Profiling in practice (Academy)

Lilian Minne, Firely
Who am I?

- **Name:** Lilian Minne
- **Company:** Firely, Amsterdam
- **Background:**
  - PhD in Medical Informatics
  - FHIR consultant / profiler
- **Contact:**
  - [lilian@fire.ly](mailto:lilian@fire.ly)
Learn profiling with the Profiling Academy

- Launched one year ago
- For anyone willing to learn more about Profiling (beginners and experts)
- 17 modules published since last year!

https://simplifier.net/ui/ig/ProfilingAcademy/ProfilingAcademy
Profiling in practice

- **Design**
  - Data model
  - Map data model to FHIR

- **Build**
  - Search existing profiles and terminology
  - Create profiles and terminology
  - Collaborate

- **Validate**
  - Validate profiles and terminology

- **Share**
  - Create Implementation Guide(s)
  - Publish your work

- **Maintenance**
  - Maintain profiles and terminology
  - Version management
Part 1: Design

• Learn FHIR
• Data model
• Modeling approach
• Naming conventions
• Map to FHIR
Design – Learn FHIR

• Learn the basics
  • Introduction to FHIR and profiling
  • Start Profiling
  • Slicing
  • Extensions

• Design
  • Logical models
  • FHIR mapping language
Design – Data model

• Backbone of business concepts, principles and values
• Common terminology and definitions
• Be aware of (national) standards

• Modeling documents
  • Logical model
  • Data dictionary
  • Data mapping
  • Glossary

Module tip: Logical models
Design – Modeling approach

Open model

Closed model
### Design – Modeling approach

<table>
<thead>
<tr>
<th>Open</th>
<th>Closed</th>
</tr>
</thead>
</table>
| • Generic models, fit more data  
• Larger, vaguer models | • More specific models  
• Smaller, straightforward models |
| • Forward compatibility | • Only backwards compatibility |
| • Needs support of all elements  
• Focus on supported elements | • No need to support all elements  
• More versions of models  
• New elements require new version |
| • Less implementer feedback | • More implementer feedback |
Design – Naming conventions

• Example Nictiz:
  • http://[domain]/fhir/[ConformanceResource]/[project]-[name]-[semver.major]
    • http://fhir.nl/fhir/StructureDefinition/nl-core-patient
    • http://nictiz.nl/fhir/StructureDefinition/zib-Dispense-1.0

• Example Germany:
  • http://fhir.de/StructureDefinition/condition-de-basis
  • http://fhir.de/StructureDefinition/condition-de-icd10gm
## Design - Map to FHIR

<table>
<thead>
<tr>
<th>Mapping facility</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ConceptMap</td>
<td>• Mappings between codes in different ValueSets</td>
</tr>
<tr>
<td>• Mapping field in every element</td>
<td>• Traceability from FHIR elements to other standards</td>
</tr>
<tr>
<td>• FHIR mapping language</td>
<td>• Transformations from one data format to another</td>
</tr>
<tr>
<td>• StructureMap</td>
<td>• FHIR mapping language transformations in resource form</td>
</tr>
</tbody>
</table>

*Module tip: FHIR mapping language*
Design - Map to FHIR

```
<element id="Procedure.basedOn">
  <path value="Procedure.basedOn" />
  <type>
    <code value="Reference" />
    <targetProfile value="http://hl7.org/fhir/StructureDefinition/HelseVestProcedureRequest" />
  </type>
  <mapping>
    <identity value="HelseVest" />
    <map value="Behandlingsnummer" />
  </mapping>
</element>
```

<table>
<thead>
<tr>
<th>Overview</th>
<th>Details</th>
<th>Mappings</th>
<th>Table</th>
<th>XML</th>
<th>JSON</th>
<th>History</th>
<th>Issues</th>
</tr>
</thead>
</table>
Design – Map to FHIR

• Always try to fill in the mapping to the logical mapping

• Use comments field to describe imperfect mappings

<table>
<thead>
<tr>
<th>StructureDefinition.mapping.comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
</tr>
<tr>
<td>Control</td>
</tr>
<tr>
<td>Type</td>
</tr>
</tbody>
</table>

• Use mapping in the profile (not needed in the extensions)
Part 2: Build

• Learn FHIR
• Search existing profiles and terminology
• Create profiles and terminology
• Collaborate
Build – Learn FHIR

• The basics
  • Introduction to FHIR and profiling
  • Start Profiling
  • Slicing
  • Extensions
  • Profiling tools
  • Terminology
  • Best-practices

• Specific topics
  • SDC and questionnaires
  • Contained resources
  • Search operations and parameters

• Advanced stuff
  • Advanced slicing
  • Advanced search parameters
  • Custom constraints
Build – Search for existing profiles / terminology

- FHIR specification
- Simplifier
- FHIR registry
- FHIR IG registry
Build – Use of existing profiles

- Race
- National identifier (BSN)
- Hospital patient number

- US Patient
  - Mayo Clinic Patient
  - NY Presbyterian Patient
- Dutch Patient
  - AMC Patient
  - LUMC Patient
Build – Use of (national) profiles

• Example: nl-core-patient

Dutch national identifier

use of family name prefix in Dutch name, e.g.: “Piet de Vries”
Build – Create profiles / terminology

• Forge
  • Create (derived) profiles, extensions and conformance packages

• Snapper Author
  • Create terminology resources (CodeSystems, ValueSets, ConceptMaps)

• SMART FRED
  • Create example resources

Module tip: Profiling tools
Build – Profile guidelines

• Always fill the narrative with:
  • A short description what it is (profile or extension)
  • What FHIR resource it is profiling or extending
  • One-sentence description what it does
  • E.g. An extension <description> or A profile on <resource> <description>

• Set status to empty
  • As it is neither generated nor additional content

Module tip: Best-practices
Build – Profile guidelines

• Copy-paste description into definition
  • So it shows up at the start of the resource in Details view
Build – Profile guidelines

• When profiling, keep in mind:
  • Searches you want to do later
  • How you want to transport data

• Example Nictiz:
  • ZIB TreatmentDirective
  • ZIB AdvanceDirective

GET [base]/Consent?patient.identifier=[patient-id]&category=http://snomed.info/sct|1129100146105
Build – Profile guidelines

• Coded datatypes
  • Code
    • Bound to non-extensible and required code list
  • CodeableConcept
    • Supports textual representation
    • Contains one or more Codings
  • Coding
    • Use is limited
Build – Profile guidelines

• Complete description in:
  • Extension itself
  • Profile(s) it is used in

• So viewer doesn’t need to dig up extension

```xml
<element id="Procedure.extension:timestamp">
  <path value="Procedure.extension"/>
  <sliceName value="timestamp"/>
  <short value="Waypoint registration"/>
  <definition value="Waypoint registration (e.g. time of first cut / start of surgery)"/>
</element>
```
Build – Profile guidelines

• Always fill in why you create the extension
• Not per se necessary for the profile

• Avoid future confusion due to evolving FHIR landscape:
  • New extensions
  • Updates core specification to cover your use case
  • Resulting in duplicated functionality
Build – Profile guidelines

• Constrain extensions at profile level
• Make them as generic and reusable as possible
Build – Profile guidelines

• Provide isModifier flag at extension level
• Once created, you know it effects the meaning of the resource you apply it in
Build – Profile guidelines

• Preferred binding on extension level
  • Choose binding level in profile level
  • Extension is more reusable this way

• Place binding on value[x] not extension itself

• Selecting type of reference
  • Explicit ValueSet: Reference(ValueSet), preferably canonical URL
  • Implicit ValueSet: uri
Build – Collaborate

• Simplifier project
• Github integration
• Resource visualization
• Issue tracker

Module tip: Get started with Simplifier
Part 3: Validate

• Validate profiles and terminology
Validate profiles and terminology

- .NET GUI validator
- Forge
- Simplifier
- Snapper Author
- FHIR server

Module tip: Publishing and validating your work
Part 4: Share

- Create Implementation Guide(s)
- Publish your work
Share – Create Implementation Guide(s)

• Describe how to implement usecase(s)

• Include at least:
  • Data definitions
  • Use cases
  • Actors
  • Interactions
  • Examples

Module tip: Publishing and validating your work
Share – Create Implementation Guide(s)

- IG-editor (Simplifier)
- IG Publisher
- Other formats:
  - WiKi page
  - Custom website
  - PDF
  - …
Share – Publish Implementation Guide(s)

- Simplifier repository
- Linked to official HL7 FHIR registry
- Organizations, projects, resources
Part 5: Maintenance

• Maintenance
• Version management
Maintenance – Maintenance

• Transfer IT management
• Ownership
• Simplifier project
• Github integration
• Issue Tracker
• Package management
Maintenance – Version management

- Packages
- Simplifier (beta release)
- Forge
- Implementation Guide(s)

Module tip: Packages (coming soon!)
Summary

- **Design**
  - Data model
  - Map data model to FHIR

- **Build**
  - Search existing profiles and terminology
  - Create profiles and terminology
  - Collaborate

- **Share**
  - Create Implementation Guide(s)
  - Publish Implementation Guide(s)

- **Validate**
  - Validate profiles and terminology

- **Maintenance**
  - Maintain profiles and terminology
  - Version management
Thank you for your attention!

Any questions?

Share your thoughts here: https://simplifier.net/ProfilingAcademy/~issues

Contact me: lilian@fire.ly