Article <u>Toshihiko Minamoto</u> · Feb 17, 2022 16m read

Let's create an FHIR profile using SUSHI Part 1

Hello, developers!

In this series, I will not show you how to use IRIS for Health, but rather how to use SUSHI, a tool for creating FHIR profiles, as an associated technology.

With the right tools, the profile information (specifications, limitations, extensions, etc.) of a FHIR project can be well organized and published.

Before we begin, what is SUSHI? I will briefly explain it.

What is FHIR?

FHIR is an acronym for Fast Healthcare Interoperability Resources and is defined as a standard for healthcare information exchange that can be implemented in a short period. REST is a standard technology for web communication to exchange a set of data (resources) in JSON/XML format that is highly readable and easy to handle.

In simple terms, the idea is to use a common format to represent medical data to facilitate the transfer and exchange of information between systems and facilities.

There are various <u>resources</u> defined in FHIR. For example, patient information has a definition called <u>Patient</u> <u>resource</u>, which represents the patient's information.

There are many <u>samples</u> on the FHIR official website, so I thought I'd excerpt some of them.

For example, the data is expressed in JSON format like this. Patient number (Identifier), name, gender, etc., are represented.

```
{
  "resourceType": "Patient",
  "id": "pat1",
  "text": {
    "status": "generated",
    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">\n
                                                                  ∖n
                                                                           Patient Do
nald DUCK @ Acme Healthcare, Inc. MR = 654321\n
                                                               </div>"
                                                         ∖n
  },
  "identifier": [
    {
      "use": "usual",
      "type": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
            "code": "MR"
          }
        ]
      },
```

```
"system": "urn:oid:0.1.2.3.4.5.6.7",
      "value": "654321"
    }
  ],
  "active": true,
  "name": [
    {
      "use": "official",
      "family": "Donald",
      "given": [
        "Duck"
      1
    }
  ],
  "gender": "male",
  "* snip *"
}
```

What is the FHIR profile?

In FHIR, there are JSON and XML representation formats and rules on what information should be listed with JSON key name, what kind of code should be used, and what type of structure should be used. This is called the FHIR <u>Profile</u>.

Profile is a term that is used in many different ways.

In the broadest sense:

A set of definitions for constraints on FHIR resources and FHIR servers. An artifact that represents it.

In a narrower sense:

In the narrower sense, a conformance resource is a resource to which specific constraints have been applied.

In this case, a profile exists for each resource (e.g., Patient profile, Observation profile).

For details, please watch the Japan Virtual Summit 2021 video on FHIR profiles here. (about 20 minutes).

Although the official FHIR website provides default specifications for each resource, the degree of freedom in the use of each resource is very high. Unfortunately, there is a significant degree of freedom in the way each resource is used, and it is difficult to achieve interoperable data exchange with the resources as they are. Therefore, it is important to define new rules for the description of resources based on the prior "agreement". This "agreement" and "rules" are equivalent to the Implementation Guide and Profile.

Implementation guidelines are largely written in text using Word, Excel, HTML, etc. One of the features of FHIR is that the FHIR profile itself can also be expressed using FHIR resources. For example, it is possible to express specs in JSON format so that products such as IRIS for Health can incorporate the definitions and extend the functions. On the other hand, the profile is written in JSON format, which can be processed using FHIR's StructureDefinition resource.

The HL7 Association of the United States has released IG Publisher, which automatically generates implementation guidelines from Profiles. Using this tool, you can generate HTML files of implementation guidelines in the format published by the HL7 Association. The other half of this article will show you how to do it.

As an example, this is a resource called "StructureDefinition" that represents the notational conventions for Patient

resources suggested for standard use in the US, called US Core. (<u>Reference</u>)

```
{
  "resourceType" : "StructureDefinition",
  "id" : "us-core-patient",
  "text" : {
    "status" : "extensions",
    "div" : "(snip)"
  },
  "url" : "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient",
  "version" : "3.1.1",
  "name" : "USCorePatientProfile",
  "title" : "US Core Patient Profile",
  "status" : "active",
  "experimental" : false,
  "date" : "2020-06-27",
  "publisher" : "HL7 US Realm Steering Committee",
  "contact" : [
    {
      "telecom" : [
        {
          "system" : "url",
          "value" : "http://www.healthit.gov"
        }
      ]
    }
  ],
  "(snip)"
```

There are also other resources that describe the FHIR profile, such as the <u>ImplementationGuide</u> and the <u>CapabilityStatement</u> that summarizes the series of functions of the FHIR server.

What is FHIR Shorthand?

To create a FHIR profile, all you have to do is complete the JSON structure shown above! However, it would be difficult and troublesome to do this manually. I would probably make a mistake.

There are applications and tools available to assist in this process, including several commercial products and open source options.

See this page.

One example is the well-known Forge from Firely in the Netherlands, although more recently a domain-specific language for defining FHIR artifacts called FHIR Shorthand (<u>link</u>) has also become widely used.

FHIR Shorthand is a language that allows you to create a FHIR profile while creating a definition file = FSH(fish) file (example)

A sample FSH file is shown below. The <u>quote</u> contains the name of this profile (Profile: CancerDiseaseStatus), the original FHIR resource on which it is based (Parent: Observation), and the specification of the element whose cardinality is to be changed (bodySite 0..0). The following is a list of the features.

```
Alias: LNC = http://loinc.org
Alias: SCT = http://snomed.info/sct
```

```
Profile:
          CancerDiseaseStatus
Parent:
          Observation
Id:
          mcode-cancer-disease-status
Title:
          "Cancer Disease Status"
Description: "A clinician's qualitative judgment on the current trend of the cancer,
e.g., whether it is stable, worsening (progressing), or improving (responding)."
* ^status = #draft
  extension contains EvidenceType named evidenceType 0..*
* extension[evidenceType].valueCodeableConcept from CancerDiseaseStatusEvidenceTypeVS
 (required)
* status and code and subject and effective[x] and valueCodeableConcept MS
* bodySite 0..0
* specimen 0..0
* device 0..0
* referenceRange 0..0
* hasMember 0..0
* component 0..0
* interpretation 0..1
* subject 1..1
* basedOn only Reference(ServiceRequest or MedicationRequest)
 (snip)
```

What is SUSHI?

After explaining the FHIR/FHIR Profile/FHIR Shorthand, it is time to explain SUSHI.

SUSHI (an acronym for "SUSHI Unshortens SHorthand Inputs") (4) is a reference implementation of a FSH compiler that translates FSH into FHIR artifacts such as profiles, extensions, and value sets. SUSHI is installed on your own computer and runs locally from the command line (<u>Reference</u>).

In short, when SUSHI processes the FSH(fish) file describing the FHIR Shorthand mentioned earlier, files such as StructureDefinition will be generated.

Here's one illustration, both obvious and somewhat obscure, showing how this works! (Citation)



(In this picture, it is described as processing (i.e., compiling) the fish to make sushi, but the SUSHI compiler does the compiling, and the finished product is FHIR artifacts such as profiles, which is a bit different.)

I hope you have a better understanding of what SUSHI is after this introduction.

Let's go to the FSH School

Instead of explaining how to install and basically use SUSHI here, I would recommend you to visit the official site where you can find a very precise introduction. The name of the site is <u>FSH School</u>.

The resources (JSON files) such as StructureDefinition are generated by using SUSHI, and by using the "IG Publisher tool" also introduced in this site, you can generate the HTML source that combines them. You can also use the "IG Publisher Tool," also presented on this site, to generate the HTML source for all of them.

First of all, we recommend that you follow the contents of this <u>SUSHI Tutorial</u> to check the basic functions.

If you have problems, you can refer to the FishExampleComplete directory, which is the complete version of the FSH Tank(!) included in the downloadable folder.

I tested it in a Windows environment, and since I didn't have Node.js installed, I used the information on this site to help me!

Also, as the tutorial states below, you will need a tool called Jekyll to output HTML files using IG Publisher.

Warning

Before proceeding to the next command: If you have never run the IG Publisher, you may need to install Jekyll first. See Installing the IG Publisher for details.

You can get the Jekyll kit from this site.

SUSHI execution example

The following are the results of running the SUSHI command using the completed version of the tutorial in my environment.

For more information about the command, etc., please visit this site (Running SUSHI).

```
>sushi .
info Running SUSHI v1.2.0 (implements FHIR Shorthand specification v1.1.0)
info Arguments:
info
       C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\fsh-tutorial-
master\FishExampleComplete
info No output path specified. Output to .
info Using configuration file: C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\fsh-
tutorial-master\FishExampleComplete\sushi-config.yaml
info Importing FSH text...
info Preprocessed 2 documents with 3 aliases.
info Imported 4 definitions and 1 instances.
info Checking local cache for hl7.fhir.r4.core#4.0.1...
info Found hl7.fhir.r4.core#4.0.1 in local cache.
info Loaded package hl7.fhir.r4.core#4.0.1
(node:26584) Warning: Accessing non-
existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:26584) Warning: Accessing non-
existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency
info Converting FSH to FHIR resources...
info Converted 3 FHIR StructureDefinitions.
info Converted 1 FHIR ValueSets.
info Converted 1 FHIR instances.
```

Let's create an FHIR profile using SUSHI Part 1 Published on InterSystems Developer Community (https://community.intersystems.com)

info Exporting FHIR resources as JSON... info Exported 5 FHIR resources as JSON. info Assembling Implementation Guide sources... info Generated ImplementationGuide-fish.json info Assembled Implementation Guide sources; ready for IG Publisher. ? Profiles ? Extensions ? ValueSets ? CodeSystems ? Instances ? ? ? 1 0 ?? 2 ? 1 ? ? ? 1 ? ? ? ? It doesn't get any betta than this! 0 Errors 0 Warnings ?

Once you run it, it will compile the FSH file and display how many Profiles and Extensions were generated in the process. If everything is fine, only "info" will be printed, otherwise "Warning and Error" will be displayed. The error messages are relatively friendly and make it easy to figure out what the problem is.

After its execution, you can confirm that the JSON file of StructureDefinition has been successfully generated in the fsh-generated folder in the project folder.

Following that, let's try to obtain the IG Publisher tool by using the "updatePublisher" command, start IG Publisher by using the "genonce" command, and also generate a group of HTML files. Since the execution log is long, I'll skip it.

After the execution, check the output folder in the same project folder, and you will see that many files have been generated. And if you open the index.html file, you will see that the following page has been developed.

The resource description pages that you are used to seeing on the FHIR official site are also automatically generated.

Let's write an Implementation Guide

Although I have only been working with this set of tools for a short time, I would like to show you how to write an implementation guide as a simple start-up.

For more detail on how to use them, please refer to the information on the FSH School website. You may also find the <u>FHIR DevDays slides and other information</u> very helpful.

To begin, use the sushi --init command to create a template for your project structure.

C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject>sushi --init

Let's create an FHIR profile using SUSHI Part 1 Published on InterSystems Developer Community (https://community.intersystems.com)

```
Status (Default: draft):
Version (Default: 0.1.0):
Initialize SUSHI project in C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject\M
yFirstSUSHIProject? [y/n]: y
Downloading publisher scripts from https://github.com/HL7/ig-publisher-scripts
(node:13972) Warning: Accessing non-
existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:13972) Warning: Accessing non-
existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency
? Project initialized at: ./MyFirstSUSHIProject
? Now try this:
                                                 ?
?
                                                 ?
? > cd MyFirstSUSHIProject
                                                 ?
? > sushi .
                                                 ?
?
                                                 ?
? For guidance on project structure and configuration see
                                                 ?
? the SUSHI documentation: https://fshschool.org/docs/sushi ?
```

After running it, it will generate the minimum required configuration and FSH files. Now let's make some modifications.

Next, let's make some modifications.

Before that, I would like to introduce the editor; I recommend using Visual Studio Code, as there is an <u>extension</u> available to modify fsh files.

Since I'm going to be doing this, I'd like to input some Japanese information and see how it is reflected.

First, modify sushi-config.yaml.

The title of the implementation guide is added, the menu screen is changed to Japanese, and the contents list page (tuc.html) and the custom page (mycustompage.html) are added.

sushi-config.yaml

```
# ? The properties below are used to create the ImplementationGuide resource. For a
list of
          ?
# ? supported properties, see: https://fshschool.org/sushi/configuration/
          ?
id: myfirstsushi
canonical: http://example.org/myfirstsushi
name: MyFirstSUSHIProject
# Add a title to be displayed at the top of the page.
title: ??FHIR Project Implementation Guide
status: draft
publisher: InterSystems Japan/S.Kaminaka
description: This is a sample implementation guide for the FHIR project using SUSHI.
version: 0.1.0
```

Let's create an FHIR profile using SUSHI Part 1 Published on InterSystems Developer Community (https://community.intersystems.com)

fhirVersion: 4.0.1 copyrightYear: 2021+ releaseLabel: ci-build # ? To use a provided input/includes/menu.xml file, delete the "menu" property below # Configure the menu to be displayed in Japanese. menu: Implementation Guide Home: index.html Contents List: toc.html FHIR Artifact Summary: artifacts.html Custom page: mycustompage.html Index pages and custom pages can be written in markdown: index.md # MyFirstSUSHIProject Feel free to modify this index page with your own awesome content! ### Project Background pagecontent/index.md You can modify the file to change the description in the html file.

To describe a page, you can use markdown notation.

Links to reference information
 (omitting)

mycustompage.md

This is a custom page.

If you provide a markdown file, an html file will be generated.

Project-specific pages can be produced and included in the implementation guide.

Finally, we will modify the most critical FSH file. This template contains the FSH file for the Patient profile, so we changed it slightly.

patient.fsh

// This is a simple example of a FSH file. // This file can be renamed, and additional FSH files can be added. // SUSHI will look for definitions in any file using the .fsh ending. Profile: MyPatient Parent: Patient

```
Title: "Patient profile for ?? project"
* name 1..* MS
// The description part of the list view can be changed by changing the ^short.
* name ^short = "Contains the patient's name.?"
* name ^definition = "Element to store the patient's name. Kanji and Katakana based o
n NeXEHRS JP CORE"
```

Now execute the following command to see the generated implementation guide page.

sushi.

<u>updatePublisher</u>

genonce

You can easily generate a page that contains the following information.

It also generates JSON files such as StructureDefinition and ImplementationGuide.

Summary

Were you satisfied with the results?

This tool generates JSON files of FHIR profiles and can create HTML files, so if this tool is used well, it will be easy to create content that can communicate FHIR specifications in an easy-to-understand manner.

Although this tool itself is not directly related to InterSystems, we thought it would be helpful to introduce it to the developer community to exchange information about the FHIR project.

For those of you who have tried it out after reading this article, or are already familiar with it, I would be happy to hear your thoughts on how to use it, as well as any useful features or syntax you'd like to share.

In the following article of this series, I would like to extend the search parameters by reading the definition file of the Search Parameter generated by SUSHI in IRIS for Health.

(2021/4/20 We have corrected the ambiguous part of the explanation, especially about the profile. Thank you for pointing it out.)

#FHIR #InterSystems IRIS for Health

Source URL: https://community.intersystems.com/post/lets-create-fhir-profile-using-sushi-part-1