IHE EUROPE EXPERIENCE DAY 13 SEPTEMBER 2022

Development of profiles that allow conformance testing

IHE CONNECTATHON 2022 BUILDING THE BRIDGE MONTREUX+ATLANTA SEPTEMBER 12-16

Perspective and status on improved testing of IHE profiles and FHIR specification

Introductions

José Costa Teixeira

jose.a.teixeira@gmail.com

- Chair, HL7 Belgium
  - VAZG / RIZIV
- Co-chair, IHE Pharmacy
- PATH / Digital Square

Why make testable specifications?

"There are 2 ways to be happy: Improve your reality, or lower your expectations"

…

"If you don't set your expectations, you have to be happy with whatever you get"

From "best effort" testing to uniform, repeatable testing

Better testing → Better apps → Better care (really?)

Agenda

IHE process - gap and analysis

IHE metamodel description and needs for testing

What would “good testing” be like? Q&A

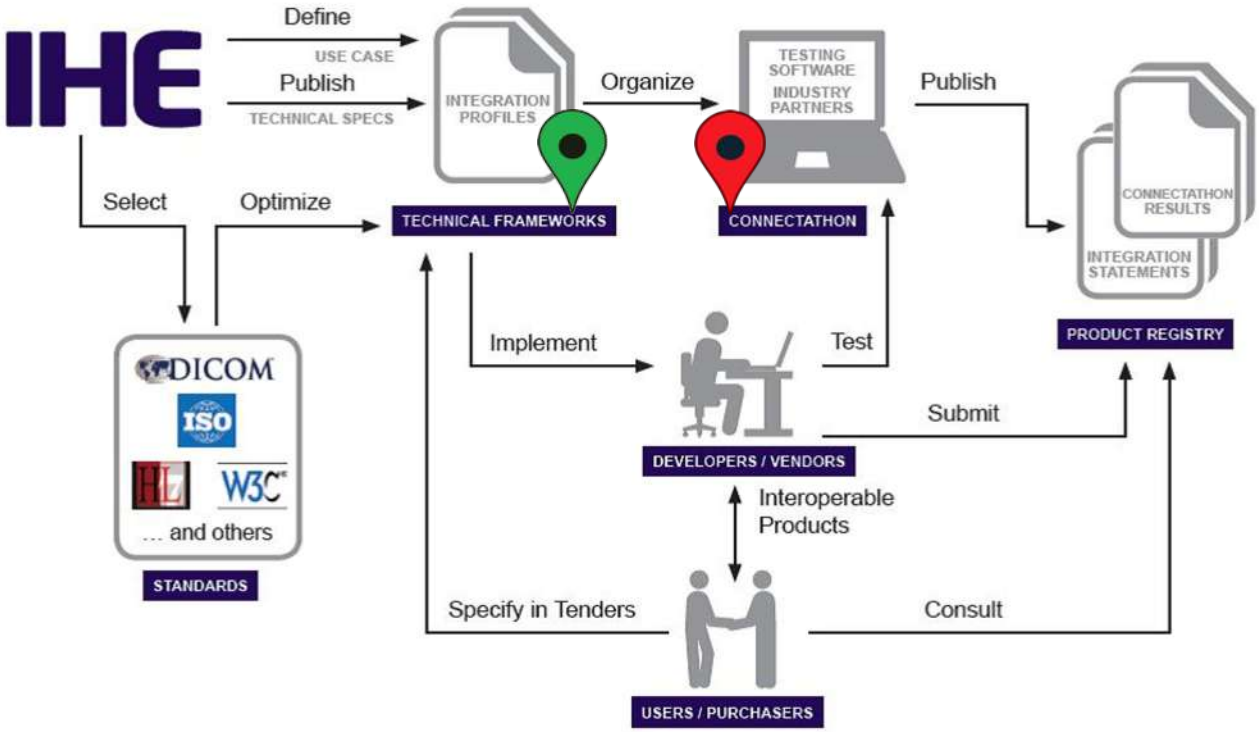
Vision: self-asserting systems

Gap summary

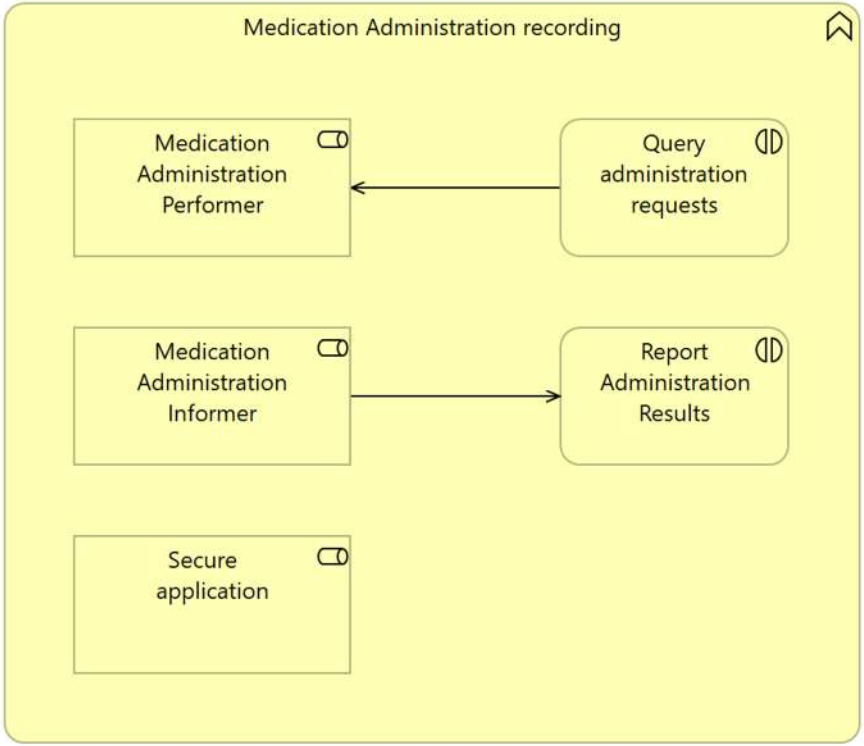
Improving the way to make testable specifications

- Embed structured testing design within the specs
- Build in self-validation - through test data and test criteria - and clarify the scope to make it scalable and automatable
  - Less guesswork
  - Less best-effort
- Enabling automation - devops - real agility

The IHE process



IHE Testable application



What are we testing?

IHE profiles specify part of the interoperable part of an application. That's what we can test.

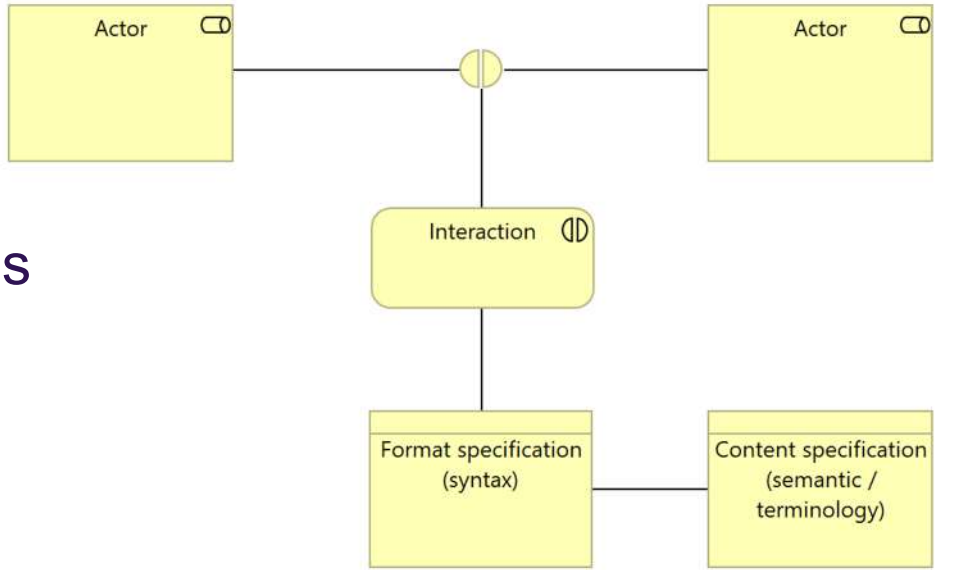
IHE specifications

IHE has a “language”:

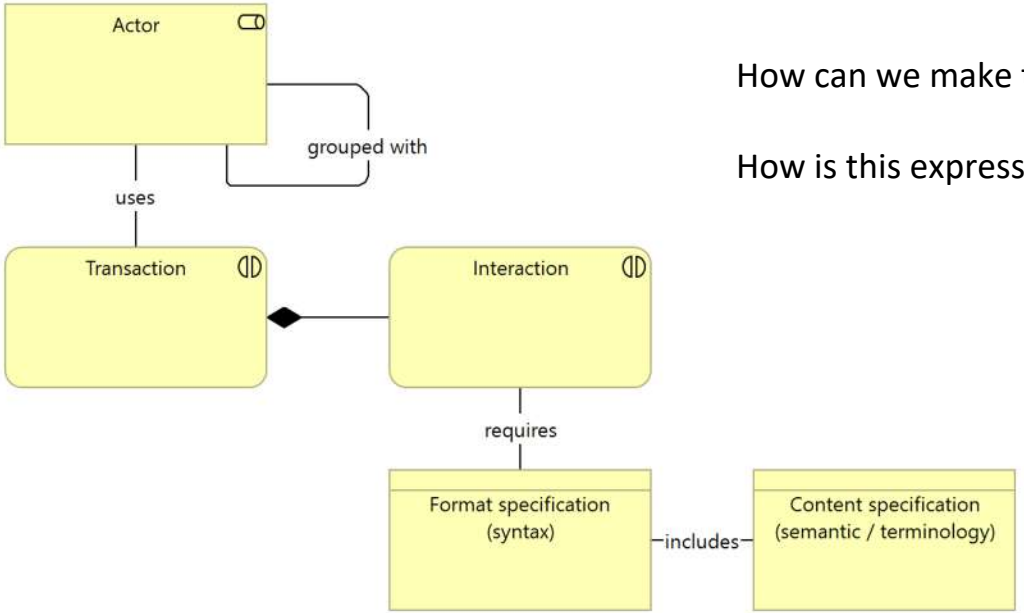
IHE Actors: abstract functional role specifications (for interoperability)

IHE Transactions: standards-based data exchanges

…



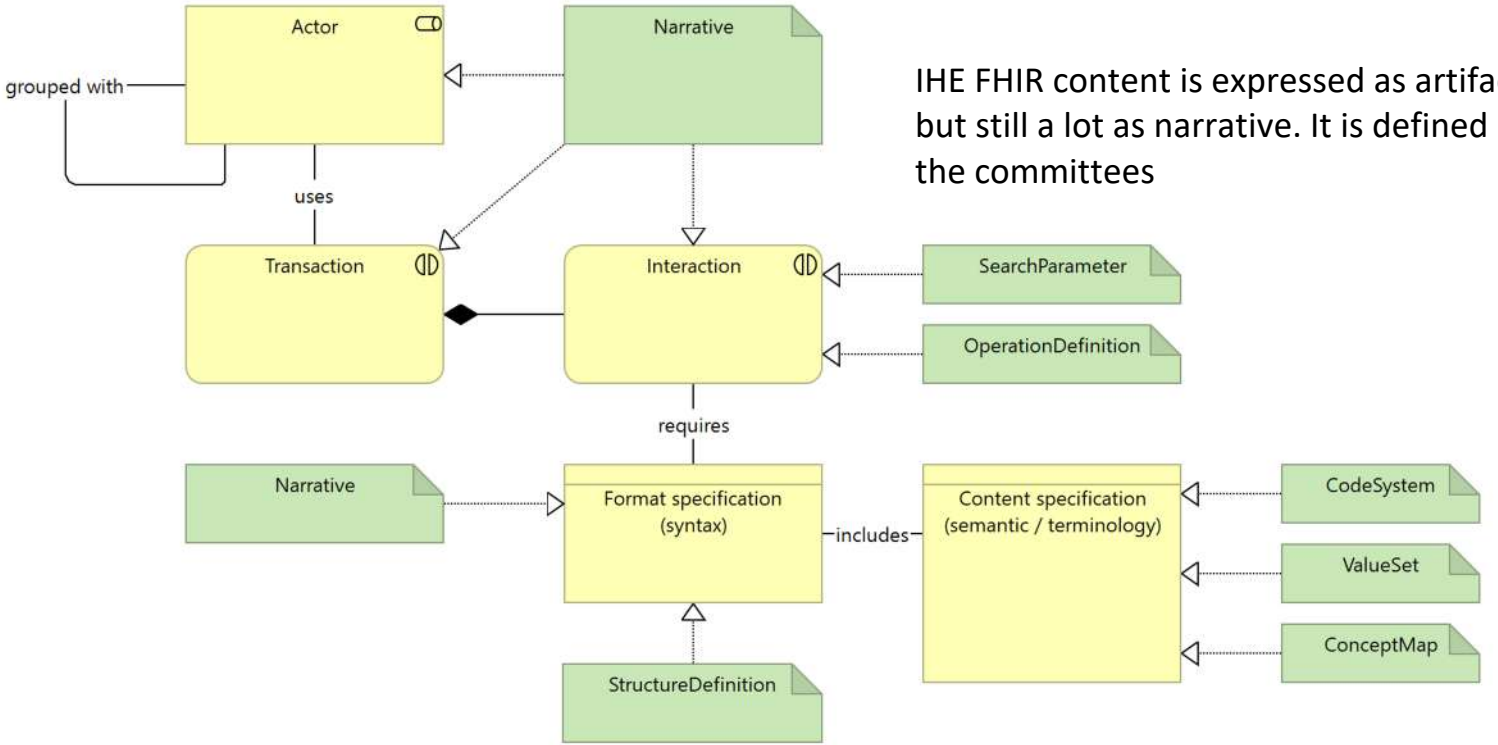
IHE Meta model



How can we make this testable?

How is this expressed? By whom?

IHE content on FHIR



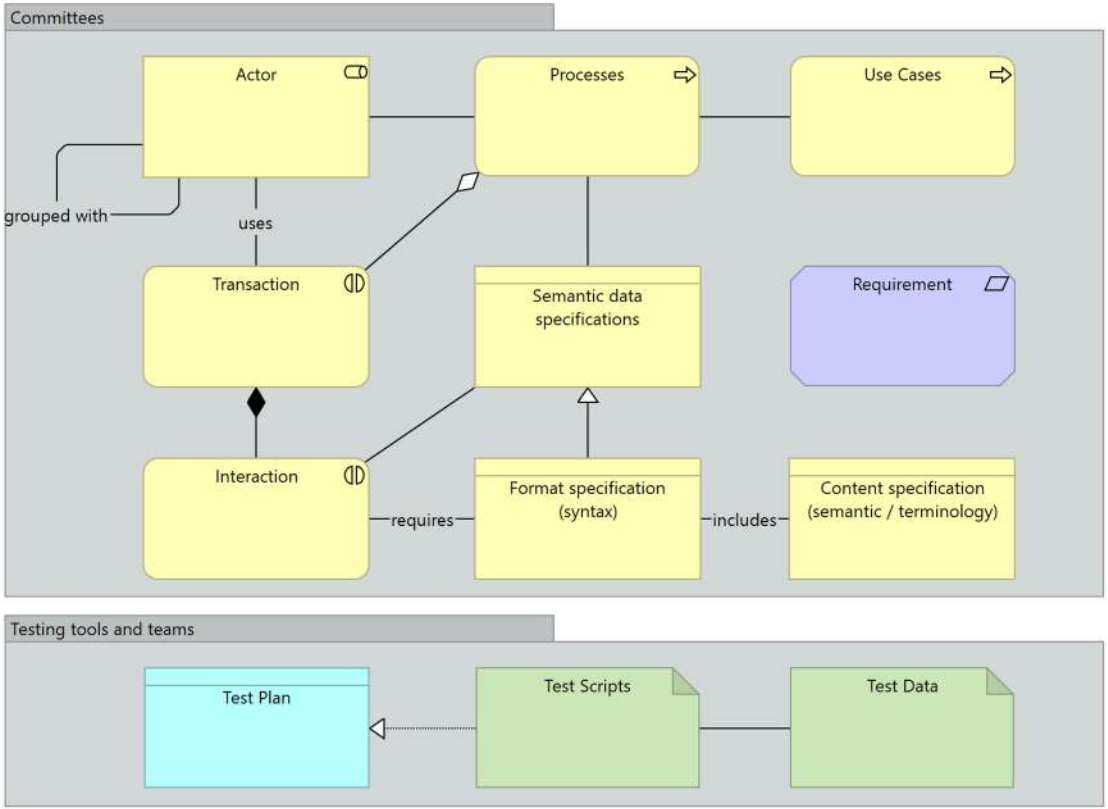
IHE FHIR content is expressed as artifacts but still a lot as narrative. It is defined by the committees

IHE testing (to-date)

From spec to testing

- Read the specs
- Define test cases (which? How many?)
- Create test data
- Run the tests
- Report the tests
- Feedback of testing into the specs

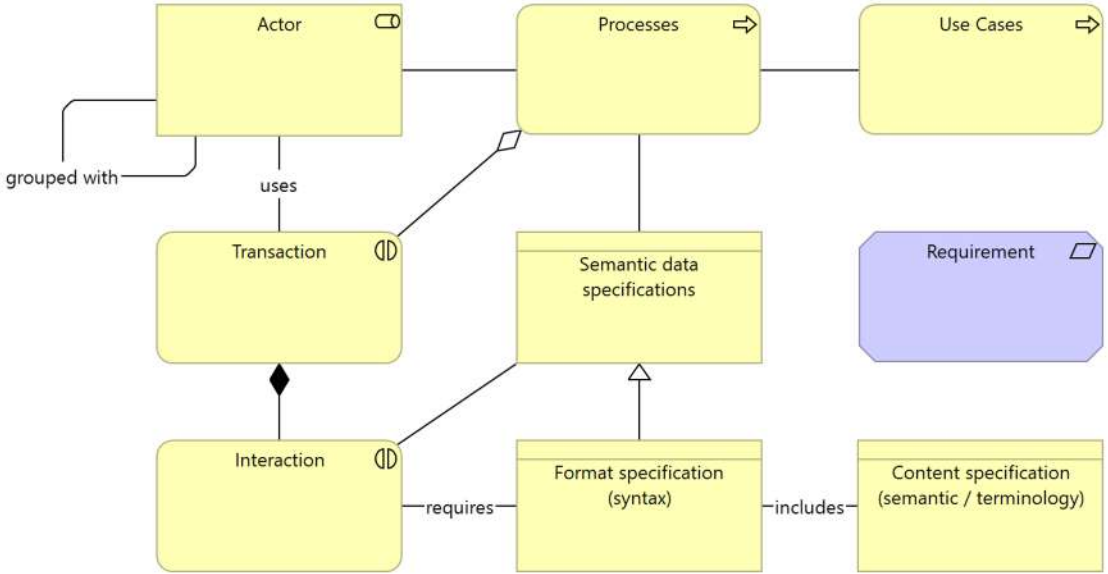
What are “testable” specifications?



Designed by 2 teams, in different moments.

Some guess work

What should we be testing (against)?



Processes: Do the parts work together?

Semantics: Do we mean the same?

Requirements: Are the rules met?

Vision / WIP

1. Facilitate the alignment TCommittee - Connectathon

1. Give implementers something to get started

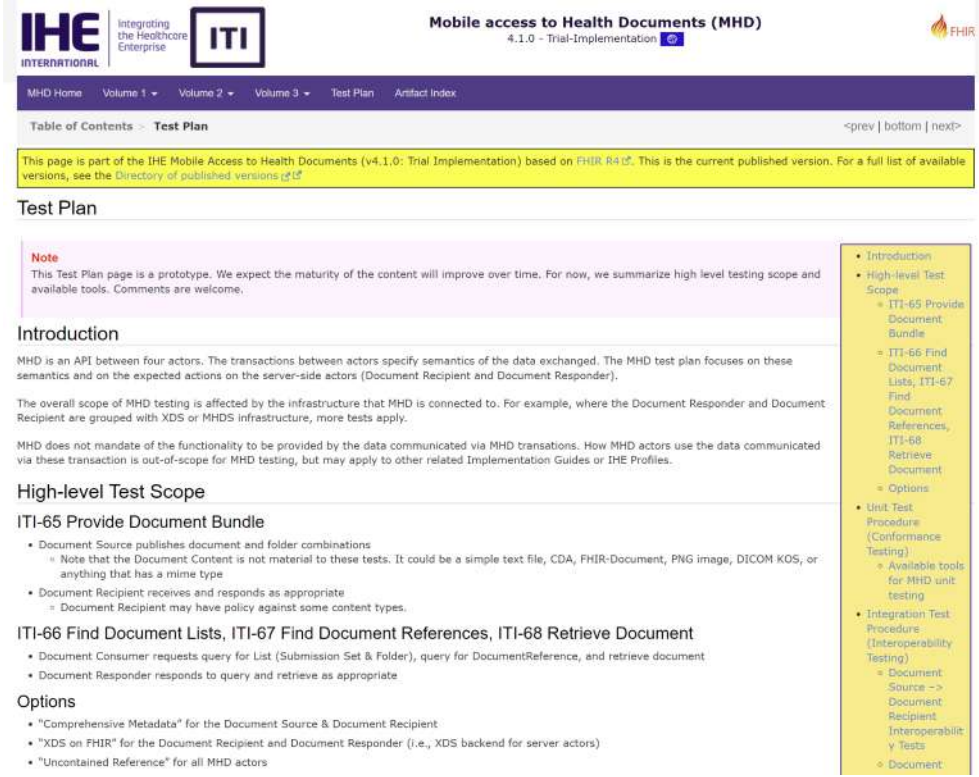
1. DevOps the whole thing

Status and expectation

- IHE has been moving to FHIR-based specifications
- Lots of interest to make things more testable
- Tooling being developed, for better integration

- We can expect that authors will be able to create more testable specs soon

Test plans as part of IGs



IHE INTERNATIONAL — Integrating the Healthcare Enterprise — ITI

Mobile access to Health Documents (MHD)
4.1.0 - Trial-Implementation

MHD Home | Volume 1 | Volume 2 | Volume 3 | Test Plan | Artifact Index

Table of Contents > Test Plan <prev | bottom | next>

This page is part of the IHE Mobile Access to Health Documents (v4.1.0: Trial Implementation) based on FHIR R4. This is the current published version. For a full list of available versions, see the Directory of published versions

Test Plan

Note
This Test Plan page is a prototype. We expect the maturity of the content will improve over time. For now, we summarize high level testing scope and available tools. Comments are welcome.

Introduction

MHD is an API between four actors. The transactions between actors specify semantics of the data exchanged. The MHD test plan focuses on these semantics and on the expected actions on the server-side actors (Document Recipient and Document Responder).

The overall scope of MHD testing is affected by the infrastructure that MHD is connected to. For example, where the Document Responder and Document Recipient are grouped with XDS or MHDS infrastructure, more tests apply.

MHD does not mandate of the functionality to be provided by the data communicated via MHD transations. How MHD actors use the data communicated via these transaction is out-of-scope for MHD testing, but may apply to other related Implementation Guides or IHE Profiles.

High-level Test Scope

ITI-65 Provide Document Bundle

- Document Source publishes document and folder combinations
  - Note that the Document Content is not material to these tests. It could be a simple text file, CDA, FHIR-Document, PNG image, DICOM KOS, or anything that has a mime type
- Document Recipient receives and responds as appropriate
  - Document Recipient may have policy against some content types.

ITI-66 Find Document Lists, ITI-67 Find Document References, ITI-68 Retrieve Document

- Document Consumer requests query for List (Submission Set & Folder), query for DocumentReference, and retrieve document
- Document Responder responds to query and retrieve as appropriate

Options

- "Comprehensive Metadata" for the Document Source & Document Recipient
- "XDS on FHIR" for the Document Recipient and Document Responder (i.e., XDS backend for server actors)
- "Uncontained Reference" for all MHD actors

Unit Test Procedure (Conformance Testing)

Unit testing this context entails testing a SUT with a simulator or validator tool. A simulator is a implementation of an actor that is designed specifically to test the opposite pair actor. The simulator might be a reference implementation or may be a specially designed test-bench. Often, when a reference implementation is used, the negative tests are harder to simulate. A validator is a implementation that can check conformance. A validator may be a simulator, but may also be a standalone tool used to validate only a message encoding. Some reference implementations may be able to validate to a StructureDefinition profile, but often these do not include sufficient constraints given the overall actor conformance criteria.

Available tools for MHD unit testing

FHIR Toolkit (aka "Asbestos") - Simulator and Validator

- Provider: NIST (US National Institute of Standards and Technology)
- FHIR Toolkit online: https://tools.ihe.usa.org:9743/home
- Tool distribution: https://github.com/usnistgov/asbestos/releases/
  - Release 2.x.x is compatible with MHD version 3.x
  - As of June 2021, FHIR Toolkit is not compatible with MHD version 4.x
  - Features may be added to FHIR Toolkit in subsequent releases. Check release notes in the tool distribution link above
- Documentation (installation): https://github.com/usnistgov/asbestos/wiki/xInstallation-Guide-v2.x.x
- Tool support: http://groups.google.com/group/ihe-mhd-implementors
- Actors (options) tested: Document Source (minimal metadata), Document Source (comprehensive metadata), Document Source (minimal metadata), Document Source (comprehensive metadata)
- Other notes:

Gazelle External Validation Service (aka "EVS Client") - Validator

- Provider: INRIA (Rennes, France), KEREVAL (https://www.kereval.com/), and Mallinckrodt Institute of Radiology (Saint Louis, USA)
- Tool location: https://gazelle.ihe.net/EVSClient/home.seam
- Documentation: https://gazelle.ihe.net/content/evsfhirvalidation
- Scope of testing: validation of FHIR Resources using StructureDefinitions for MHD
- Other notes: StructureDefinitions available in EVSClient validation are published by IHE here: https://github.com/IHE/fhir/tree/master/StructureDefinition

Integration Test Procedure (Interoperability Testing)

Integration Testing in this context is where two SUT of paired actors test against each other. Integration testing is often limited by the capability of the client (Document Source or Document Consumer), which may support only a subset of the semantics required to be supported by the server (Document Recipient or Document Responder). Full message semantics and failure-modes are more thoroughly exercised with unit (conformance) tests.

The tests listed below are defined in Gazelle Master Model (https://gazelle.ihe.net/GMM) and are performed by systems testing MHD at IHE Connectathons.

Document Source –> Document Recipient Interoperability Tests

- MHD_Create_ITI-65
- MHD_Create_with_List
- MHD_Submit_XDSonFHIR
- MHD_XDSonFHIR_Replace
- MHD_XDSonFHIR_Append
- MHD_XDSonFHIR_Transform

Document Consumer –> Document Responder Interoperability Tests

- MHD_Search_ITI-66
- MHD_SearchRead_ITI-67_ITI-68
- MHD_QryRetr_XDSonFHIR

This is not new - a TEDS talk

IEEE 1451 had TEDS - Electronic Datasheet

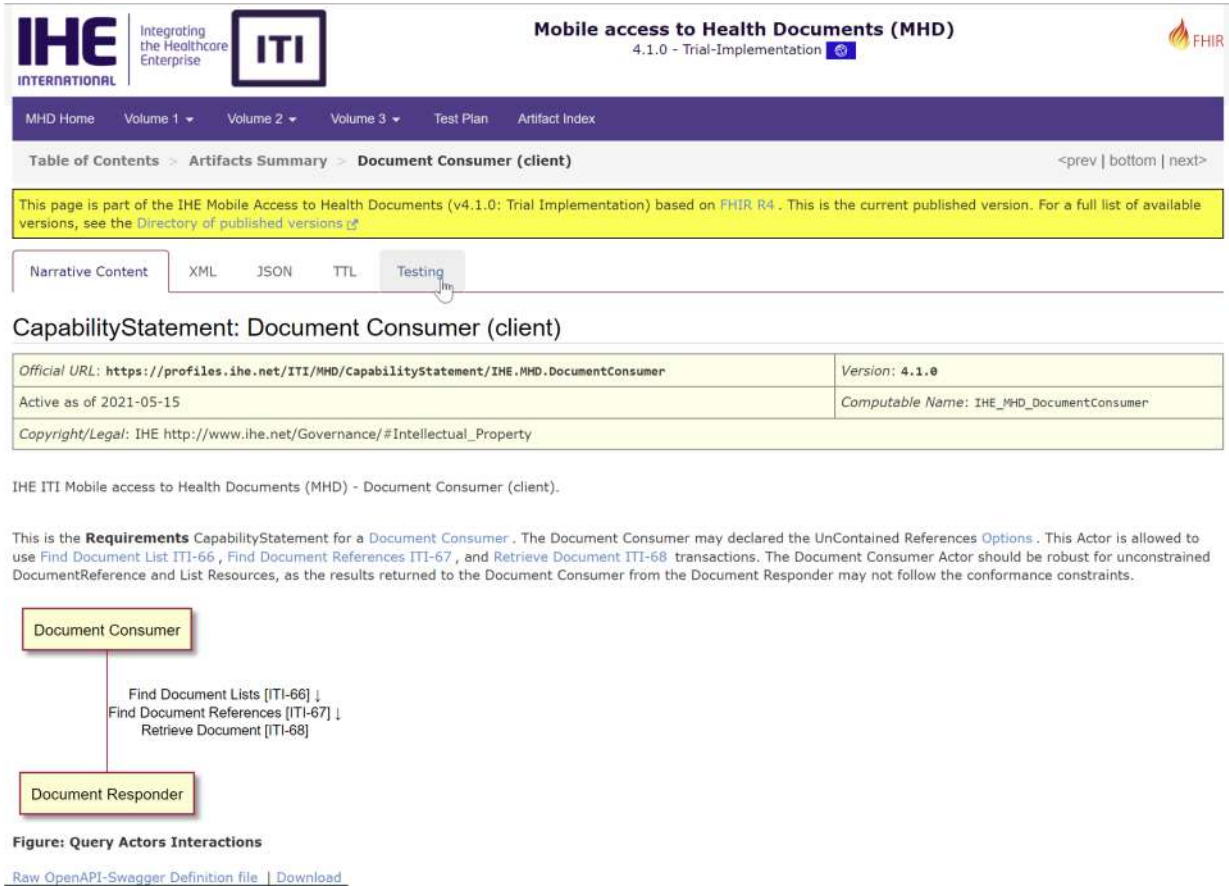
- Allowed systems to “negotiate” their interaction
- Each sensor has their own datasheet
- ⇒ FHIR CapabilityStatement for Actors

FHIR TestScript

Add testability into the specification

Built-in, enforceable, testable

Answering the question: "What makes a good testable IG"



IHE INTERNATIONAL Integrating the Healthcare Enterprise ITI

Mobile access to Health Documents (MHD)
4.1.0 - Trial-Implementation

FHIR

MHD Home | Volume 1 | Volume 2 | Volume 3 | Test Plan | Artifact Index

Table of Contents > Artifacts Summary > Document Consumer (client) <prev | bottom | next>

This page is part of the IHE Mobile Access to Health Documents (v4.1.0: Trial Implementation) based on FHIR R4. This is the current published version. For a full list of available versions, see the Directory of published versions

Narrative Content | XML | JSON | TTL | Testing

CapabilityStatement: Document Consumer (client)

Official URL: https://profiles.ihe.net/ITI/MHD/CapabilityStatement/IHE.MHD.DocumentConsumer	*Version*: 4.1.0
Active as of 2021-05-15	*Computable Name*: IHE_MHD_DocumentConsumer
Copyright/Legal: IHE http://www.ihe.net/Governance/#Intellectual_Property	

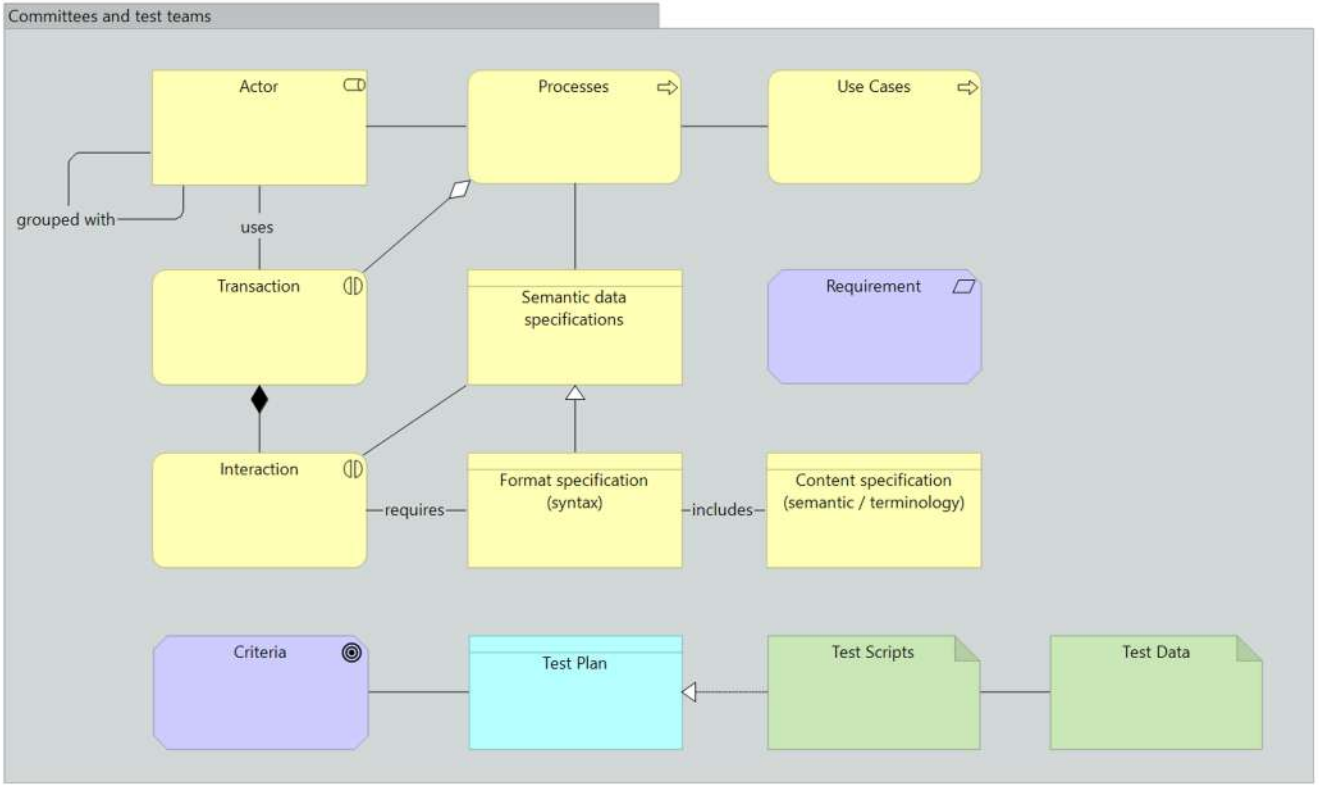
IHE ITI Mobile access to Health Documents (MHD) - Document Consumer (client).

This is the **Requirements** CapabilityStatement for a Document Consumer. The Document Consumer may declared the UnContained References Options. This Actor is allowed to use Find Document List ITI-66, Find Document References ITI-67, and Retrieve Document ITI-68 transactions. The Document Consumer Actor should be robust for unconstrained DocumentReference and List Resources, as the results returned to the Document Consumer from the Document Responder may not follow the conformance constraints.

Figure: Query Actors Interactions

Raw OpenAPI-Swagger Definition file | Download

Testing requires expectations

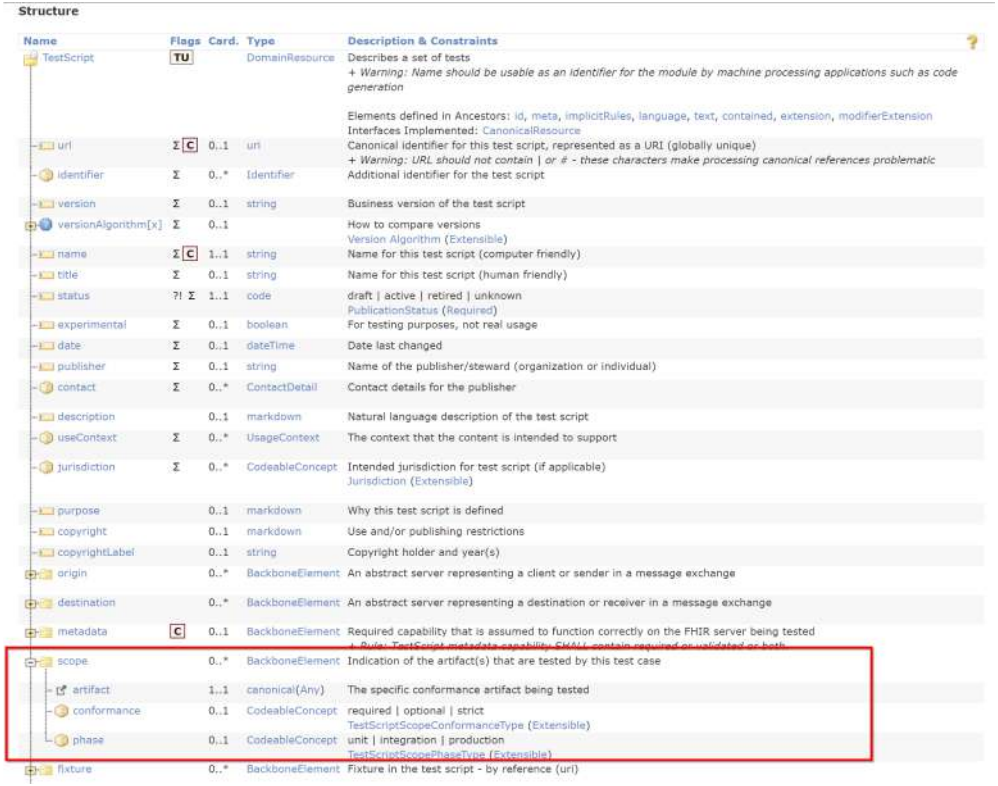


Adding Testing to artifacts

Adding a test case for an artifact:

Create TestScripts with the artifact as scope.

(Artifacts >> narrative)



Structure

Name	Flags	Card.	Type	Description & Constraints
TestScript	TU		DomainResource	Describes a set of tests *+ Warning: Name should be usable as an identifier for the module by machine processing applications such as code generation* Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension Interfaces Implemented: CanonicalResource
url	Σ C	0..1	uri	Canonical identifier for this test script, represented as a URI (globally unique) *+ Warning: URL should not contain | or # - these characters make processing canonical references problematic*
identifier	Σ	0..*	Identifier	Additional identifier for the test script
version	Σ	0..1	string	Business version of the test script
versionAlgorithm[x]	Σ	0..1		How to compare versions Version Algorithm (Extensible)
name	Σ C	1..1	string	Name for this test script (computer friendly)
title	Σ	0..1	string	Name for this test script (human friendly)
status	?! Σ	1..1	code	draft | active | retired | unknown PublicationStatus (Required)
experimental	Σ	0..1	boolean	For testing purposes, not real usage
date	Σ	0..1	dateTime	Date last changed
publisher	Σ	0..1	string	Name of the publisher/steward (organization or individual)
contact	Σ	0..*	ContactDetail	Contact details for the publisher
description		0..1	markdown	Natural language description of the test script
useContext	Σ	0..*	UsageContext	The context that the content is intended to support
jurisdiction	Σ	0..*	CodeableConcept	Intended jurisdiction for test script (if applicable) Jurisdiction (Extensible)
purpose		0..1	markdown	Why this test script is defined
copyright		0..1	markdown	Use and/or publishing restrictions
copyrightLabel		0..1	string	Copyright holder and year(s)
origin		0..*	BackboneElement	An abstract server representing a client or sender in a message exchange
destination		0..*	BackboneElement	An abstract server representing a destination or receiver in a message exchange
metadata	C	0..1	BackboneElement	Required capability that is assumed to function correctly on the FHIR server being tested
scope		0..*	BackboneElement	Indication of the artifact(s) that are tested by this test case
artifact		1..1	canonical(Any)	The specific conformance artifact being tested
conformance		0..1	CodeableConcept	required | optional | strict TestScriptScopeConformanceType (Extensible)
phase		0..1	CodeableConcept	unit | integration | production TestScriptScopePhaseType (Extensible)
fixture		0..*	BackboneElement	Fixture in the test script - by reference (uri)