

# HL7-FHIR⇔MML

## 相互エクスチェンジ と実用化への課題

2024/5/17

ラジエンスウェア株式会社

山田勝之

# 内閣府 先端的サービスの開発・構築に関する調査事業

令和4年度に続きまして、令和5年度については  
データ連携パターンの拡充として  
テーマ2 電子カルテのデータエクステンションにおける  
MML-HL7 FHIRの相互変換を行った

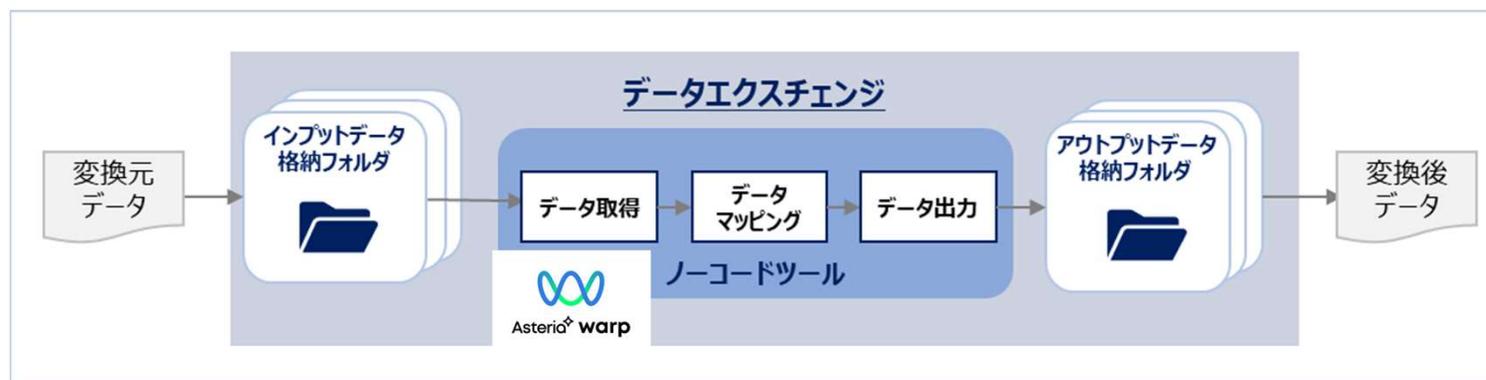


対象データは前年度同様、患者基本・診断・処方・注射・検査結果

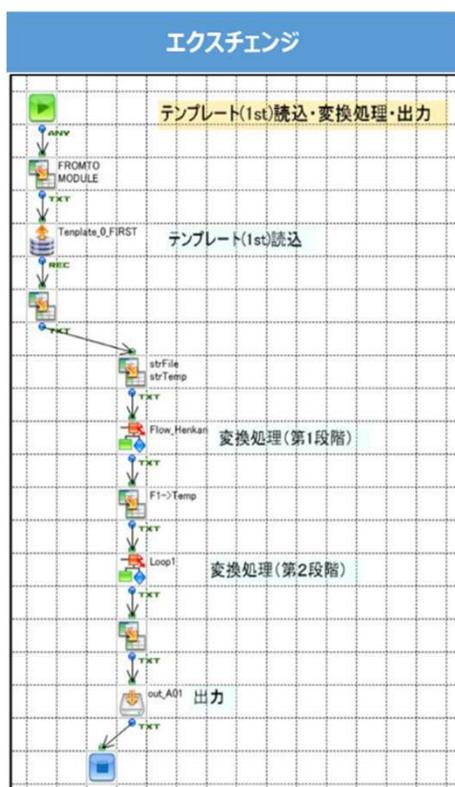
データ種別 (文書単位)	対象MMLモジュール	対象FHIRリソース (JP Core プロファイル)
患者基本	mmlPi:PatientModule	Patient
病名	mmlRd:RegisteredDiagnosisModule	Condition
処方	mmlPs:PrescriptionModule	MedicationRequest
注射	mmlInj:InjectionModule	MedicationRequest_injection
検査	mmlLb:TestModule	Observation_LabResult

## データエクステンジの基本的な動作

1. インพุットフォルダに格納されたファイルを取込
2. 取り込んだデータをasteria warpを用いてターゲットとするデータ形式に変換
3. アウトプットフォルダへ変換後ファイルを出力



令和4年度事業において試作・開発を行ったデータエクステンジと同様に、多様なデータ形式の取込やデータ項目の変化への対応を容易にする目的でノーコードツールによるグラフィカルコーディングを主体とした開発を行った。



```

<content>
  <mmIPs:PrescriptionModule>
    <mmIPs:issuedTo>external</mmIPs:issuedTo>
    <mmIPs:medication>
      <mmIPs:medicine>
        <mmIPs:name>・ワソラン錠40mg</mmIPs:name>
        <mmIPs:code mmIPs:system="local">VAS02110</mmIPs:code>
      </mmIPs:medicine>
      <mmIPs:dose>3</mmIPs:dose>
      <mmIPs:doseUnit>錠</mmIPs:doseUnit>
      <mmIPs:frequencyPerDay>3</mmIPs:frequencyPerDay>
      <mmIPs:startDate>2022-03-29</mmIPs:startDate>
      <mmIPs:duration>P98D</mmIPs:duration>
      <mmIPs:instruction>1日3回朝昼夕食後 03-29から98日分(投与実日数)</mmIPs:instruction>
      <mmIPs:PRN>false</mmIPs:PRN>
      <mmIPs:form>錠剤</mmIPs:form>
      <mmIPs:batchNo>1</mmIPs:batchNo>
      <mmIPs:brandSubstitutionPermitted>true</mmIPs:brandSubstitutionPermitted>
      <mmIPs:longTerm>true</mmIPs:longTerm>
    </mmIPs:medication>
  </mmIPs:PrescriptionModule>
  
```



```

"intent": "order",
"medicationCodeableConcept": {
  "coding": [
    {
      "system": "urn:oid:1.2.392.200119.4.403.1",
      "code": "VAS02110",
      "display": "・ワソラン錠40mg"
    }
  ]
},
"subject": {
  "identifier": [
    {
      "system": "urn:oid:1.2.392.100495.20.3.51.1##<医療機関ID>##",
      "value": "##<患者ID>##"
    }
  ]
},
"authoredOn": "2022-03-29T09:23:28",
"dosageInstruction": [
  {
    "doseQuantity": {
      "value": 3,
      "unit": "mg",
      "text": "錠"
    },
    "frequency": "3 times a day",
    "aspects": [
      {
        "code": "P98D",
        "text": "1日3回朝昼夕食後 03-29から98日分(投与実日数)"
      }
    ]
  }
]
  
```

## 課題1 保守性等向上

ノーコードツール（ASTERIA WARP）のグラフィカルコーディングだけでなくテキストコーディングでの実装を組み込むみ保守性の高い構造を検討する。

構造解析の部品などはテキストコーディングも併用したコンポーネント化を検討し、HL7 FHIRとMMLのデータマッピングを行い、MMLからHL7 FHIRへの変換と逆変換を実施する。

## 課題2 可読性・処理速度の向上

コンポーネント化をする事で取り扱う部品数を減少させフロー（ノーコードツールで作成するプログラム）の可読性向上に取り組む。

さらに、ノーコードツールのログ情報等からデータエクスチェンジでの変換開始から終了までの処理において、処理時間の改善を目指す。

## 保守性向上と可読性、処理速度等向上手法

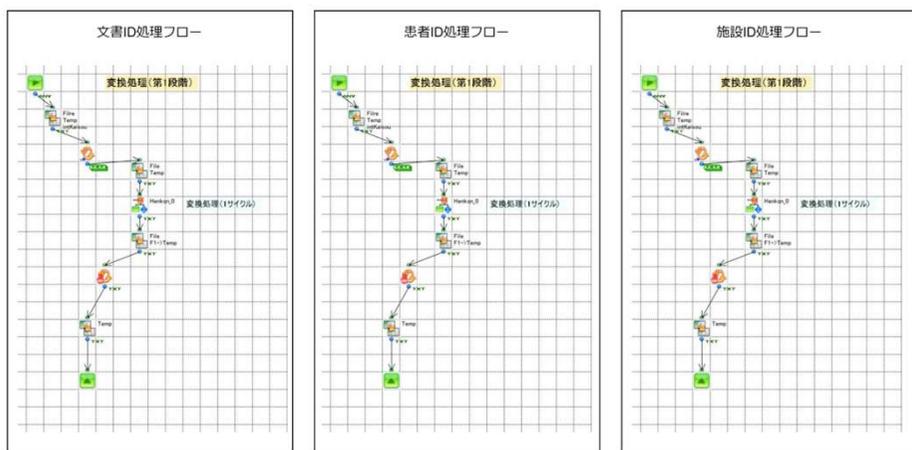
令和4年度事業において課題であった複雑なデータ形式の取扱いに改善を加える。  
構造解析や繰り返し処理等のグラフィカルコーディングではフロー（ノーコード上のプログラム）が複雑になってしまう機能について、テンプレート化とコンポーネント化の2つの手段で保守性が向上するように改修を加えることとした。



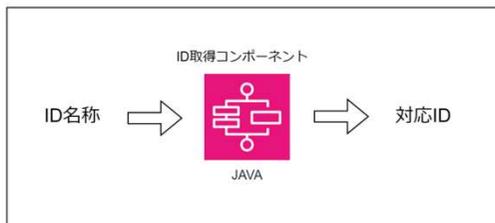
# コンポーネント化

項目情報の取得や、定型的なデータ変換処理についてJava言語によるコンポーネント化を行い、データエクスチェンジ処理フローを簡易化させる

ノーコードツールによるID取得フロー



↓  
コンポーネント化



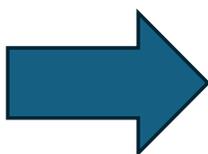
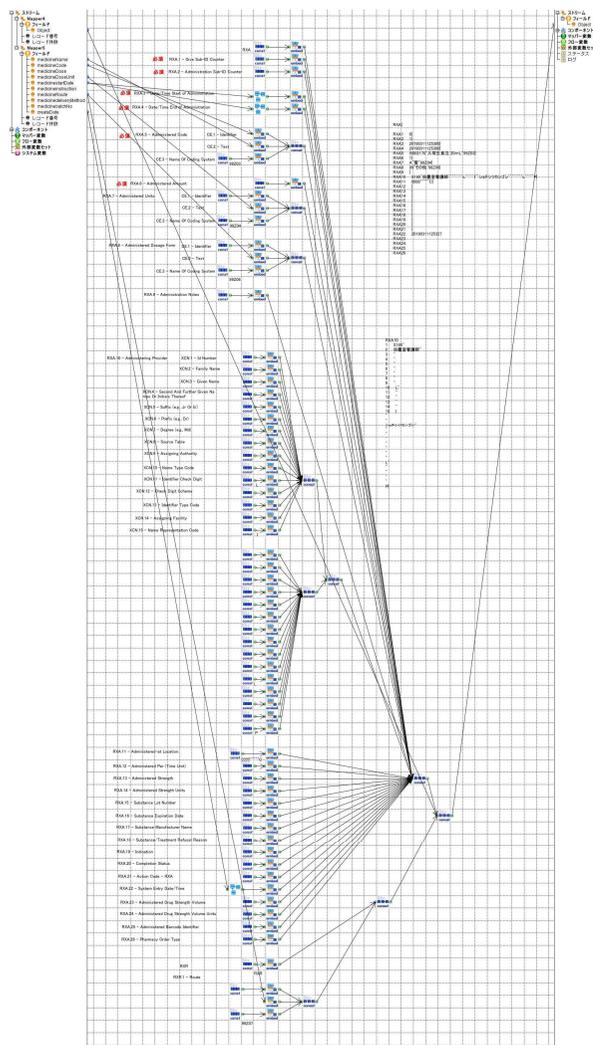
繰り返し行われる処理をコンポーネント化、及びノーコードツールでは煩雑になる処理をJAVAにて記述パーツ化する。

```
import com.infoteria.asteria.flowengine2.compile.CompileException;
import com.infoteria.asteria.flowengine2.execute.ExecuteContext;
import com.infoteria.asteria.flowlibrary2.mapper.Function;
import com.infoteria.asteria.flowlibrary2.mapper.MapperException;
import com.infoteria.asteria.license.LicenseException;
import com.infoteria.asteria.value.Value;

public class fun_mml_fhir01 extends Function {
    private static final String FUNCTION_NAME = "rad.fun_mml_fhir01";
    private static final String PROPERTY_MDB_PATH = "MDBPath";
    public String getFunctionName() { return FUNCTION_NAME;}
    protected void internalInit() {
        registerProperty(PROPERTY_MDB_PATH, Value.TYPE_STRING, false);
    }
    public int getMinInputCount() { return 1;}
    public int getMaxInputCount() { return 1;}

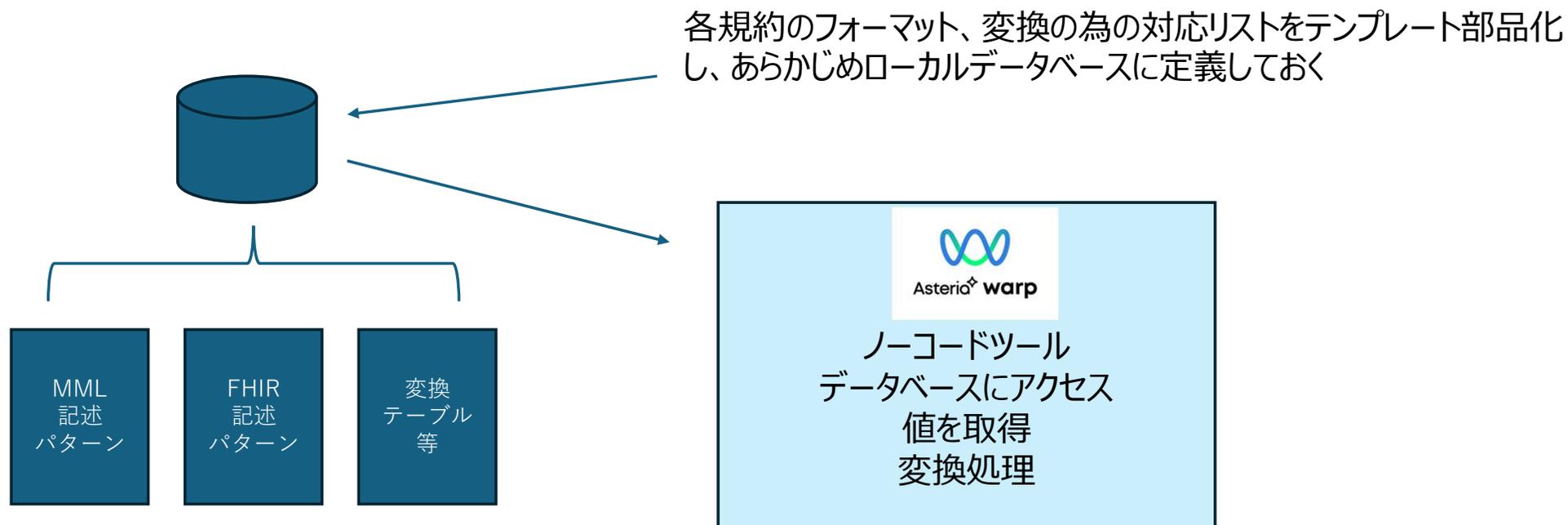
    public void execute(ExecuteContext context, Value[] in, Value out) throws MapperException {
        try {
            String commandStr = in[0].strValue();
            String resStr = CovertFunc(commandStr);
            out.setValue(resStr);
            :
            :
        }
    }
}
```

複数のフローが必要な処理をコンポーネント化する事で、簡素化される。



# テンプレート化

各規約のフォーマットをテンプレート部品化し、ローカルDBに保持することで、規約の変更、対応内容の変更が発生した場合の修正箇所を限定的にする



## (例)変換テーブル

ID	TableName	Comment	MMLValue	MMLDescrip	MMLTag	FHIRCode	FHIRText	FHIRTag
1	MML0010	性別	female	Female	<mmlPi:sex>female</mmlPi:sex>	female	female	"gender": "female"
2	MML0010	性別	male	Male	<mmlPi:sex>male</mmlPi:sex>	male	male	"gender": "male"
3	MML0010	性別	other	Other	<mmlPi:sex>other</mmlPi:sex>	other	other	"gender": "other"
4	MML0010	性別	unknown	Unknown	<mmlPi:sex>unknown</mmlPi:sex>	unknown	unknown	"gender": "unknown"
6	MML0015	Diagnosis cate	confirmedDiag	確定診断	<mmlRd:category mmlRd.tableId="	confirmed		"verificationStatus": { "coding": [
7	MML0015	Diagnosis cate	suspectedDiag	疑い病名	<mmlRd:category mmlRd.tableId="	unconfirmed		"verificationStatus": { "coding": [
8	MML0015	Diagnosis cate		該当なし		confirmed		"verificationStatus": { "coding": [
9	MML0015	Diagnosis cate					該当なし	
10	MML0016	Outcome (転)	died	死亡		O	その他	
11	MML0016	Outcome (転)	worsening	悪化		O	その他	
12	MML0016	Outcome (転)	unchanged	不変		O	その他	
13	MML0016	Outcome (転)	recovering	回復		O	その他	
14	MML0016	Outcome (転)	fullyRecovered	全治		M	寛解	
15	MML0016	Outcome (転)	sequelae	続発症(の発)		O	その他	
16	MML0016	Outcome (転)	end	終了		O	その他	
17	MML0016	Outcome (転)	pause	中止		I	中止	
18	MML0016	Outcome (転)	continued	継続		C	継続	
19	MML0016	Outcome (転)	transfer	転医		O	その他	
20	MML0016	Outcome (転)	transferAcute	転医(急性病)		O	その他	
21	MML0016	Outcome (転)	transferChron	転医(慢性病)		O	その他	
22	MML0016	Outcome (転)	home	自宅等へ退院		O	その他	
23	MML0016	Outcome (転)	unknown	不明		O	その他	
24	EX001	病名コードシ	ICD-10				http://hl7.org	

コンポーネンにて変換テーブルより値を読み込む

## (例)テンプレート

FROM TO	MODULE	KEYWORD	CONTENTS
MMLFHIR	Patient	0_FIRST	<pre> {   "resourceType": "Patient",   "id": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:docInfo/mml:docId/mml:uid&gt;%",   "meta": {     "lastUpdated": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:docInfo/mml:confirmDate&gt;%"   },   "language": "ja",   ##&lt; GetTable,MMLFHIR,Patient,identifier&gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,name &gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,telecom &gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,gender &gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,birthDate &gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,address &gt; ## } </pre>
MMLFHIR	Patient	identifier	<pre> "identifier": [   {     "system": "urn:oid:1.2.392.100495.20.3.51.1##&lt; 医療機関ID&gt; ##",     "value": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:content/mml:Pi:PatientModule/mml:Pi:uniqueInfo/mml:Pi:masteRid/mml:cm:Id&gt;%"   } ] </pre>
MMLFHIR	Patient	name	<pre> "name": [   {     "extension": [       {         "url": "http://hl7.org/fhir/StructureDefinition/iso21090-EN-representation",         "valueCode": "IDE"       }     ],     "use": "usual",     "text": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:content/mml:Pi:PatientModule/mml:Pi:personName/mml:Nm:Name[@mmlNm:repCode="I"]/mml:Nm:fullName&gt;%",     "family": "##&lt; sepText,%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:content/mml:Pi:PatientModule/mml:Pi:personName/mml:Nm:Name[@mmlNm:repCode="I"]/mml:Nm:fullName &gt;%, ,1 &gt; ##",     "given": [       ##&lt; sepText %&lt; </pre>

## テンプレートによる実際の変換処理イメージ (例は患者基本情報のMML→FHIR変換)

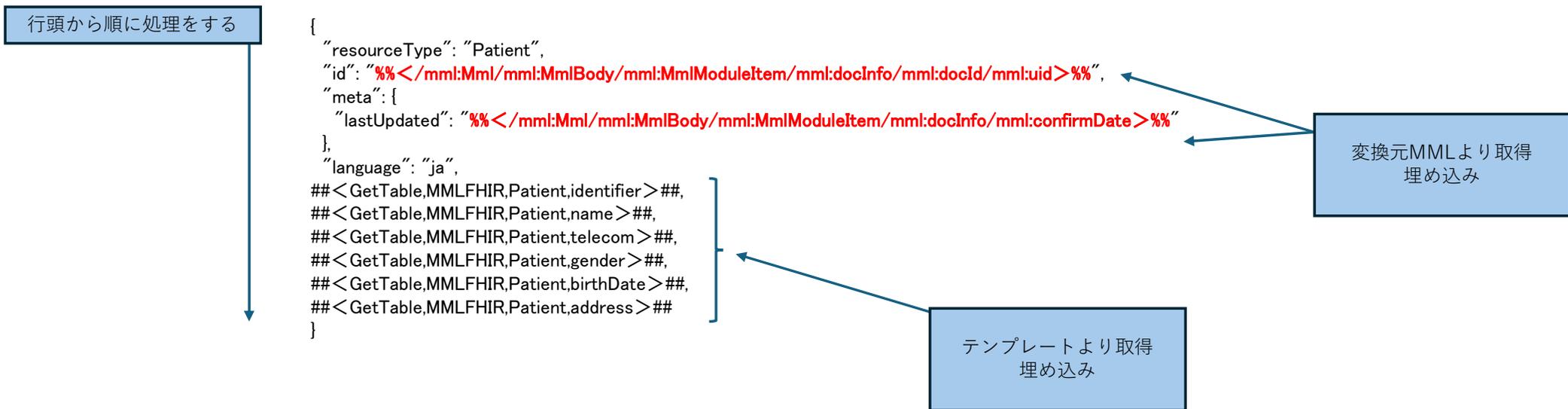
FROM TO	MODULE	KEYWORD	
MMLFHIR	Patient	O_FIRST	<pre>{   "resourceType": "Patient",   "id": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:docInfo/mml:docId/mml:uid&gt;%",   "meta": {     "lastUpdated": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:docInfo/mml:confirmDate&gt;%"   },   "language": "ja",   ##&lt; GetTable,MMLFHIR,Patient,identifier&gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,name &gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,telecom&gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,gender&gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,birthDate&gt; ##,   ##&lt; GetTable,MMLFHIR,Patient,address&gt; ## }</pre>



Patient用MML-FHIR変換用のテンプレートを読み込み変換処理を行う。

```
{
  "resourceType": "Patient",
  "id": "%</mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:docInfo/mml:docId/mml:uid>%",
  "meta": {
    "lastUpdated": "%</mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:docInfo/mml:confirmDate>%"
  },
  "language": "ja",
  ##< GetTable,MMLFHIR,Patient,identifier> ##,
  ##< GetTable,MMLFHIR,Patient,name > ##,
  ##< GetTable,MMLFHIR,Patient,telecom> ##,
  ##< GetTable,MMLFHIR,Patient,gender> ##,
  ##< GetTable,MMLFHIR,Patient,birthDate> ##,
  ##< GetTable,MMLFHIR,Patient,address> ##
}
```

# 入力ファイルの情報から、データベースより取得したコンテンツ情報をもとにしたテンプレートの構文を解析し変換する



## identifierのテンプレート

```
"identifier": [
  {
    "system": "urn:oid:1.2.392.100495.20.3.51.1##<医療機関ID>##",
    "value": "%<
/mml:Mml/mml:MmlBody/mml:MmlModuleItem/mml:content/mmlPi:PatientModule/mmlPi:uniqueInfo/mmlPi:masterId/mmlCm:Id>%"
  }
]
```

## テンプレートを処理する事でエクスチェンジ出力データを組み立てられていく

```
{
  "resourceType": "Patient",
  "id": "patient",
  "identifier": [{
    "system": "urn:oid:1.2.392.100495.20.3.51.11234567890",
    "value": "29234294"
  }],
  "active": true,
  "name": [{
    "extension": [{
      "url": "http://hl7.org/fhir/StructureDefinition/iso21090-EN-representation",
      "valueCode": "IDE"
    }],
    "use": "official",
    "text": "千年 患者",
    "family": "千年",
    "given": [ "患者" ]
  }],
  {
    "extension": [{
      "url": "http://hl7.org/fhir/StructureDefinition/iso21090-EN-representation",
      "valueCode": "SYL"
    }],
    "use": "official",
    "text": "せんねん かんじゃ",
    "family": "せんねん",
    "given": [ "かんじゃ" ]
  }],
  "gender": "male",
  "birthDate": "1968-02-10"
  "address": [
    {
      "text": "京都府京都市左京区下鴨下川原町",
      "postalCode": "6041133"
    }
  ]
}
```

identifierのテンプレート

## テンプレート化によるメンテナンス性の向上

各規約のフォーマットをテンプレート部品化し、ローカルDBに保持することで、規約の変更、対応内容の変更が発生した場合の修正箇所を限定的にする

FROM TO	MODULE	KEYWORD	
MMLFHIR	Patient	O_FIRST	<pre>{   "resourceType": "Patient",   "id": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlModuleItem v   "meta": {     "lastUpdated": "%&lt;/mml:Mml/mml:MmlBody/mml:MmlI   },   "language": "ja",   ##&lt; GetTable,MMLFHIR,Patient,identifier&gt;##,   ##&lt; GetTable,MMLFHIR,Patient,name &gt;##,   ##&lt; GetTable,MMLFHIR,Patient,telecom&gt;##,   ##&lt; GetTable,MMLFHIR,Patient,gender&gt;##,   ##&lt; GetTable,MMLFHIR,Patient,birthDate&gt;##,   ##&lt; GetTable,MMLFHIR,Patient,address&gt;## }</pre>

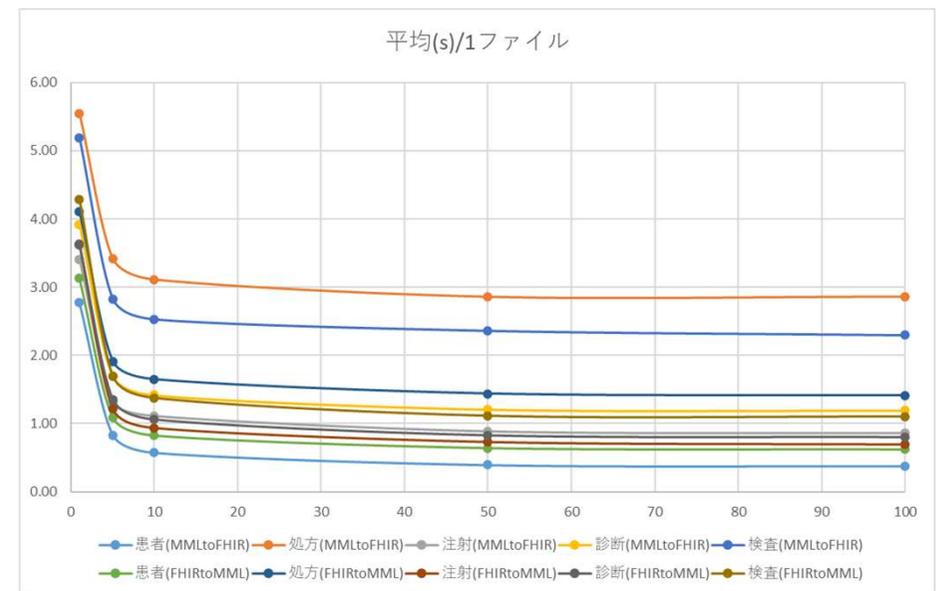
テンプレート化によりフローを修正することなく  
軽微な修正変更はテンプレートデータベースの  
内容を変更する事で行う事が可能

## 性能

処理性能について、複数ファイルの一括変換を実施した場合に1ファイルあたりの変換に要する平均処理時間は表の通りとなった。10件以上になってくると、1ファイルあたりの処理速度の変化が少なくなり、収束していく傾向が見られた。

約2.4secはエクステンジ処理以外の処理時間

変換パターン		平均処理時間(s) / 1ファイル				
データ形式	文書	1件処理	5件処理	10件処理	50件処理	100件処理
MMLtoHL7 FHIR	患者	2.77	0.83	0.58	0.39	0.38
	処方	5.55	3.42	3.11	2.86	2.86
	注射	3.41	1.32	1.11	0.88	0.86
	診断	3.92	1.69	1.42	1.21	1.19
	検査	5.19	2.83	2.53	2.36	2.29
HL7 FHIRtoMML	患者	3.13	1.09	0.83	0.64	0.62
	処方	4.11	1.91	1.65	1.44	1.41
	注射	3.63	1.21	0.93	0.73	0.69
	診断	3.62	1.35	1.06	0.83	0.80
	検査	4.29	1.69	1.37	1.11	1.10



## 変換率

MMLのモジュールとHL7 FHIRのリソースについて、どの程度割り当てることができたかについて表に示す

No	対象システム	文書種別 (種)	データ項目数 (件)					変換率 (%)	
			a.相互変換全項目数	b.変換先全項目数	c.補完用項目数	d.課題あり	e.対処後	課題除く (b-d)/b	対処後
1	MML to HL7 FHIR	6	97	80	17	25	0	68.8%	100%
2	HL7 FHIR to MML	6	97	61	36	11	0	82.0%	100%

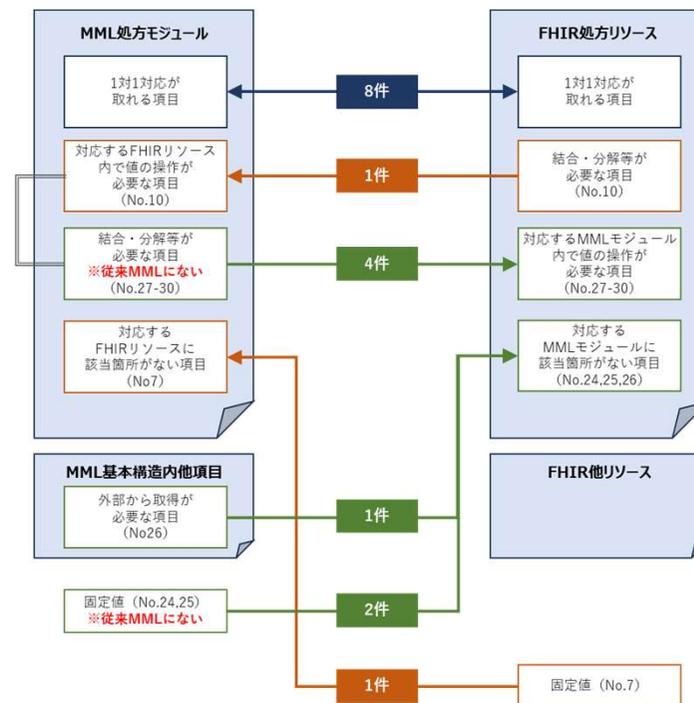
文書種別は「患者基本」、「病名」、「処方」、「注射」、「検査結果」の5つ+MMLの基本構造上、Header部およびBody部が存在するため、この「基本構造」部の再構成を含めた6文書として整理を行った。

データ項目数は、実際に各文書の中に含まれる「氏名」、「性別」、「病名」、「薬剤名称」等のデータ項目の数を示す。  
データ項目数については以下の分類で整理を行った。

- a. 相互変換全項目数 : 相互変換の変換対象とする項目数の総計 (= b + c)
- b. 変換先全項目数 : 変換先 (MMLtoHL7 FHIRの場合、HL7 FHIR) の項目数
- c. 補完用項目数 : 逆変換のために補完が必要な項目数
- d. 課題あり件数 : 変換対象項目がない等の課題があった項目数
- e. 対処後件数 : dに対して、変換方法の整理を行い解決した項目数

結果として、HL7 FHIRとMMLについて相互にマッピングを行うことができ、対処後は100%の変換率となっている。  
変換内容として、固定値付与などの暫定対処内容を含むが、HL7 FHIRとMML間のデータ形式の変換については実現可能であると判断できる。

ただし、課題の内容を見てみると、対象とする文書単位だけで変換が完結しないパターンが多く、対象とする文書以外から項目取得が必要となるケースや暫定の値を作成して付与するケースが多く見られた。これは、MMLとHL7 FHIR間に対応する文書同士の保有するデータの粒度が異なるために発生しており基本構造を除く5文書内で対応する文書同士で変換が完結するパターンは「患者基本」情報のみである。



相互変換については実現可能ではあるものの、出力されたデータを正しく理解して活用していくためには、データ粒度の違いによる不足データの補完をどこまで許容するかについて共通認識を形成しておく必要がある。

# 課題

課題	各種IDやコードの取得・変換方法を確立する必要がある (患者ID、ドキュメントID、医療機関コード 等)
対応方針	本事業でテーブル参照による動的な対象IDやコードの取得の仕組みは構築しており、実際の変換においてはこの仕組みを活用できる。 ただし、具体的なコード値等については、継続的にテーブルの内容をメンテナンスしながら、精度の高いテーブルを維持していく必要があるため、このようなテーブルを共通的にメンテナンス・管理する体制づくりを並行して推進する。

課題	相互変換を行う際に、不足項目の補完や複数インスタンスから必要なデータの取得を行う仕組みが必要である (作成者、診断医、診療科 等)
対応方針	PHRとの連携を前提として、必要なリソースや文書単位で指定して取得できるようにする。 項目不足時の暫定対応として、入力される固定値等についての共通的な認識を作ると共に、並行してPHRに対するデータ登録を推進し、必要なデータが最新の状態で格納されるようにデータの充足度合いを高める。

ご清聴ありがとうございました