Agenda & Structure

- House rules, intro, agenda
- 1 Recap:
  - FHIR basics and Profiling
  - Recap: Terminologies
- 2: ImplementationGuides
  - ImplementationGuide: what, when, how
  - FHIR Artifacts
  - Tools and process considerations
- 3. Practical example walkthrough
Remarks and disclaimers

• FHIR® is the registered trademark of Health Level Seven® (HL7®) International.

• The use of the FHIR® trademark does not constitute endorsement of this course/product/service by HL7®.

• This is not an official HL7 training. For such training opportunities, you are encouraged to
  • http://www.hl7.org/training

• This presentation is a collection of freely available materials.
  • All diagrams, except those that state otherwise, are original materials or taken from the FHIR website and support materials; all screenshots are from the FHIR website

• This presentation is shared under a Creative Commons Attribution 4.0 (CC BY 4.0) license - (ok to share and adapt if credits are given)
Goals

1. Recall the basic of FHIR® with regards to Profiling
   1. FHIR (and FHIR profiling) is about technical, computable artifacts
   2. Resources, Data Structures and Data Elements
   3. Terminologies – when to use, ValueSets, CodeSystems

2. Understand what is a FHIR® ImplementationGuide
   1. Concepts
   2. Tools
   3. Practices

3. Develop a simple ImplementationGuide
Setup your toolbox

• We’ll use open-source tools
  • Github account is required except for local experimentation
    • (Local implementation is harder to do, so GitHub account is highly recommended)
  • Github client is recommended – e.g. github desktop
  • Local build (not required if you just want to learn and experiment a little):
    • JAVA (JDK)
    • Jekyll (https://jekyllrb.com/docs/installation)
    • Sushi (https://fshschool.org/docs/sushi/installation) : npm install -g fsh-sushi
    • you need to install node.js if you don’t have it (https://nodejs.org/)
    • May need to update your settings in Windows:
      Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
      https://go.microsoft.com/fwlink/?LinkId=135170
Part 1
Quick recap – FHIR, profiling, terminologies
FHIR Foundations and Profiling
FHIR Profiling

• FHIR® resources and profiles
  • Core resources are represent the common agreed data sets for exchange
  • *Can be Constrained and Extended*

• FHIR Terminologies
  • Define our own valuesets
  • If needed, CodeSystems, etc.

Profiling is done technically – and FHIR has a language for that
Profiling = defining FHIR content

• A “profile” is the name given to a constrained resource in FHIR®.
  • Profiled Resources are derived from other FHIR resources (or from profiles)
  • Example:
    • MedicationPrescriptionLine (profile of MedicationRequest).
    • MyPatient (Profile of USCore Patient)

• In FHIR, everything is defined with resources
  • StructureDefinition resource defines resources
  • ElementDefinition defines the individual data elements
  • ValueSets and CodeSystems define terminologies

http://build.fhir.org/profiling.html
Profiling data structures

• Select the right profile to constrain - from core specification or from existing profiles

• Take one resource as base, (re)define the data elements by adding constraints
  • Change the name
  • Change the cardinality (0..0 removes the element, 1..1 or 1..* makes it mandatory)

• Extensions:
  • Take the Extension resource, add (Differential) constraints to its elements, and define context

• Update terminologies
Profiling in layers

- Profiling FHIR means constraining a FHIR specification
  - Fixing or binding some aspects of the specification
  - Defining which expansions to use
- This allows a layered specification – use it.
ElementDefinition

• Every element has its definition – data type, cardinality, binding…
**StructureDefinition**

- Defines a data structure – a set of elements
  - Snapshot – full structure
  - Differential – difference to base

- Can be used to define Logical Models – an abstract representation of a data structure

---

http://build.fhir.org/structuredefinition
We can extend most anything in FHIR – resources, elements, datatypes
There are many standard extensions already out there:
• HL7: https://build.fhir.org/extensibility-registry.html
• (we can define our own extensions)
Terminologies
Terminology - Coded Data elements

- Meaning and system are implied by the element
- Used in “core” aspects of the specification –
  - Bundle.type
  - Patient.gender

- Version, display, userSelected are useful when implementing your own codes
  - (not very commonly used)

- Same concept can be represented by different codes
- Text to represent the concept (or when a code is not available)
Terminology - Coded Data

- **Code System**
  - Defines a set of concepts with a coherent meaning
  - Code
  - Display
  - Definition
  - e.g. SNOMED CT

- **Value Set**
  - A selection of a set of codes for use in a particular context
  - e.g. “SNOMED CT fracture codes”

- **Element Definition**
  - Data element, binding characteristics and value set reference
  - e.g. Condition.code

- **Element (instance)**
  - Coded Data Type
  - code/
  - Coding/
  - CodeableConcept
  - e.g. 263204007 | Fracture of shaft of ulna|

- **Diagram**
  - Code System selects Value Set
  - Value Set binds Element Definition
  - Element (instance) refers to Code System
  - Element (instance) conforms to Element Definition
CodeSystem and Value Set

- **Code systems** define **symbols** with **specific meanings**
  - E.g. LOINC, SNOMED, ICD-10, IETF language codes, local lab result codes, etc.

- **Value sets** define **collections of codes** for use in a **particular context**
  - Can come from a single code system or multiple code systems
    - E.g. “European country codes”
    - “The LOINC codes that I use”
    - All LOINC order codes
    - A particular SNOMED CT hierarchy
    - Substance codes plus “No known allergy”

**Code System**
- Defines a set of concepts with a coherent meaning
  - Code
  - Display
  - Definition
  - E.g. SNOMED CT

**Value Set**
- A selection of a set of codes for use in a particular context
  - E.g. “SNOMED CT fracture codes”
Terminology Binding

- **example**: These codes just give an idea of what you might use
  No expectation (or recommendation) of use
- **preferred**: You SHOULD use the specified codes
  But if you have a good reason, you can use something else instead – it is not required to use the specified codes in order to be conformant
- **extensible**: You must use the specified codes if they apply
  Free to use other codes or text if the value set doesn’t cover the concept
- **required**: You must use the specified codes
  Or omit the element if no code applies for the concept
Considerations when using profiling
Validate often

• Use examples

• Use Logical Models, discuss with stakeholders

• Set up a way of working, recurring calls, for review
Choose your strength

- Choose cardinalities: Making something mandatory can seem to make data will be “cleaner” but there is risk of information loss because non-compliant messages are rejected

- Same for terminology bindings
ImplementationGuide: What, when, how
ImplementationGuides

• The FHIR community uses ImplementationGuides to transport functional (and technical) requirements into FHIR technical specifications.

• ImplementationGuides and associated tooling produce a web publication that meets the common needs of implementers
  • Data specifications
  • Narratives
  • Examples
  • Etc.
Purpose of Implementation Guides

- Requirements and expectations should define the technical specifications
  - Not the other way around
  - Of course, the existing technical base provides a good starting point – don’t reinvent if not needed; use best practices

- Purpose is important:
  - Be clear about the purpose: Are you defining your system? Or expectations for many systems?
  - When a constraint is applied, it cannot be removed in upper layers.
    - Be flexible with what you accept, strict with what you send.
    - Avoid systems to become non-compliant because of “ideal” constraints.
Example

https://costateixeira.github.io/FoodAllergy-Webinar/
4.2.1 Resource Profile: Food Allergy

**Definition URL:** http://schema.org/Intangible#FoodAllergy

**Version:** 0.1.0

**Name:** FoodAllergy

**Type:** Food Allergy

**Notes:**
- Active as of: 2019-02-24 18:45:36 +00:00
- Description: Food allergy

**Publisher:** Quality of Life Workgroup

**Source:** http://www.nhlbi.nih.gov/health/dci/Da/daa/daa.html

4.3.1 ValueSet: Food Allergens

**Summary:**
- **Defining URL:** http://schema.org/Intangible#FoodAllergy
- **Version:** 1.0.0
- **Name:** FoodAllergy
- **Type:** Food Allergens
- **Status:** active as of: 2019-02-24 18:45:36 +00:00
- **Definition:** Food allergens

**Publisher:** Quality of Life Workgroup

**Source:** http://www.nhlbi.nih.gov/health/dci/Da/daa/daa.html

**References:**
- Food Allergy

**Logical Definition (LOD):**
- Includes these codes as defined in http://www.nlm.nih.gov/medlineplus:
  - Allergy to peanut
  - Allergy to tree nut
  - Allergy to egg
  - Allergy to milk
  - Allergy to shellfish
  - Allergy to fish
  - Allergy to wheat
  - Allergy to soy
  - Allergy to mustard
  - Allergy to fish

**Expansion:**
- Expend based on SNOMED CT International edition: 31 Jan 2026

**All codes from:** http://www.nlm.nih.gov/medlineplus:
FHIR artifacts
ImplementationGuide – a FHIR resource
ImplementationGuide source artifacts

• Creating an ImplementationGuide means to create a set of files that provide the content

• This is why we use Version Control and repositories e.g. github

• ImplementationGuides follow a standard structure
ImplementationGuide source artifacts

- FHIR resources – in json, xml or ttl format
  - Profiles, logical models, valuesets, examples, etc.
- Shorthand resources (in files in dedicated folder)
- Narrative pages in markdown or xhtml format
- Images that are needed for the narrative
ImplementationGuide target artifacts

- Web page publication (as deployable HTML content)
  - Consistent navigation – so that everyone can find their way around others’ IGs
  - Table of contents, menus, artifacts
  - Other technical artifacts e.g. packages for reusing, etc.

- IG URL
Tools and process
Specification process

• Create the Implementation Guide
  • Get requirements - Data definitions, elements, terminologies
  • Create boilerplate / initial content
  • Check the FHIR community for existing guidance or interested people
  • Implement your content – profiles, extensions, terminologies, narrative, examples
    • Always checking if there is already something similar or reusable
  • Build your IG & repeat

• Publish and deploy
Tools needed

• Creating content:
  • (by hand, using any text editor)
  • Forge – free for non-commercial use - https://fire.ly/products/forge
  • FHIR Shorthand – a FHIR specification to type (little) text for profiling https://fshschool.org – allows you to experiment and share

• Publishing
  • Simplifier.net – a tool to produce and discover the ImplementationGuides from the community
  • FHIR Implementation Guide Publisher – an open source standard tool

• Editors and IDEs (for editing text): Visual Studio, Notepad++, ...

• Repository / Version control – GitHub, GitLab, BitBucket..
Build process overview

Source content
- Config
- Text (md, xhtml, puml, ...)
- Computable content (xml, json, fsh)

HL7 FHIR IG Publisher

Template

Offline (Local)

Repository setup

Online

Source content

Template

HL7 FHIR IG Publisher

Static web site

build.fhir.org/ig

pages.github.io

Digital Square | connecting the world for better health
Publication process

• When we iterate through an ImplementationGuide, we want to share with others and keep track

• There are several ways to publish the “Development” version of an IG
  • If the IG is hosted on GitHub: (github.com/<org>/<repo>)
    • HL7 CI Builder publishes it on build.fhir.org/ig/<org>/<repo>
    • The online template provides a workflow which publishes it on <org>.github.io/<repo>
Part 3
Let’s do it
Tools used in this exercise

- Repository: GitHub + online build
- Git client: Github desktop
- IDE: VisualStudio
- Profiling: Sushi
- Template – OpenHIE IG template
- Publishing: ImplementationGuide Publisher – online and offline
Agenda / Steps

• Create an IG and setup IG repository - local and online
  • Sushi --init → upload
  • Online template → download

• Configure & customize IG (if you didn’t use sushi in previous step)

• Add narrative

• Add Logical Model

• Add Profiles

• Add ValueSet

• Add Example

• Check output quality
Creating your IG

- Locally: **sushi --init**
  or
- Remotely: [https://github.com/openhie/empty-fhir-ig-custom](https://github.com/openhie/empty-fhir-ig-custom)
Sync offline - online
1. Change your IG filename

- Just rename the xml
- Make sure you update the ig.ini file that points to it

- (not needed if you use sushi –init)
1. Adapt your IG id, name, etc.

- In the ig.xml, change
  - id
  - url
  - name
  - title
  - publisher
  - contact
  - description
  - packageId

(Not needed if you use sushi --init)
Build

• Locally: Run _genonce.bat / _genonce.sh
  • *The first time you need to download the publisher*
    • just run _updatePublisher.bat / _updatePublisher.sh

• Online: Setup online continuous build
  • If you use the template provided – just check that your repository uses github pages
    • `<org>.github.io/<repo>`

  • If you commit to the online repository, check it out: `build.fhir.org/ig/<org>/</repo>`
    • (instructions on https://github.com/FHIR/auto-ig-builder)
2. Add narrative pages

• Narrative pages can be added by creating markdown or xhtml files – and adding them to the ImplementationGuide resource

  • Create the .md or .xhtml file in the folder input/pagecontent
  • Reference the page in the ig xml
  • (Optionally, add a menu link to that page)

• Add a page to describe “Food Allergy Reporting and Sharing”
• Not forgetting to add it to the ImplementationGuide resource XML
3. Add a Logical Data Model (functional)

- Logical models are StructureDefinitions, based on a special resource (Base)

- Add a .fsh file to your repository
- You can try and share specific shorthand content using https://fshschool.org/FSHOnline/

- Publish for validation
Example content

- FoodAllergy
  - Patient (Mandatory)
  - Clinical status (Mandatory, coded)
  - Verification status (Mandatory, coded)
  - Allergen (Mandatory, coded) – ASK FOR LIST OF CODES
  - Date recorded (if known)
  - Recorder (if known)
  - Asserter (if known)
  - History of known reactions
    - Manifestation (required)
    - Certitude (optional)
    - Exposure route (if known)
    - Note (if exists)

https://fshschool.org/FSHOnline/#/share/3eVQNWY
4. Add a profile

• **StructureDefinitions** are sets of DataElements and their characteristics.
• **StructureDefinitions** can contain a Differential from a base resource or profile

• Select your base profile - [http://hl7.org/fhir/allergyintolerance.html](http://hl7.org/fhir/allergyintolerance.html)
• Create a StructureDefinition that changes some of the elements
  • Cardinality
  • Short description
  • Definition

• More tricks on [https://build.fhir.org/ig/HL7/fhir-shorthand/reference.html](https://build.fhir.org/ig/HL7/fhir-shorthand/reference.html)
  • You can test on FSH
4. Add a profile

• One approach to follow is to ensure that all elements mentioned on the Logical Model are reflected in the profile
  • If element is in there, mark it as “Must Support”
  • If element is not in there, add extension

• If element needs changes, add constraints
  • If constraints are not possible, we can’t change it – just add another element
4. Add a profile – compare with base resource

<table>
<thead>
<tr>
<th>Name</th>
<th>Flag</th>
<th>Card.</th>
<th>Type</th>
<th>Description &amp; Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllergyIntolerance</td>
<td></td>
<td></td>
<td>DomainResource</td>
<td>Allergy or Intolerance (generally: Risk of adverse reaction to a substance)</td>
</tr>
<tr>
<td>Identifier</td>
<td>X</td>
<td>0..*</td>
<td>Identifier</td>
<td>External Id for this item</td>
</tr>
<tr>
<td>ClinicalStatus</td>
<td>M</td>
<td>0..1</td>
<td>CodableConcept</td>
<td>active</td>
</tr>
<tr>
<td>verificationStatus</td>
<td>M</td>
<td>0..1</td>
<td>CodableConcept</td>
<td>unresolved</td>
</tr>
<tr>
<td>type</td>
<td></td>
<td>0..1</td>
<td>code</td>
<td>allergy</td>
</tr>
<tr>
<td>category</td>
<td></td>
<td>0..*</td>
<td>code</td>
<td>food</td>
</tr>
<tr>
<td>uncertainty</td>
<td></td>
<td>0..1</td>
<td>code</td>
<td>low</td>
</tr>
<tr>
<td>code</td>
<td></td>
<td>0..1</td>
<td>CodableConcept</td>
<td>Code that identifies the allergy or intolerance</td>
</tr>
<tr>
<td>patient</td>
<td></td>
<td>1..1</td>
<td>Reference(Patient)</td>
<td>Who the sensitivity is for</td>
</tr>
<tr>
<td>encounter</td>
<td></td>
<td>0..1</td>
<td>Reference_ENCOUNTER</td>
<td>Encounter when the allergy or intolerance was asserted</td>
</tr>
</tbody>
</table>

- **patient** - MS
- **clinicalStatus** - Mandatory
- **verificationStatus** - Mandatory
- **code** – Mandatory
- **recordedDate** MS
- **recorder** MS
- **asserter** MS
- **reaction** MS
  - **manifestation** MS
  - **Certitude** – Need extension
  - **exposureRoute** - MS
  - **note (if exists)** - MS

https://fshschool.org/FSHOnline/#/share/3zGvHU6
5. Add a ValueSet and binding

• ValueSets are for the coded elements

• Choose your strength: this binding should not be required, but extensible or preferred

• We’ll use 12 allergens: Peanuts, tree nuts, milk, eggs, fish, shellfish, soy, sesame seeds, mustard, celery, lupin

• Create and build https://fshschool.org/FSHOnline/#/share/2UQuaMG
6. Add an example

- Examples can also be defined in sushi / shorthand

- Create an instance for a suspected allergy to peanut

https://fshschool.org/FSHOnline/#/share/3eX1sAo
7. Check the QA report

• The QA report shows errors. If you want to deploy this `ImplementationGuide`, these errors must be fixed.
9. Use it

• In a validator

• Deploy a server
Additional notes
More ImplementationGuide content

• These techniques can be used to document more specifications

  • Defining content exchange aggregates with Bundles, Composition, MessageHeader..

  • Defining Operations and Search Parameters

  • Defining other requirements in narrative format
Guidance

• Check out the IG registry: http://fhir.org/guides/registry/

• FHIR Sample IG: https://github.com/FHIR/sample-ig
  • Example content, standard techniques

• FHIR Guidance IG: http://build.fhir.org/ig/FHIR/ig-guidance
  • Changing colors, adding features

• Always check chat.fhir.org
Final Questions and Answers

• Has this answered your questions?
• How do you expect to use ImplementationGuides?
Get in touch, be active

• Check with others (at chat.fhir.org or community.fhir.org)
• Create (or ask someone to create) a change request
• Join a FHIR® event like DevDays (devdays.com), discuss
• Join a FHIR® connectathon, test and provide feedback
Digital Square is supported by:

Digital Square is a PATH-led initiative funded and designed by the United States Agency for International Development, the Bill & Melinda Gates Foundation, and a consortium of other donors.

This presentation was made possible by the generous support of the American people through the United States Agency for International Development. The contents are the responsibility of PATH and do not necessarily reflect the views of USAID or the United States Government.