記事
Shintaro Kaminaka · 2021年11月3日 15m read

SUSHIを使ってFHIRプロファイルを作成しようパート2

開発者の皆さん、こんにちは。

この記事は、FHIRの関連技術として、FHIRプロファイル作成ツールであるSUSHIの使い方を紹介するシリーズの第2弾です。パート2である今回まで半年の期間が経ってしまいました。

前回のパート1では、FHIRとは？FHIRプロファイルとは？FHIR Shorthandとは？そしてSUSHIとはどのようなツールなのか？どのような物を作成できるのか？について、サンプルの成果物のスクリーンショットを交えながら説明しました。

今回の記事では、SUSHIで作成したプロファイルの実際の活用例として、SUSHIを使ってPatientリソースに **Extension** を追加し、さらにそのExtensionの項目に対する新しい **SearchParameter** を定義し、IRIS for HealthのFHIR Repositoyで新しいSearchParameterが使えるようになるまで、をご紹介します。

SUSHIのアップグレード

いきなり本筋とそれて恐縮ですが、SUSHIを久しぶりに触る私のような方はSUSHIのアップグレードを行いましょう。
この半年の間もSUSHIは精力的な機能Enhancementが行われており、8月にはversion 2.0.0がリリースされています。この記事の執筆段階の最新バージョンは SUSHI 2.1.1 でした。

このリンク先でも紹介されている通り、アップグレードはインストール同様以下のコマンドです。

```
$ npm install -g fsh-sushi
```

sushi -versionを実行すればバージョンが確認できます。
同様に、SUSHIで生成されたProfileをベースに、実装ガイドのHTMLファイル群を作成してくれるIG Publisherツールも、**updatePublisher** コマンドを実行してアップグレードすることができます。

FISHファイルの作成

まずは、前回同様 sushi --init
コマンドを使って、プロジェクトを作成します。この記事ではテンプレートで生成される、**patient.fsh** ファイルを修正していきます。

今回は、出身都道府県を表現するstring型の **birthPlace** のExtensionを追加し、さらにそのbirthPlaceに対するSearchParameterも定義することで、その患者の出身都道府県で検索できるような拡張を行います！

Extensionを追加する

まず、Extensionを追加するために以下の定義を追加します。

US CoreやJP Coreのように、通常はAddress型を使うことが多いですが、ここでは単純にstring型にしています。

```
Extension: BirthPlace
Id: birthPlace
Title: "???"
Description: "???????string??????"
* ^url = "http://isc-demo/fhir/StructureDefinition/patient-birthPlace"
* value[x] only string
```

各項目は以下のようにExtensionのStructureDefinitionに対応しています。項目によっては複数の箇所に設定されます。ベースとなるfhirのバージョンや、このExtension自体のバージョンなどの情報は、sushi-config.ymlファイルから取得されているものもあります。

SUSHIの項目	対応するStructureDefinitionの項目
Extensions	name
Id	id
Title	title/differencial.element[id=Extension].short
Desctiption	description/differencial.element[id=Extension].definition
^url	url//differencial.element[id=Extension.url].fixedUri
value[x]	differencial.element[id=Extension.value[x]].type.code

実際に生成されたExtensionのStructureDefinitionです。
手書きでこれを１から作るのは大変ですが、SUSHIを使えば比較的簡単です。

```
{
  "resourceType": "StructureDefinition",
  "id": "birthPlace",
  "url": "http://isc-demo/fhir/StructureDefinition/patient-birthPlace",
  "version": "0.1.0",
  "name": "BirthPlace",
  "title": "???",
  "status": "active",
  "description": "???????string??????",
  "fhirVersion": "4.0.1",
  "mapping": [
    {
      "identity": "rim",
      "uri": "http://hl7.org/v3",
      "name": "RIM Mapping"
    }
  ],
  "kind": "complex-type",
  "abstract": false,
  "context": [
    {
      "type": "element",
      "expression": "Element"
    }
  ],
  "type": "Extension",
  "baseDefinition": "http://hl7.org/fhir/StructureDefinition/Extension",
  "derivation": "constraint",
  "differential": {
    "element": [
      {
        "id": "Extension",
        "path": "Extension",
        "short": "???",
        "definition": "???????string??????"
      },
      {
```

```
        "id": "Extension.extension",
        "path": "Extension.extension",
        "max": "0"
      },
      {
        "id": "Extension.url",
        "path": "Extension.url",
        "fixedUri": "http://isc-demo/fhir/StructureDefinition/patient-birthPlace"
      },
      {
        "id": "Extension.value[x]",
        "path": "Extension.value[x]",
        "type": [
          {
            "code": "string"
          }
        ]
      }
    ]
  }
}
```

このExtensionで追加したPatientリソースへのExtensionデータは、実際はこのようなデータになります。

```
  "extension": [
    {
      "url": "http://isc-demo/fhir/StructureDefinition/patient-birthPlace",
      "valueString": "???"
    }
  ],
```

SearchParamterを追加する

次は、先ほど追加したExtensionの項目をキーにして、リソースを検索できるように、**SearchParamter**を追加します。FHIRの場合、各リソースには構造化された要素(エレメント)が定義されていますが、**そのすべての要素で検索ができるわけではなく、SearchParamterに定義された項目(要素)でのみ検索することができます**。ここがSQLのテーブルとは少し異なる点ですね。

SearchParamter名は要素名とは別に定義されており、Patientリソースで言えば、genderのように要素名=SearchParameter名で一致するものもあれば、要素名＝address.country -> SearchParamter名＝address-country
のように構造化された要素では一致しないものもあります。

Extensionに追加される項目は当然ながら(何がはいってくるかわからないので)デフォルトではSearchParameterにはならないわけですが、あえてExtensionを定義して格納する方針を定めるようなExtensionは重要な項目であることも多いですよね。

以下のようなSearchParameter定義を生成するための内容をpatient.fshファイルに追加します。

```
Instance: BirthPlaceSearchParameter
InstanceOf: SearchParameter
Usage: #definition
* url = "http://isc-demo/fhir/SearchParameter/patient-birthPlace"
```

```
* version = "0.0.1"
* name = "birthPlace"
* status = #active
* description = "???????????"
* code = #birthPlace
* base = #Patient
* type = #string
* expression = "Patient.extension.where(url='http://isc-
demo/fhir/StructureDefinition/patient-birthPlace').value"
* comparator = #eq
```

SearchParameterで生成されるStructureDefinitionはこちらです。
比較的シンプルな定義なので、上記SUSHIの情報とのマッピングは理解しやすいと思います。

```
{
  "resourceType": "SearchParameter",
  "id": "BirthPlaceSearchParameter",
  "url": "http://isc-demo/fhir/SearchParameter/patient-birthPlace",
  "version": "0.0.1",
  "name": "birthPlace",
  "status": "active",
  "description": "???????????",
  "code": "birthPlace",
  "base": [
    "Patient"
  ],
  "type": "string",
  "expression": "Patient.extension.where(url='http://isc-
demo/fhir/StructureDefinition/patient-birthPlace').value",
  "comparator": [
    "eq"
  ]
}
```

特にSearchParameterの定義として、重要になるのは **expression** の項目と **comparator** になります。
expressionには対象となるSearchParameterへの **FHIRPath**
式を記述します。FHIRPathも詳しく
説明すると長くなるので興味のある方はこちらの公式ページをご覧ください。

今回の定義で使っている

Patient.extension.where(url='http://isc-demo/fhir/StructureDefinition/patient-birthPlace').value"

こちらの式は、Patientリソースの Json構造に従って、階層順にPatient.extensionと指定し、複数存在する可能性があるExtensionの中から、url=(省略) である今回のExtensionを絞り込み、そのvalueを指定しています。

comparatorはどのような比較式が使えるかを指定します。詳細はこちらをご覧下さい。

Patientに作成したExtension定義を追加する

もう一つ大事な変更があります。Patientリソースでこの作成した BirthPlace Extensionを追加することです。元々自動生成されたPatientリソースのProfile定義MyProfileを以下のように変更します。name要素のCardinalityの変更はコメントアウトしました。

```
Profile: MyPatient
Parent: Patient
Description: "An example profile of the Patient resource."
//* name 1..* MS
* extension contains BirthPlace named birthPlace 0..1
```

先ほど追加した"BirthPlace"という名前のExtensionを、Patientリソース内にbirthPlaceという名前でCardinality 0..1で追加しています。

ついでにテスト用リソースを作成

SUSHIでは、例示用などの目的で使用できるリソースのInstanceを作成することもできます。テストのためにこちらも利用しておきましょう。今定義したExtensionも含めることができます。

```
Instance: KamiExample
InstanceOf: MyPatient
Description: "Patient?????????"
* name.family = "??"
* extension[BirthPlace].valueString = "???"
```

どんなデータができたかは最後のテストでご覧いただきたいと思います。

Let's SUSHI!

FSHファイルの用意ができました！それでは
SUSHIコマンドで、fshファイルから各定義ファイルを生成しましょう！
sushi コマンドを実行し、以下のように2つのProfile(拡張されたPatientとExtension）、二つのInstance（SearchParameterとサンプルリソース）が生成されたら成功です。

```
C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject\MyProfileProject>sushi .
info  Running SUSHI v2.1.1 (implements FHIR Shorthand specification v1.2.0)
info  Arguments:
info    C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject\MyProfileProject
info  No output path specified. Output to .
info  Using configuration file: C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProje
ct\MyProfileProject\sushi-config.yaml
info  Importing FSH text...
info  Preprocessed 1 documents with 0 aliases.
info  Imported 2 definitions and 2 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:27132) 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:27132) Warning: Accessing non-
existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency
info  Converting FSH to FHIR resources...
info  Converted 2 FHIR StructureDefinitions.
info  Converted 2 FHIR instances.
info  Exporting FHIR resources as JSON...
info  Exported 4 FHIR resources as JSON.
```

```
info  Assembling Implementation Guide sources...
info  Generated ImplementationGuide-myprofileproject.json
info  Assembled Implementation Guide sources; ready for IG Publisher.

??????????????????????? SUSHI RESULTS ???????????????????????????
? ?????????????????????????????????????????????????????????????? ?
? ?    Profiles   ?  Extensions  ?   Logicals   ?   Resources   ? ?
? ?????????????????????????????????????????????????????????????? ?
? ?       1       ?      1       ?      0       ?       0       ? ?
? ?????????????????????????????????????????????????????????????? ?
? ?????????????????????????????????????????????????????????????? ?
? ?      ValueSets     ?    CodeSystems    ?     Instances      ? ?
? ?????????????????????????????????????????????????????????????? ?
? ?         0          ?         0         ?         2          ? ?
? ?????????????????????????????????????????????????????????????? ?
?                                                                  ?
????????????????????????????????????????????????????????????????
? FSHing for compliments? Super job!     0 Errors      0 Warnings ?
????????????????????????????????????????????????????????????????

C:\Users\kaminaka\Documents\Work\FHIR\SUSHI\TestProject\MyProfileProject>
```

fsh-generated\resource フォルダには以下のような成果物が作成されました。

ファイル名	内容
ImplementationGuide-myprofileproject.json	今回の全ての内容を取りまとめたImplemamtionGuide
StructureDefinition-MyPatient.json	PatientにExtensionを追加したStructureDefinition
StructureDefinition-birthPlace.json	Extension birthPlaceの定義を含むStructureDefinition
SearchParameter-BirthPlaceSearchParameter.json	birthPlace SearchParameterの定義ファイル
Patient-KamiExample.json	Patientのサンプルインスタンス

IRIS for HealthにFHIR Profileをインポートしてテストしてみる

IRIS for Health のFHIRリポジトリへの適用

前回の記事ではこの後、updatePublisherを実行してIGファイル群を生成しましたが、今回は、このStuructureDefinitino/SearchParameterファイルをIRIS for HealthのFHIRリポジトリに取り込んで、新しいSearchParameterで検索できるようになるところを見ていきましょう。

FHIR Profileのインポート等について詳細は、こちらの開発者コミュニティ記事 FHIRプロファイル
をご覧章ください。
FHIRリポジトリの構築方法などは、こちらの記事も参考になると思います。

インポートの対象となるのは、先ほど生成された5つのファイルのうち、
- StructureDefinition-MyPatient.json
- StructureDefinition-birthPlace.json
- SearchParameter-BirthPlaceSearchParameter.json

の3つです。これを別のフォルダにコピーし、さらにパッケージ全体の情報を管理するための
package.jsonファイルを用意します。

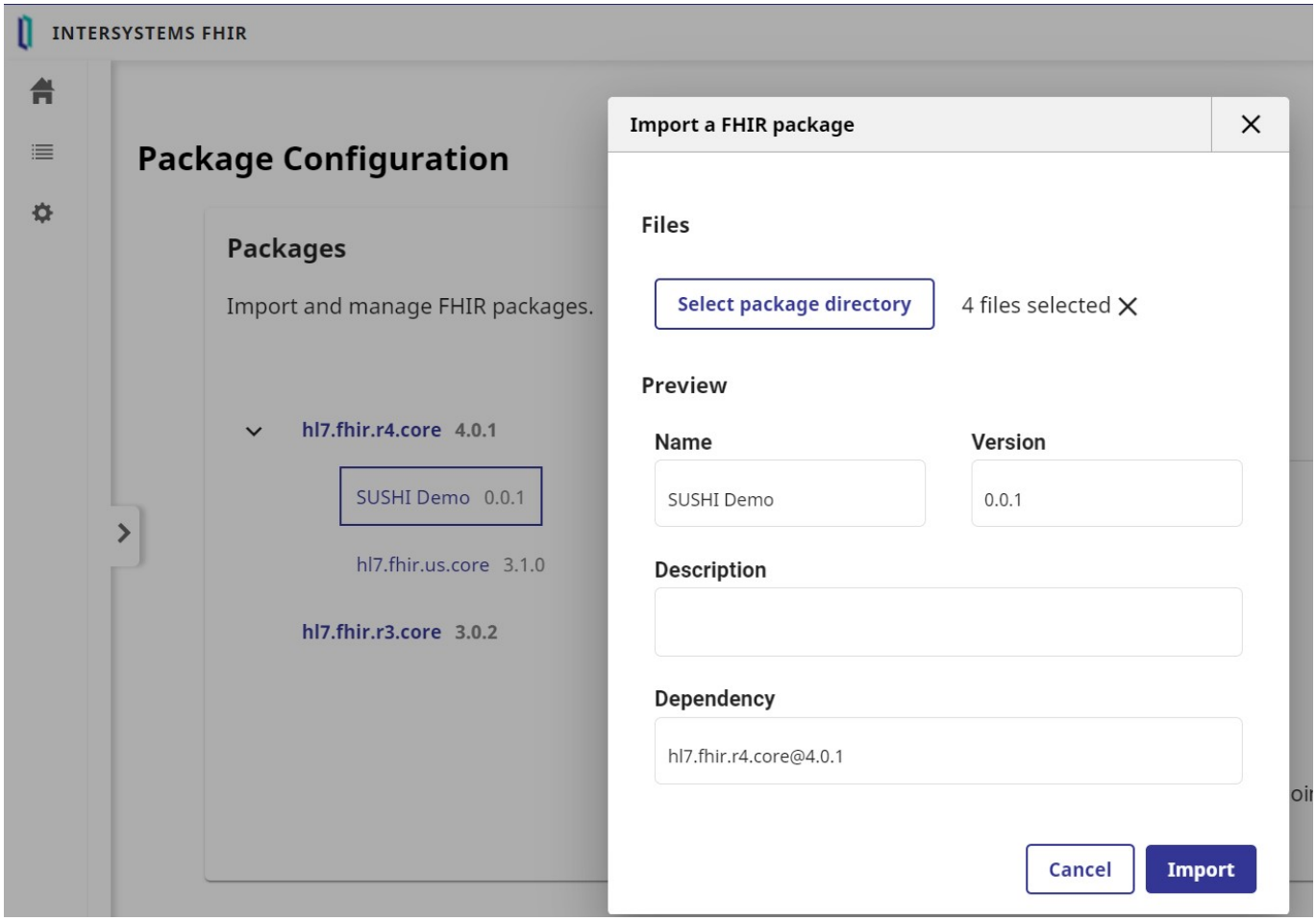
package.json

```
{
  "name": "SUSHI Demo",
  "title": "SUSHI Demo",
  "version": "0.0.1",
```

```
  "author": {
    "name": "ISC"
  },
  "fhirVersions": [
    "4.0.1"
  ],
  "bundleDependencies": false,
  "date": "20201208205547",
  "dependencies": {
    "hl7.fhir.r4.core": "4.0.1"
  },
  "deprecated": false
}
```

nameやtitle,author,dateなどの項目は適宜変更して問題ありません。
（注意）各プロファイルを変更してIRISに再インポートする場合は、versionを適切に変更していく（上げていく）必要があります。
（現在のバージョン2021.1ではFHIRリポジトリには、プロファイルを削除する機能がないため、テスト環境で適切に動作確認した上で、本番環境への適用は最小の回数に抑えるなど、本番環境等でプロファイルが増えすぎないように注意をする必要があります。）

IRISの管理ポータルからHealth -> FHIR Configuration -> Package Configurationと進み、Import Packageから上記4ファイルを含むフォルダを選ぶと以下のような画面になります。



Importをクリックして、IRISへのインポートを完了します。

次に Server Configuration画面で、新規FHIRリポジトリを作成します。

(既存のFHIRリポジトリへ追加することも可能です。)

POSTMANからテストする

先ほどSUSHIで生成された、テスト用リソースをPOSTします。検証のためには他の値のbirthPlaceを含むデータや、そもそもbirthPlaceを含まないPatientリソースなども生成するほうが良いかもしれません。

FHIRリポジトリのSearchParameter に正しく
birthPlaceが追加されていれば、以下のGETリクエストでこの患者情報を取得できるはずです！

```
GET http://localhost:52785/csp/healthshare/sushi/fhir/r4/Patient?birthPlace=???
```

正しく結果を取得できるようになったでしょうか？
新しいSearchParameterである birthPlaceが正しく追加されていない場合は、GETリクエストの応答の最初に以下の「birthPlaceというパラメータが認識されていません」というエラー情報をが記述されたOperationOutcomeリソースの情報が含まれています。このメッセージがでていないか応答メッセージを確認してみてください。

```
{
     "resource": {
         "resourceType": "OperationOutcome",
         "issue": [
             {
                 "severity": "error",
                 "code": "invalid",
                 "diagnostics": "<HSFHIRErr>ParameterNotSupported",
                 "details": {
                     "text": "Unrecognized parameter 'birthPlace'. ???"
                 }
             }
         ]
     },
     "search": {
         "mode": "outcome"
     }
 },
```

まとめ

SUSHIを使ってFHIRのProfile(StructureDefinition/SearchParameter)を作成し、IRIS for
HealthのFHIRリポジトリにインポートして機能を拡張する流れをみていただきました。
今回は、Extensionに追加した項目をSearchParameterに追加しましたが、FHIR標準仕様で存在するが、SearchParameterにはなっていない要素(エレメント)に対して、SearchParameterを追加するということも可能です。

自由度の高いFHIRの開発では、このように機能を拡張することが可能になっていますが、一方ではInteroperabilityを担保するためにどのような拡張を行ったかという情報の共有、つまりImplemantationGuide等の作成も重要になってきます。
このシリーズのPart1,2で見てきたようにSUSHIはその両面をカバーすることができる非常にユニークで強力なオープンソースのツールです。

このようなツールとIRIS for
Healthを組み合わせて、新しいFHIRソリューションが構築されることを期待しています。

今回の記事で使用したSUSHIのfshファイルおよび、生成されたStructureDefinition/SearchParameterのサンプルフ

ァイルはこちらからダウンロードすることができます。

#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%882