

MMLのこれから

京都大学EHR共同研究講座

特定講師 小林慎治

Agenda

- MMLの存在意義および役割とは
- MMLの問題点
- MML4.x
- MML5にむけて

MMLとの関わり

- 2003年
 - 未踏ソフトウェア創造事業
- 2006年末から2007年はじめくらい
 - MMLのメーリングリストで暴れる
- 2007年2月
 - 町屋ミーティング
 - openEHR.jp
- 2013年
 - 介護モジュールの設計
- 2015年
 - 処方・注射モジュールの設計
- 2016年
 - MML4.xの発表

MMLの存在意義とは何か

- HL7, SS-MIX2の他に医療情報標準が必要なのか
 - “Good competitor”
 - SS-MIX1/2が対応しない文書種別が存在している
- あえて国内標準を作るよりも、国際標準を国内でも利用できるようにした方がいいのではないか
 - 国内だけでしか流通し得ない情報もある
 - 保険情報
 - 日本の医療の流儀と海外の流儀とは異なるものもある。
 - 人名表記、住所表記
 - 国際標準化のプロセスを経ての国内標準化には多大なコストが掛かる

MML/CLAIMの現在の役割

- メッセージとして
 - 施設間連携
 - 電子カルテとレセコン
- 保存形式として
 - バックアップデータ
- データモデルとして
 - 電子カルテ内部のデータモデル、

メッセージと保存文書

メッセージの特性

- 可搬性重視
 - コンパクト
- 一時的情報
 - 汎用性重視
 - 定型的

保存文書の特性

- 可用性重視
 - 検索性
- 永続性重視
 - 変更・改ざん対策
 - 版管理、監査証跡
- *XML database

MMLの現状と問題点

- XMLに由来する問題点
 - 処理コストが大きい
- 設計上の問題点
 - 汎用性のために、拡張性が犠牲となっている
 - モジュールの一部を取り出して他のモジュールで利用することができない
 - モジュール間の統一性を一部欠いている
 - 検歴モジュールでは独自の部門情報などを定義
 - モジュールの区分けが曖昧
 - 初診時特有情報モジュール、基礎的診療情報モジュール、生活習慣情報モジュール
- API設計
 - MML Query
- ライセンスの問題
 - 実装報告義務

MMLがこの先生きのこるために

- 他のフォーマットにも対応する
 - JSON, Protobuf, Apache Avro, Messagepakなど
- 情報モデル設計を整備する
 - モジュールの再設計
 - 分割、再合成
- ライセンス変更
 - HL7 license -> CC-0(Creative Commons 0, Public domain)
 - openEHR license -> Apache 2, CC-SA

MML 4.xの開発方針

- MML 1-3のまとめ
 - 多フォーマット対応の前に、これまでの定義について整理し、明確化する
 - 構造の変更は最小限度にとどめる
- DTDからXSDへの変更
 - 型定義の見直し
- 新規モジュール追加
 - バイタルサイン、体温表、処方、注射、透析
- ツール類の整備
 - 形式チェックの自動化
 - OSSとして公開

MML 4.0

- DTDからXSDへ変更
 - Oxygen XML EditorでDTDより変換
- 文字コードの変更
 - Shift-JISよりUTF-8に変更
- 公開文書の整備
 - AsciiDocを採用。
 - GitHubでの公開、課題管理、変更履歴
- MML Queryの廃止

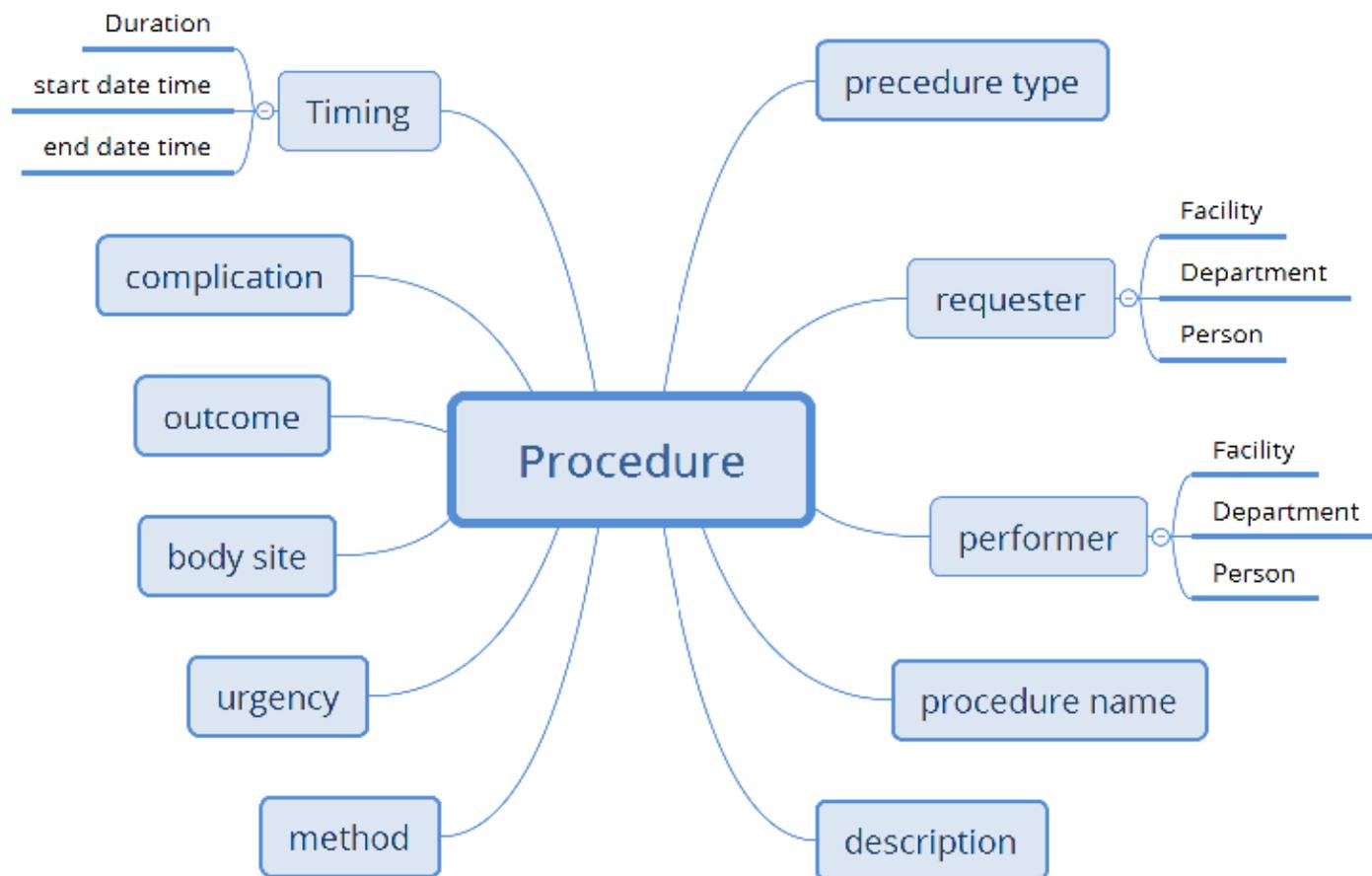
MML 4.1.2

- 累積バグの修正
 - 文書、スキーマの不整合など
- 透析モジュールの取り込み
 - MML最大モジュール
- 千年カルテ対応
- mmlHeaderの必須項目tocをオプションに変更

MML 4.2.0にむけて

- 累積バグの修正
- 文書生成の自動化
 - Schemaからのテーブル、解説のスケルトン生成
 - 文書とスキーマ間の不整合防止
- 処置モジュールの追加

処置モジュール



CLAIMについて

- 現行CLAIMについて
 - レセプト - 電子カルテ間の汎用API
 - 予約・請求モジュール
- 保険請求を扱える数少ない医療標準規格
 - レセ電フォーマット、EFファイルなど
- 存続の方針
 - Minor update + REST対応

MML 5.0の概要

- XML, JSON両対応
- モジュール構成の見直し
 - 既存のモジュールを分割する
 - 例) 生活習慣情報モジュールを、喫煙歴、飲酒歴、職業歴に分割し、サマリーモジュールでも使用できるようにする
- 1インスタンス1文書構成とする
 - 複数文書をまとめる構成は廃止する
- ドキュメントの更新履歴対応
- セキュリティモジュールへの利用形態の追加
 - 二次利用についての許諾条項
- ターミノロジー対応を強化する
 - 意味論的相互運用性のため

HL7 FHIR

- HL7, V3, CDAの失敗
 - 仕様が肥大化しすぎて実装コストがついてこなかった
 - よくあるXML規格の失敗の一つ
- 医療独自標準というよりも既存の標準に医療情報を載せていく方針
 - Atom形式(RSS形式の流れをくむ) から強く影響
 - XML, JSON, RDFのフォーマットに対応
 - 要求の80%をタグ付けし、残りはextensionで個別に対応
 - 概念モデルを細かく分割し、共通のResource(資源)として表現する
- 現在Release 3, DSTU3(Draft Standard for Trial Use 3)

FHIRの構造

- メタデータ、人間可読部、拡張定義(URL)
- 標準データ

```
<Patient xmlns="http://hl7.org/fhir">
  <id value="glossy"/>
  <meta>
    <lastUpdated value="2014-11-13T11:41:00+11:00"/>
  </meta>
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>Henry Levin the 7th</p>
      <p>MRN: 123456. Male, 24-Sept 1932</p>
    </div>
  </text>
  <extension url="http://example.org/StructureDefinition/trials">
    <valueCode value="renal"/>
  </extension>
  <identifier>
    <use value="usual"/>
    <type>
      <coding>
        <system value="http://hl7.org/fhir/v2/0203"/>
        <code value="MR"/>
      </coding>
    </type>
    <system value="http://www.goodhealth.org/identifiers/mrn"/>
    <value value="123456"/>
  </identifier>
  <active value="true"/>
  <name>
    <family value="Levin"/>
    <given value="Henry"/>
    <suffix value="The 7th"/>
  </name>
  <gender value="male"/>
  <birthDate value="1932-09-24"/>
  <careProvider>
    <reference value="Organization/2"/>
    <display value="Good Health Clinic"/>
  </careProvider>
</Patient>
```

Resource
Identity &
Metadata

Human
Readable
Summary

Extension
with URL to
definition

Standard
Data:
• MRN
• Name
• Gender
• Birth Date
• Provider

FHIRメッセージの例(Detected Issue)

XML

```
<DetectedIssue xmlns="http://hl7.org/fhir">
  <!-- from Resource: id, meta, implicitRules, and language -->
  <!-- from DomainResource: text, contained, extension, and modifierExtension -->
  <patient><!-- 0..1 Reference(Patient) Associated patient --></patient>
  <category><!-- 0..1 CodeableConcept Issue Category, e.g. drug-drug, duplicate therapy, etc. --></category>
  <severity value="[code]"/><!-- 0..1 high | moderate | low -->
  <implicated><!-- 0..* Reference(Any) Problem resource --></implicated>
  <detail value="[string]"/><!-- 0..1 Description and context -->
  <date value="[dateTime]"/><!-- 0..1 When identified -->
  <author><!-- 0..1 Reference(Practitioner|Device) The provider or device that identified the issue --></author>
  <identifier><!-- 0..1 Identifier Unique id for the detected issue --></identifier>
  <reference value="[uri]"/><!-- 0..1 Authority for issue -->
  <mitigation> <!-- 0..* Step taken to address -->
  <action><!-- 1..1 CodeableConcept What mitigation? --></action>
  <date value="[dateTime]"/><!-- 0..1 Date committed -->
  <author><!-- 0..1 Reference(Practitioner) Who is committing? --></author>
</mitigation>
</DetectedIssue>
```

JSON

```
{
  "resourceType": "DetectedIssue",
  // from Resource: id, meta, implicitRules, and language
  // from DomainResource: text, contained, extension, and modifierExtension
  "patient": { Reference(Patient) }, // Associated patient
  "category": { CodeableConcept }, // Issue Category, e.g. drug-drug, duplicate therapy, etc.
  "severity": "<code>", // high | moderate | low
  "implicated": [{ Reference(Any) }], // Problem resource
  "detail": "<string>", // Description and context
  "date": "<dateTime>", // When identified
  "author": { Reference(Practitioner|Device) }, // The provider or device that identified the issue
  "identifier": { Identifier }, // Unique id for the detected issue
  "reference": "<uri>", // Authority for issue
  "mitigation": [{ // Step taken to address
    "action": { CodeableConcept }, // R! What mitigation?
    "date": "<dateTime>", // Date committed
    "author": { Reference(Practitioner) } // Who is committing?
  }]
}
```

HL7 FHIRをMMLのベースとする 場合の問題点

- 情報モデルの構造
 - リソースが増えて、若干の混乱が見られる
 - リソースごとにタグをハードコードしているため、修正や変更にかかるコストがかかる。
 - スキーマの維持管理についてのコストが大きくなる
- MML情報モジュールとの互換性
 - FHIR リソースとMMLモジュールでの合成する場合にMMLのnamespaceとの衝突が発生する場合がある
 - スキーマチェックが煩雑となる

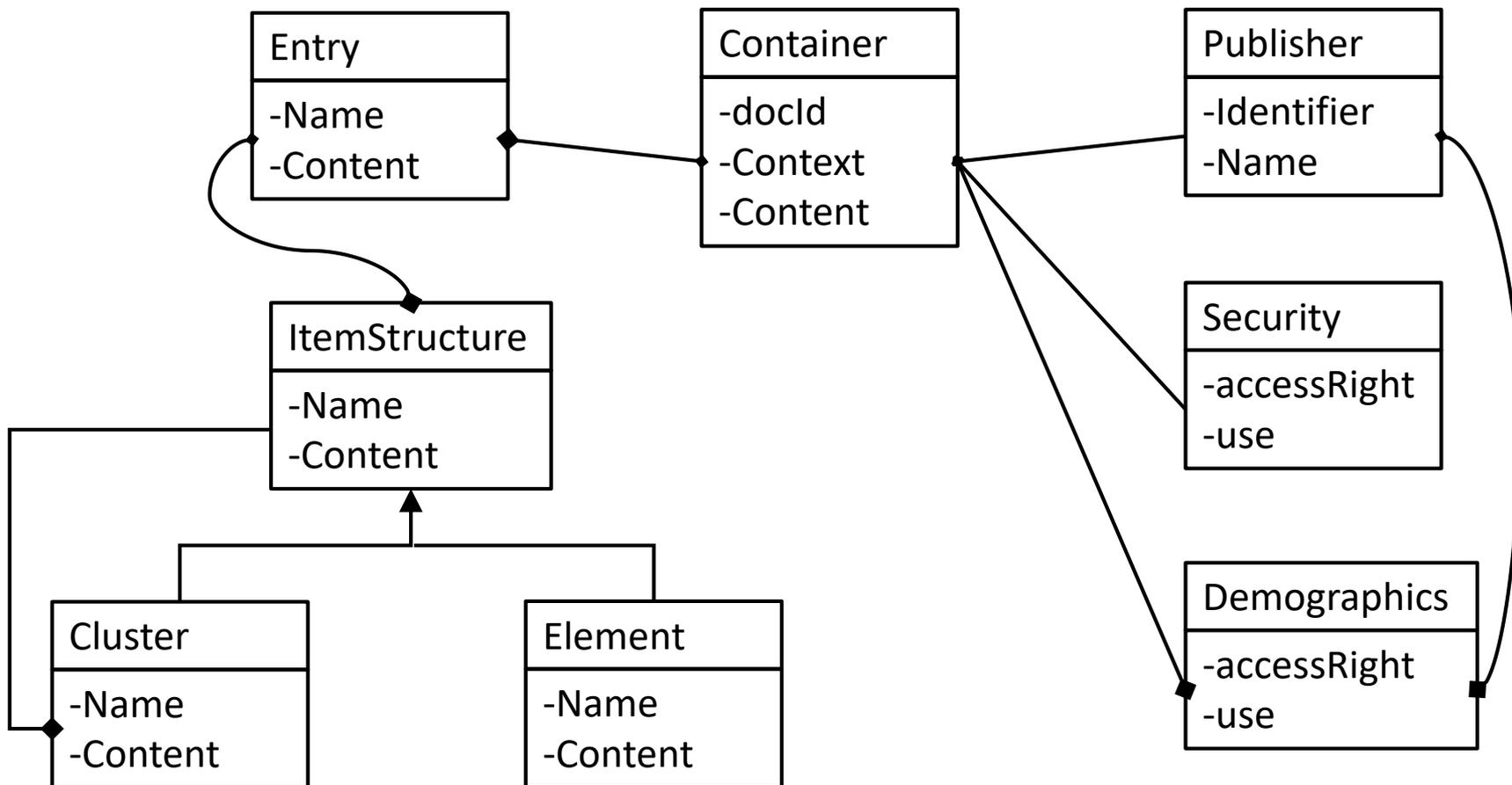
現在のMML5.0開発状況

- openEHR/ISO 13606のように基本構造と情報モデルを分離する
- JSON Schema, XML schema同時開発中
 - Date型など、XML Schemaにはあって、JSONにはない型の取り扱い
- Modelの変更
- 4系列と同時開発
- ネームスペースの廃止

MML 5.0 基本設計

- 1インスタンス1文書
 - HeaderとdocInfoの統合
- ISO 13606のサブセット
 - Container – Entry – Cluster – Element
- 情報使用範囲の設定
 - Use
 - ANY(どのような目的でも使用可能) , POC(臨床用途のみ) , RESEARCH(研究用途限定)

MML 5.0 クラス図



例:LifestyleModule 5.0

XML

```
<container>
  <docId>1234-5233-3333</docId>
  <type>MML</type>
  <created>2017-05-18</created>
  <name>Lifestyle</name>
  <entry>
    <name>Smoking</name>
    <element>
      <name>Description</name>
      <dataType>Text</dataType>
      <value>Cigarette 20x30years</value>
    </element>
  </entry>
</container>
```

JSON

```
{
  "resourceType": "container",
  "docId": "1234-5233-3333",
  "type": "MML",
  "created": "2017-05-18"
  "name": "Lifestyle",
  "entry": {
    "name": "Smoking",
    "element": {
      "name": "Description",
      "dataType": "Text",
      "value": "Cigarette 20x30years",
    }
  }
}
```

Entry(例：身長)

```
<entry>
  <name>Body
height</name>
  <element>
    <name>Height</name>

<datatype>Quantity</datatyp
e>
  <value>178</value>
  <unit>cm</unit>
</element>
</entry>
```

```
{
  "entry": {
    "name": "Body height",
    "element": {
      "name": "height",
      "datatype": "quantity",
      "value": 178
      "unit": "cm"
    }
  }
}
```

Entry(例：血压)

```
<entry>
  <name>Blood pressure</name>
  <cluster>
    <element>
      <name>Systolic</name>
      <datatype>Quantity
    </datatype>
      <value>120</value>
      <unit>mmHg</unit>
    </element>
    <element>
      <name>Diastolic</name>
      <datatype>Quantity
    </datatype>
      <value>80</value>
      <unit>mmHg</unit>
    </element> </cluster>
</entry>
```

```
{
  "entry": {
    "name": "Blood pressure",
    "cluster": {
      "element": {
        "name": "Systolic",
        "datatype": "quantity",
        "value": 120,
        "unit": "mmHg"
      }
    }
  }
}
```

リリーススケジュール

MML 4.2.0

- 2017年6月
 - アルファリリース
- 2017年10月
 - ベータリリース
- 2018年1月
 - 正式リリース

MML 5.0.0

- 2017年11月
 - アルファリリース
- 2018年5月
 - ベータリリース