

記事

[Shintaro Kaminaka](#) · 2021年4月19日 18m read

SUSHIを使ってFHIRプロファイルを作成しようパート1

開発者の皆さん、こんにちは。

このシリーズでは、IRIS for Healthの使い方ではなく、関連技術として、FHIRプロファイル作成ツールであるSUSHIの握り方使い方を紹介していきたいと思います。

このツールをうまく使うことで、FHIRプロジェクトのプロファイル情報（仕様や制限、拡張などの情報）をうまく整理し、公開することができます。

その前にSUSHIとは何でしょうか？簡単にですが、順番に説明していきたいと思います。

FHIR って？

FHIRとは Fast Healthcare Interoperability Resources の略であり、Web通信の一般的技術であるRESTを使用して

医療情報交換標準規格、という定義になっています。

簡単に言えば、医療のデータの表現方法として皆で共通したフォーマットを使うことによって、システム間や施設間などでの情報の伝達や交換をやりやすいようにしよう！ということですね。

FHIRには様々な[リソース](#)が定義されています。例えば患者さんの情報には[Patientリソース](#)という定義があり、これを使って表現されます。

FHIR公式サイトには多くの[サンプル](#)が掲載されていますので、一部抜粋してみます。例えばこのようなJSON形式で表現されます。患者番号(Identifier)、氏名(name)、性別(gender)などが表現されています。

```
{
  "resourceType": "Patient",
  "id": "pat1",
  "text": {
    "status": "generated",
    "div": "<div xmlns=\\"http://www.w3.org/1999/xhtml\\">\n      \n      <p>Patient Donald DUCK @ Acme Healthcare, Inc. MR = 654321</p>\n      \n    </div>"
  },
  "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"
    ]
  }
],
"gender": "male",
"???"
}
```

FHIRプロファイルって？

FHIRではJSONやXMLという表現形式だけではなく、何の情報をどのようなJSONキー名称で記載するか、どのようなコードを使用するか、どのような構造で表現するかといった決まりが存在します。それを[FHIRプロファイル](#)と呼んでいます。

プロファイルは用語として色々な意味で使われています。

広義では・・・

FHIRリソースおよびFHIRサーバに関する制約の定義の集合。それを表すアーティファクト(成果物)。

狭義では・・・

あるリソースに対して、特定の制約を適用したコンフォーマンス・リソース(適合性リソース)。
この場合、プロファイルはリソース単位に存在する(例)Pateintプロファイル、Observationプロファイル...

詳細については、こちらの[FHIRプロファイルに関するJapan Virtual Summit 2021動画](#)をご覧ください。(約20分)

FHIRの公式Webサイトでは各リソースについて既定の仕様が開示されています。ですが、各リソースの使い方の自由度がとても高く、そのままでは実際に相互運用性のあるデータ交換をすることは困難です。ですので、事前の「申し合わせ」にもとづいて、リソースの記述に「規則」を改めて設ける必要があります。この「申し合わせ」「規則」に相当するのは、実装ガイドライン(Implementation Guide)とプロファイル(Profile)に相当します。

実装ガイドラインは主に文章で記述されたもので、WordやExcel、HTML等でも記述されています。一方、ProfileはFHIRのStructureDefinitionリソースを使って計算可能なJSON形式で記述をしています。このFHIRのプロファイル自体もFHIRのリソースで表現できる、というのがFHIRの特徴の一つでもあります。例えばIRIS for Healthのような製品でその定義を取り込んで、機能拡張ができるように、JSON形式で仕様まで表現できるようになっているのです。

実装ガイドラインは様々なツールで作れますが、米国HL7協会はProfileから実装ガイドラインを自動的に生成するIG Publisherを公開しています。このツールを使えば、米国HL7協会が出しているフォーマットで実装ガイドラインのHTMLファイル等を生成することができます。この記事後半ではその方法についても紹介しています。

例えばこれは、US Coreと呼ばれる米国で標準的に使用されることが推奨されたPatientリソースの記法上の規約を表現した、「StructureDefinition」というリソースです。

([引用元](#))

```
{
  "resourceType" : "StructureDefinition",
  "id" : "us-core-patient",
  "text" : {
    "status" : "extensions",
    "div" : "???"
  },
  "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"
        }
      ]
    }
  ]
}
```

FHIRプロファイルを表現されるリソースとしては他にも[ImplementationGuide](#)やFHIRサーバの一連の機能をまとめた[CapabilityStatement](#)などがあります。

FHIR Shorthand とは？

というわけで、ではFHIRプロファイルを作成するには のJSON構造を作っていけばいいんだな！？ということになる訳ですが、これを手作業でやるのはどう考えて難しいし煩雑ですよ。間違えそうです。

これを補助するためのアプリケーションやツールが公開されており、商用製品やオープンソースなどいくつかの選択肢があります。

こちらの[ページ](#)をご覧ください。

例えばオランダのFirely社のForgeなど是有名ですが、最近ではFHIR Shorthand ([リンク](#))という

FHIRアーティファクトを定義するためのドメイン固有の言語も広く使われるようになってきています。

FHIR Shorthandは言語の一種であり、例えば以下のような定義ファイル=FSH(フィッシュ)ファイルを作成しながら、FHIRプロファイルを作成することができます。

以下にサンプルのFSHファイルを例示します。[引用元](#) 例えば、このプロファイルの名前(Profile:

CancerDiseaseStatus)、ベースとなる元のFHIRリソース(Parent:

Observation)、カーディナリティを変更するエレメントの指定(bodySite 0..0)などの内容を含んでいます。

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)

(??)

SUSHIって？

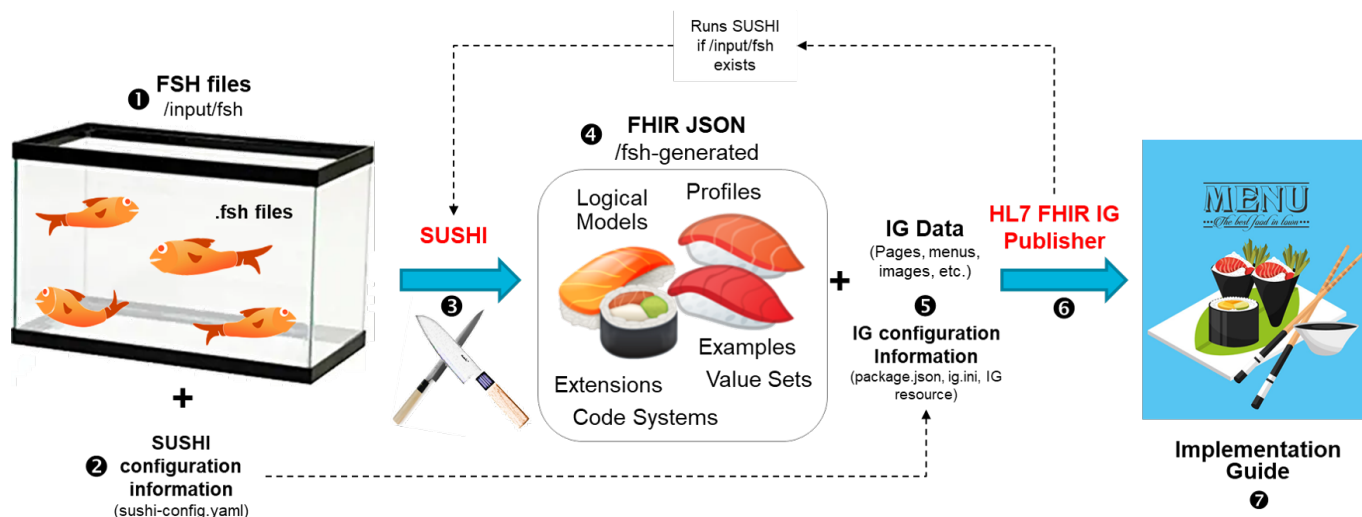
FHIR/FHIRプロファイル/FHIR Shorthandと順に説明してきましたが、ついにSUSHIの説明です。

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. ([引用元](#))

(訳)SUSHI ("SUSHI Unshortens SHorthand Inputs "の略) は、FSHファイルをプロファイル、エクステンション、バリューセットなどのFHIRアーティファクトに変換するFSHコンパイラのリファレンス実装である。SUSHIは自分のコンピュータにインストールされ、コマンドラインからローカルに実行される。

つまり、先ほどのFHIR Shorthandを記述したFSH(フィッシュ)ファイルを、SUSHIで処理すると、StructureDefinitionなどのファイルが生成される、ということです。

この仕組みを説明したわかりやすいようで、ちょっとわかりにくい一枚の絵があります！ [引用元](#)



(この絵では、魚を処理(=コンパイル)して、寿司ができるようなイメージで書かれてるんですが、実際はコンパイルをしているのが「SUSHIコンパイラ」で、できあがるのは「プロファイルなどのFHIRアーティファクト」なのでちょっと違うと思うんですよね・・・。お寿司の説明にProfilesやExtensions等の記載はありますけども。)

駆け足でSUSHIとは何か、までご紹介してきましたがご理解いただけたでしょうか？

FSH Schoolに行こう

肝心のSUSHIのインストール方法や基本的な使い方ですが、それらについてはここで説明するよりも、非常に丁寧に紹介されたオフィシャルサイトがありますので、そちらをご紹介したいと思います。

その名も[FSH School](#)です。

SUSHIを使って生成されるのは、StructureDefinitionなどのリソース(JSONファイル)ですが、同じくこのサイトで紹介されている、「IG

Publisherツール"を使うことによって、それらを取りまとめたHTMLのソースまで生成することができます。

まずはこの[SUSHI Tutorial](#)に内容に沿って、基本的な機能の確認をされることをお勧めします。

うまくいかないときは、ダウンロードできるフォルダに含まれる完成版FSH Tank(!)であるFishExampleCompleteディレクトリを参照されると良いと思います。

私はWindows環境で試しました

。Node.jsもインストールされていなかったもので、[こちらのサイト](#)の情報を参考にさせていただきました。

また、チュートリアルに以下の記載がある通り、IG Publisherを使ったHTMLファイルの出力には Jekyll というツールが必要になります。

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.

こちらの[サイト](#)からJekyllのキット等は入手できます。

SUSHI実行例

私の環境で、チュートリアルの完成版を使ったsushiコマンド実行結果を掲載します。

コマンド等の詳細はこちらのサイト([Running SUSHI](#))をご覧ください。

```
>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.
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.

???????????????????????????????????????? SUSHI RESULTS ?????????????????????????????????????
? ??????????????????????????????????????????????????????????????????????????????????????????
? ? Profiles ? Extensions ? ValueSets ? CodeSystems ? Instances ? ?
? ??????????????????????????????????????????????????????????????????????????????????????????
? ? 2 ? 1 ? 1 ? 0 ? 1 ? ?
? ??????????????????????????????????????????????????????????????????????????????????????????
? ?
????????????????????????????????????????????????????????????????????????????????????????
? It doesn't get any better than this! 0 Errors 0 Warnings ?
????????????????????????????????????????????????????????????????????????????????????????
```

実行するとFSHファイルのコンパイルが実行され、最後にいくつかのProfilesやExtensionが生成されたか、表示されます。問題なければ、"info"だけが表示されますが、FSHファイルの定義に誤りがあると、WarningやErrorも表示されます。エラーメッセージは比較的親切で何が問題が把握しやすいと思います。(個人的には最後の表に掲載される、なぞの「魚一言？」みたいな一文が楽しみです。)

実行後には、プロジェクトフォルダ内のfsh-generatedフォルダにStructureDefinitionのJSONファイルが生成されているのが確認できます。

続いて、「updatePublisher」コマンドで、IG Publisherツールを入手し、「genonce」コマンドでIG Publisherを起動し、HTMLファイル群も生成してみます。この実行ログは長いので割愛します。

実行後、同じプロジェクトフォルダ内の output フォルダを確認すると多くのファイルが生成されているのがわかります。index.htmlファイルを開くと以下のようなページが生成されていることが確認できます。

このようなFHIR公式サイトでも見慣れた、リソースの説明ページなども自動生成されます。

Implementation Guide(実装ガイド)を書いてみよう

私もこのツール群に触り始めて日が浅いですが、簡単なスタートアップとして、実装ガイドを記述していく方法をご紹介します。

詳細な使い方についてはFSH Schoolサイト内の情報をご覧ください。 [こちら](#)で紹介されているのFHIR DevDaysのスライド等も大変参考になるとと思います。

まず sushi --init コマンドでプロジェクト構造のひな型を作りましょう。

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

????????????????????????????????????????????????????????????????
? This interactive tool will use your answers to create a      ?
? working SUSHI project configured with your project's        ?
? basic information.                                          ?
????????????????????????????????????????????????????????????????

Name (Default: ExampleIG): MyFirstSUSHIProject
Id (Default: fhir.example): myfirstsushi
Canonical (Default: http://example.org): http://example.org/myfirstsushi
Status (Default: draft):
Version (Default: 0.1.0):
Initialize SUSHI project in C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject\MyFirstSUSHIProject? [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 ?
????????????????????????????????????????????????????????????????
```

実行すると必要最低限の設定ファイルやFSHファイルが作成されます。
次に少し修正をしてみましょう。

その前にエディタの紹介です。fshファイルを修正するための[Extension](#)が公開されているので、Visual Studio Codeの使用がおすすめです。

せっくなので、日本語の情報を入力してどのように反映されるか見ていきたいと思います。

まず、sushi-config.yaml を修正します。実装ガイドのタイトルを追加し、メニュー画面も日本語表記に変更した上

で、コンテンツ一覧ページ(tuc.html)とカスタムページ(mycustompage.html)を追加しています。

sushi-config.yaml

```
# ?????????????????????????????????????????????????????????ImplementationGuide????????????????????????????????
????????????????????
# ? 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
# title????????????????????????????????????????????
title: ??FHIR?????? ?????
status: draft
publisher: InterSystems Japan/S.Kaminaka
description: SUSHI????FHIR????????????????????????????
version: 0.1.0
fhirVersion: 4.0.1
copyrightYear: 2021+
releaseLabel: ci-build

# ?????????????????????????????????????????????????????????menu.xml????????????????????????????????
????????????????????
# ? To use a provided input/includes/menu.xml file, delete the "menu" property below
.
?
# ?????????????????????????????????????????????????????????????????????????????????????????????????
????????????????????
# ?????????????????????????????????
menu:
  ?????????: index.html
  ?????????: toc.html
  FHIR?????????????: artifacts.html
  ?????????: mycustompage.html
```

インデックスページや、カスタムページはマークダウンで記述できます。

index.md

```
# MyFirstSUSHIProject

Feel free to modify this index page with your own awesome content!

### ??????????

pagecontent/index.md ??????????html????????????????????

????????????????????????????????????

### ??????????
(?)
```


mycustompage.md

```
## ????????????????
```

```
????????????????????????html????????????????
```

```
????????????????????????????????????????
```

最後に最も重要なFSHファイルを修正します。このひな形には、Patientプロファイル用のFSHファイルが含まれているので、それを少しだけ修正しました。

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?????"
* name 1..* MS
// ^short????????????????????????????
* name ^short = "?????????????"
* name ^definition = "????????????????NeXEHR JP CORE?????????"
```

それでは、以下のコマンドを実行して、生成された実装ガイドページを確認してみましょう。

```
sushi.
```

```
updatePublisher
```

```
genonce
```

以下のような情報を含むページが簡単に生成できます。

あわせてStructureDefinitionやImplementationGuideなどのJSONファイルももちろん生成されています。

まとめ

いかがでしたでしょうか？

単にFHIRプロファイルのJSONファイルを生成するだけでなく、HTMLファイルを生成する機能も付随しているので、このツールをうまく使えばFHIRの仕様を分かりやすく伝えることが可能なコンテンツが簡単に作成できるのではないかと考えています。

このツール自体はInterSystemsとは直接関係ない製品ではありますが、FHIRプロジェクトの情報交換に役立つツールということで、この開発者コミュニティでご紹介させていただきました。

この記事を見て試していただいた方、あるいはすでに使いこなしている方、使い方のコツや便利な機能やシンタックスなどをご紹介いただけると嬉しいです。

今回のこのシリーズでは、SUSHIで生成したSearch Parameterの定義ファイルをIRIS for Healthで読み込んで、検索パラメータを拡張する内容に取り組んでみたいと考えています。

(2021/4/20 特にプロファイルの説明について、説明が曖昧な箇所をご指摘いただきましたので修正しました。ご指摘ありがとうございました。)

[#FHIR](#) [#InterSystems](#) [IRIS for Health](#)

ソースURL:

<https://jp.community.intersystems.com/post/sushi%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6fhir%E3%83%97%E3%83%AD%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%82%92%E4%BD%9C%E6%88%90%E3%81%97%E3%82%88%E3%81%86%E3%83%91%E3%83%BC%E3%83%881>