

SeagaiaMeeting in KYOTO 2004

# MML V3 セミナー

## FAQを中心に

5月27日

NPO MedXMLコンソーシアム 技術委員長

中島裕生

# FAQ

- MMLでは多くのIDが出てきますが、説明してください
  - HL7 CDA Headerでのid(必須)
    - EX (Extension) --- 文書 ID (RT, AAN 内で Global に Unique な番号)
    - RT (Root) --- その機関または所属の OID (RT は OID でなければならない)
    - AAN (Assigning Authority Name) --- 文書を作成した機関名
  - **mmlCm:Id (Id 形式)**
  - 【内容】ID を表現する共通形式.
  - 【データ型】string

属性名	データ型	省略	使用テーブル	説明
mmlCm:type	string	#REQUIRED	optional	個人 ID に対しては MML0024 テーブルを, 施設 ID に対しては MML0027 テーブルを, 診療科 ID に対しては MML0029 を使用し, mmlCm:tableId に使用したテーブル名を記載する
mmlCm:tableId	string	#REQUIRED		上記 ID の種類コードを規定するテーブル名. 施設固有の個人 ID の場合に限り, 施設 ID を記載することができる

Id属性

EX (Extension) --- 文書ID(RT, AAN内でGlobalにUniqueな番号)

RT (Root) --- その機関または所属のOID (RTはOIDでなければならない)

AAN(Assigning Authority Name) --- 文書を作成した機関名

<clinical\_document\_header>

<id AAN="佐々木医院" EX = "12345" RT = "1.2.392.114319.1.5.4.09.999.0001"/>

<document\_type\_cd DN="MML Document" S="1.2.392.114319.1.1" V="0300"/>

<origination\_dttm V=""/>

<provider>

<provider.type\_cd V="PRF"/>

<person>

<id EX="01" RT="1.2.392.114319.1.5.4.09.999.0001" />

</person>

</provider>

S --- codeSystemのOID (MML3.0ではMMLのOID"1.2.392.114319.1")

DN (displayName) --- "MML Document"という文字列で固定

<id> ; 文書識別子

<document\_type\_cd> ; 文書タイプコード(下記表のコード値 デフォルトはPRF)

<origination\_dttm> ; オリジナル文書が作成された時刻

<provider> ; 医療サービスを提供している1人以上のヘルスケア提供者の指定

<patient>

コード	表示名	定義
ASS	assistant performer	実際に同席し関与することによってサービスを補助する人。これには、アシスタント、技師、アソシエートなど、あらゆる職種が含まれる。
CON	consultant	評価やコンサルティングで医療サービスに参加するアドバイザー。
PRF	performer	実際に主として行為 (action) を実行する人。第一責任のある従事者 (actor) である必要はない。たとえば、所属外科医の監督の下で動く外科研修医は performer である。

中島裕生

# MmlHeader でのID

- mmlCi:CreatorInfo

- 【内容】生成者識別情報. 構造は MML 共通形式 (作成者情報形式) 参照

- **masterId**

- 【内容】患者主 ID

- <mml:masterId>

- <mmlCm:Id mmlCm:tableId="1.2.392.114319.5.4.99.100.0001"  
mmlCm:type="facility">test001</mmlCm:Id>

- </mml:masterId>

- <mml:masterId>

- <mmlCm:Id mmlCm:tableId="JPN452101510001" mmlCm:type="facility">P-120</mmlCm:Id>

- </mml:masterId>

- **mmlCm:Id**

- 【内容】構造は MML 共通形式 (Id 形式) 参照

- **toc**

- 【内容】table of content . header 内の「toc」のチェックをシステムで行うことで HL7 CDA の body を見なくてもどのようなデータが入っているかを認識することができる

- **tocItem**

- 【内容】本 MML 文書で使われている全ての DTD をリストアップする

- **scopePeriod**

- 【内容】MML 本文全体の対象期間

- **encryptInfo**

- 【内容】電子署名などの暗号化情報. 記載方法を特に定めない. 基本方針の「アクセス権と暗号化」を参照

MML0024ID type (ID区分) → 個人ID

MML0027Facility ID type (施設ID区分) → 施設ID

MML0024テーブル

Value	Description
national	全国統一ID
local	地域ID
facility	施設内ID

MML0027テーブル

Value	Description
ca	認証局 (Certification Authority) 発行ID
insurance	保険医療機関コード
monbusho	文部省大学附属病院施設区分
JMARI	日本医師会総合政策研究機構コード



---

# 個人ID記載法

- 医療機関職員、患者とも地域IDをもちいる
- mmlCm:Id形式で、地域のIDの場合
  - `<mmlCm:Id mmlCm:type = "local" mmlCm:tableId = "MML0024">1303123456785</mmlCm:Id>`

## MML0027 Facility ID type (施設ID区分)→施設ID でのOIDの導入の仕方

- MMLのOIDは”**1.2.392.114319.1**”である.

OIDにおけるUIDs (Unique Identifiers) の構成は次のようになる.

- “1”--- Identifies ISO
- “2”--- Identifies ANSI Member Body
- “392”--- Country code of a specific Member Body (JAPAN for ANSI)
- “**114319**”--- MedXML CONSORTIUM (provided by ANSI)
- 1.2.392.114319.1 ←MML
  - 1.2.392.114319.1.1←MML文書
  - 1.2.392.114319.1.5←NPO MedXMLが提供する施設コード
  - 1.2.392.114319.5. \*
  - **1.2.392.114319.5.4**←東京都医師会





# 関連FAQ

- 施設IDがJMARIの場合

患者ID(master ID)の定義

type=facility 「施設内」

table ID=JPN13..... 「施設のJMARI」

ID=00000001 「PT.ID」電子カルテ内独自コード(データ)  
になりますが、

施設IDがcaの場合

①想定typeがfacility

type=facility 「施設内」

table ID=1.2.392..... 「施設のOID」

ID=00000001 「PT.ID」電子カルテ内独自コード(データ)

※この場合、facility型でOIDが書けるのでしょうか？

## ■ mmlCm:tableId

- mmlCm:typeでIDの種類コードを規定するテーブル名. 施設固有の個人IDの場合に限り, 施設IDを記載することができる.

- ```
<local_header descriptor="mmlheader" render="MML"
  <mml:MmlHeader>
    <mmlCi:CreatorInfo>
      <mmlPsi:PersonalizedInfo>
        <mmlCm:Id mmlCm:tableId="1.2.392.114319.5.4.99.100.0001"
          mmlCm:type="facility">sirai</mmlCm:Id>
```

MedXML発行のOID  
を利用

データタイプはstringだから  
アルファベットのIDでも  
可

# FAQ

- MMLにはいくつものモジュールがあるが、どう使えばいいかわからない
  - MML0007テーブルにある文書種別から、電子カルテから出力されるべき文書種別を特定する
  - 特定された文書を表現するに必要な情報を電子カルテから収集し、そこに使われている書式用語と同じ意味を持つXML上の要素をMML規格書から探す
  - その際、必要な要素が複数のモジュールに渡る場合は大いにある

# MML0007 Generation purpose (文書詳細種別)

24種類

- record 診療記録
- recordAdmission 入院時診療記録
- recordInpatient 入院中診療経過記録
- recordConsult 紹介時診療記録 (状態, 処方, 検査値等)
- recordDischarge 退院時診療記録 (状態, 処方, 検査値等)
- recordOutpatient 外来診療記録
- legalRecord 法的義務に基づく診療記録
- consult 紹介状
- report 返書報告書
- reportRadiology 放射線報告書
- reportPathology 病理報告書
- reportTest 検歴報告書
- summary 要約
- summaryAdmission 要約 (入院時)
- summaryPreoperation 要約 (術前)
- summaryPostoperation 要約 (術後)
- summaryMid 要約 (中間)
- summaryDischarge 要約 (退院時)
- summaryOutpatient 要約 (外来)
- disclosure 患者診療情報開示用
- informedConsent 患者説明文書
- study 研究用記録
- other その他
- claim 医事情報

```
<mml:title  
generationPurpose="consult">紹介状  
</mml:title>
```

# FAQ

- 検歴モジュール送信時の文書ヘッダーの作成者は検査実施側と記述されていますが、外部検査会社を利用し、診療所内のシステムでは特別に検査会社情報を管理していないため、依頼側を設定しようと考えていますが、不都合が発生しますでしょうか？
  - BML、SRLなどの所謂、検査会社のセンターが検査実施側として記述されます。院内検査であれば病院が検査実施側です。検査実施施設は#REQUIREDなので、省略はダメでしょう。不明であれば、依頼側コードにブランチをつけるなどでどうでしょう。
  - **MmlHeader**
  - **mmlCi:CreatorInfo**  
【内容】生成者識別情報。構造は MML 共通形式 (作成者情報形式) 参照。  
【省略】不可
  - **検歴情報モジュールの目的と対象**  
検体検査の結果情報を記載するための MML モジュールである。一モジュールは、通常の運用では、一依頼伝票 (もしくは一検査セット) にあたる。対象検査としては、検体検査であり、内分泌負荷試験, 尿沈渣を含む。細菌培養検査, 病理組織検査, 細胞診は対象外とする。
  - **namespaces 宣言**  
xmlns:mmlLb="http://www.medxml.net/MML/ContentModule/test/1.0"  
  
mmlCi:CreatorInfo  
検歴情報モジュールにおける文書作成者とは、検査依頼者ではなく、検査を実施した施設の代表者である。よって、モジュール内の検査実施施設や検査実施者と一致することが想定される。

# FAQ

- 検歴の依頼IDは用途として何に使われているのでしょうか？  
電子カルテ開発者が任意に設定しても問題ないIDですか？
  - 検査オーダーのIDですね。一つ質問がありますが、御社電子カルテでは、検査オーダーを発行するのでしょうか？ もし発行しているならば、内部的にオーダーIDを持っているのではないのでしょうか。その場合、オーダーIDを利用することもできます。あるいは、外部検査会社側ID。

| 属性名            | データ型   | 省略        | 説明                                             |
|----------------|--------|-----------|------------------------------------------------|
| mmlLb:registId | string | #REQUIRED | 依頼ID<br>同じ検査依頼から出た検査中報告<br>と最終報告は、同じ依頼 ID とする。 |

# FAQ

## ■ 患者さんに診療情報を全て見せてしまうのでしょうか？

```
<body>
<section>
<paragraph>
<content>
<local_markup descriptor="" render="MML">
<mml:docInfo contentModuleType="progressCourse">
<mml:securityLevel>
<mml:accessRight permit="all">
<mmlSc:facility>
<mmlSc:facilityName mmlSc:facilityCode="creator">記載者施設</mmlSc:facilityName>
</mmlSc:facility>
</mml:accessRight>
<mml:accessRight permit="read">
<mmlSc:facility>
<mmlSc:facilityName mmlSc:facilityCode="experience">診療歴のある施設</mmlSc:facilityName>
</mmlSc:facility>
</mml:accessRight>
<mml:accessRight permit="read">
<mmlSc:person>
<mmlSc:personName mmlSc:personCode="patient">被記載者(患者)</mmlSc:personName>
</mmlSc:person>
</mml:accessRight>
</mml:securityLevel>
<mml:title generationPurpose="record">S:相変わらず咳が止まらない</mml:title>
```

**MML0034アクセス許可区分**  
**none** すべてのアクセスを不許可  
**read** 参照を許可

複数の accessRight エlementが存在する場合は、accessRight エlement同士は "OR" 条件となる。1 つの accessRight エlement直下に、施設、診療科、職種、個人の内 2 つ以上が存在するときは、"AND" 条件となる。この AND 条件を満たすものが実在しない場合には、アクセス権は認められない

# FAQ

- MMLV3におけるアクセス権限は現在ドキュメント単位ということですが、エレメント単位でのアクセス権の割り当ても可能なのでしょうか？  
具体的に言うと、検歴モジュールで感染症の項目がある患者は見せないといった感じです
- アクセス権限設定はdocInfo単位です。したがって、現時点では無理ということになります。見せたくないものと見せてよいものがあれば、ドキュメントを分けてもらう必要があります。



# FAQ

- 現在当院では電子カルテデータを日々落とし込んでいるのですが、リアルタイム性を重視した為、各モジュール単位でドキュメントを作成しています。MMLv3的には単体モジュールでも問題はありませんか？
- 経過記録モジュールを使用する際、電子カルテデータの構造上プレーンテキストとして格納されていますので、”FreeExpression“を使用したのですがMMLv3的に問題はありませんか？
  - FreeExpressionは全然問題ありません。プレーンテキストで記述すると、あと検索がテキストサーチとなり大変ですが、貴院では全文検索エンジンを使われるので問題にならないかもしれません。モジュール単位でドキュメントが構成されていると、患者単位で情報を見る時、やはり検索の問題がでてきます。通常は、groupIdによる文書の関連付けをお勧めしますが、別途、工夫をされているのなら問題ありません。
  - **mmlPc:ProgressCourseModule**
    - 【内容】経過記録情報. mmlPc:FreeExpression か mmlPc:structuredExpression のどちらかのみ選択

# FAQ

- `<?xml version="1.0" encoding="UTF-16" ?>`  
`<mml:levelone xmlns:mml="http://www.medxml.net/MML"`  
`xmlns:xhtml="http://www.w3.org/1999/xhtml"`  
`xmlns:mmlCm="http://www.medxml.net/MML/SharedComponent/Comm`  
`on/1.0"`  
`xmlns:mmlNm="http://www.medxml.net/MML/SharedComponent/Name/`  
`1.0"`

上記の間違いは？

- `<?xml version="1.0" encoding="UTF-16"?>`  
`<mml:levelone ...`  
→  
`<?xml version="1.0" encoding="UTF-16"?>`

`<!DOCTYPE levelone SYSTEM "MML_3.0_levelone_1.0.dtd">`

`<levelone ...`

---

# FAQ

- MMLで使用できる文字コードは？
  - `<?xml version="1.0" encoding="UTF-16" ?>`の encodingには、Shift\_JIS, UTF-8, UTF-16が利用できます。
- 外字は使用できますか？
  - レセコンなどで個別メーカーが提供しているような外字は使えません

# FAQ

- 1. シェーマ(イメージ画)を乗せるには経過記録モジュールのどこに乗せれば良いのでしょうか？
- 2. シェーマ(イメージ画)はバイナリデータとして乗せることは出来るのでしょうか？
- 経過記録情報モジュール(mmlPc:ProgressCourseModule)だけを使って退院時サマリーを書くのではなく、その他のモジュールも使って記述します。その意味で、mmlCm:extRefで添付する画像(シェーマ等)を利用します

- **mmlCm:extRef (外部参照形式) サンプルインスタンス**

```
<mml:extRefs>  
  <mmlCm:extRef  
    mmlCm:contentType = "IMAGE/ JPEG "  
    mmlCm:medicalRole = "surgicalFigure"  
    mmlCm:title = "手術記録図"  
    mmlCm:href = "patient1234/image10.jpg"/>  
</mml:extRefs>
```

**MML0033Medical  
Role**

MML インスタンスからの相対的 uri 記述. 外部参照ファイルは, MML インスタンスに添付して送られることを想定しているため, 絶対的 uri は使用しない

# FAQ

- 医療情報自体の記述はMML3.0に従い、通信規約はMML2.3に準拠する事は可能か？
  - HOT Proj.(Dolphin)では、MML V3データ交換制御(原案仕様)を実装しました
  - HOT Proj.(Dolphin)へのデータ入力、保存、データ読み込みも、MML V3インスタンスで行うことが可能です
    - 通信規約はMML2.3から次版への拡張が討議されています
  - この話題は、電子カルテからMMLインスタンスを送出し、マイカルテで情報を参照する場合には気にする必要はありませんが、電子カルテ側からMML Queryを送って、MMLインスタンスを電子カルテ側に持ってくることを想定される場合に重要になります

# MML Queryライブラリ

- JavaからMMLQueryをなげられるプログラムを作成しました
  - HOTプロジェクトに繋げる診療所電子カルテ用途
    - IDの扱いを変更すれば、他のドルフィンでも利用可能
  - ドキュメントはJavaDocで提供し、ソースコードを公開します
    - 東京都医師会のHPで公開予定
  - MMLListQuery.java ... MMLの文書リストを要求するMML インスタンスを生成
  - MMLDocQuery ... MMLの文書本体を要求するMMLインスタンスを生成
  - SDAA.java ... SDAAへのアクセスを提供
    - 動作環境について  
JDK1.3.1、1.4以上で動作。  
JUG(<http://www.doomdark.org/doomdark/proj/jug/>)を使用してUUIDを生成しているため1.3.1にはUUID制限あります。

---

# クラス MMLDocQuery

- public class **MMLDocQuery**
- extends java.lang.Object
- MML文書のリストを要求するMMLインスタンスを生成します。  
内部的には、以下の順番でStringを結合することで文書が完成します。

getXMLHeader()

getMMLQuery(startDate)

getMMLHeader(userid, userName, facilityId, facilityName, patientId )

getMMLBody( docId, userId, userName, facilityId, facilityName )

なお、リクエストIDについては

JUG(<http://www.doomdark.org/doomdark/proj/jug/>)  
でUUIDを生成して使用しています。

---

# クラス MMLListQuery

- public class **MMLListQuery**
- extends java.lang.Object
- MML文書のリストを要求するMMLインスタンスを生成します。  
内部的な話ですが以下の順番でStringを結合することで文書が完成します。

```
getXMLHeader()  
getMMLQuery(startDate)  
getMMLHeader(userid, userName, facilityId, facilityName, patientId )  
MMLBODY
```

なお、リクエストIDについては

JUG(<http://www.doomdark.org/doomdark/proj/jug/>)  
でUUIDを生成して使用しています。



---

# クラス SDAA

- public class **SDAA**
- extends java.lang.Object
- SDAAへのアクセス、およびMMLインスタンスの送信、結果の受信を行うクラス MMLQueryを発行するためには事前のログインが必要になるためこのクラスではSocketクラス、ストリームをメンバ変数にすることでセッションを保っています。

```
public static void main(String args[]) {
    SDAA sdaa = new SDAA("terminal.tokyo.med.or.jp", 5101);
    try {
        sdaa.login("13040000726", "test");

        //文書のリストを要求
        String query =
            MMLListQuery.getListMMLQuery("2001-01-01",
                "13040000726",
                "医者1",
                "99.100.0001",
                "佐々木医院",
                "13040000725");

        System.out.println(sdaa.sendMMLQuery(query));

        //文書の本体を要求
        String docquery =
            MMLDocQuery.getDocMMLQuery("10019",
                "13040000726",
                "医者1",
                "99.100.0001",
                "佐々木医院",
                "13040000725");

        System.out.println(sdaa.sendMMLQuery(docquery));
    }
    catch (Exception e) {}
    sdaa.close(); //クローズ処理(必須)
}
```



使用例 メインPGM