NotSupported	0	0	not applicable	not applicable	not applicable	not applicable	All test cases will mandate the absence of this data element.	no action	
<unspecified> (missing)	any	any	undetermined	undetermined	undetermined	undetermined	? (ask Brett, Hans, Dan) If present, it should be correctly populated based on the base standard (v2 best practice). Test cases cannot require the presence of these data elements but the best-practice recommendation is to add a derived profile that identifies additional M/R /RE data elements. This way it's clear to implementers if another data element beyond the core data is needed for this use case.	? undetermined	

Conditional, slices, conditional									
----------------------------------	--	--	--	--	--	--	--	--	--

(

- Invariant expressions are inherited from base resource/data elements

Profiles identify:

- Mandatory (is supported and must be valued)
- Supported (is supported and may be valued, if not valued a "dataAbsentReason" must be specified)
- Not supported (in the US)

Data elements not constrained in the profile are "unspecified" - they inherit the invariant and data semantics in the base standard.

Principles...

1. A profile must contain at least one "required" data element
2. All data elements affected by the project requirements will be annotated with the "must support" indicator: set to "true" or "false"(if the data element is explicitly out-of-scope)
3. If a data element has a min cardinality of 1 in the base resource (e.g. status) the profile cannot mark that data element as "not supported". As a best-practice it should be included in either "mandatory" or "required"
4. CapabilityStatement
 - a. Multiple levels of capability statements that can be reused across servers
 - b. US ARC statement may be reused by implementers

TODO: server vs client responsibilities for read/only ("search", "get") APIs

- display responsibilities clients:
 - may be transformed/computed for display
 - unless explicitly specified, all supported/mandatory data will be displayed

Modifier Extensions

By default, not supported in US Realm implementation guides

If they are added by an organization or project, these extensions MUST be registered to the US Core Registry.

Q: US registry of templates and extensions will be available before January 2020?

Adding Local Extensions

Z-segments to FHIR extensions

Outline for Guidance Paper

List of what should be included:

- Want to add a cookbook of Conformance Methodology that matches what we did for HL7 v2.
- Summarize all of the existing built-in constraints of the base standard
 - Expand the implications of those constraints
 - Have notion of cardinality, but it is often left to interpretation of exactly what it means
 - Example: An optional field in FHIR might be a field that has a minimum cardinality of 0 and Must Support not flagged. (I.e. build a table with all the combinations, then state the interpretation in "plain English")
 - Describe the notion of levels of further constraining or profiling
 - What are allowable transitions for every level
 - Do we want to introduce the concept of base level, constrainable, implementable level, etc? Yes, we will keep the term.
- We want to create a set of constraints on what allowable to do within that framework.
 - Already exists: Structured definitions allow you specify snapshots and deltas. From that perspective it is complete, but it still allows you to do things like use extensions. Mechanism is available to use.