

「DMD仕様 v3.0」ワーキングドラフトに対するご意見募集への回答

独立行政法人情報処理推進機構

2017年2月15日

| No | ご意見  | 回答   |
|----|--|--|
| 1  | <p>「3.1 DMD ヘッダー」では、XML 版と RDF 版の二種類のヘッダーが必須であると規定していますが、これは過剰ではないでしょうか？ 一種類でいいと考えます。仕様が複雑になって読者にとっては理解の妨げになりますし、各種ツールの開発者は常に双方の整合や相互変換を意識して工数を割く必要があります。デメリットのほうが大きいと感じています。なお、RDF 版は Endpoint に投入して検索に利用するなどの明確な用途が想像できるので、二択であれば RDF 版を残すのがよいと考えます。</p> | <p>DMD3.0 では、JSON-LD を必須とし、XML を任意とする対応を検討させていただきます。</p>   |
| 2  | <p>「2.概要」では、DMD のメタデータとして「最新版」が定義されていますが、不要ではないでしょうか？ データベースなどで機械的に管理しないかぎり、最新版が更新されるたびに過去の DMD を更新しつづけるのは困難だと考えます。</p>  | <p>DMD メタデータの「最新版」は任意であることを明記する対応を検討させていただきます。また運用上メンテナンスは困難ではありますが、XML スキーマの互換性を維持するためすぐには廃止せず現状のままとさせていただきます。</p>  |
| 3  | <p>「2.概要」では、DMD のメタデータとして「最終更新日」「バージョン」が必須とされていますが、これらは任意（または不要）ではないでしょうか？ もしも DMD が更新を認めず、都度新バージョンを作るような運用の場合、これらのメタデータは使用されないと考えます。</p>  | <p>DMD メタデータの「最終更新日」及び「バージョン」について任意設定に変更する対応を検討させていただきます。</p>  |
| 4  | <p>「3.1.XML 版」では、Component 要素の属性として、media-type が規定されています。この属性に関して仕様書内で言及がないので正確な用途は不明ですが、ファイルの Content-Type を指定するものとした場合、このような属性は不要ではないでしょうか？ DMD は zip ファイルですので、Content-Type によって中身を指定するのではなく、展開した際に利用者がファイル形式を容易に推測できるよう、適切な拡張子の付与をルールとするのがよいと考えます。</p>  | <p>「media-type」は Content-type を指定する意図のものですが説明が不足しておりました。その旨を追記する対応を検討させていただきます。また任意設定であることも明記するよう対応を検討させていただきます。</p> |

| No | ご意見   | 回答  |
|----|---|---|
| 5  | <p>「3.1.2 RDF 版」では、DMD ヘッダーの RDF 版は Turtle ではなければならぬと規定しています。これをデータ構造を固定した JSON-LD として提供できないでしょうか？ 多くの開発者にとって XML や Turtle よりも JSON 形式のほうが受け入れやすいと考えます。たとえば次のような JSON 形式であれば、RDF としての情報量を損なわず、非 RDF な開発者にも扱いやすいはずです。</p> <pre> {   "@context": "http://imi.go.jp/ns/dmd/3.0/context.json",   "id": "http://example.org/imins/00068574/",   "class": "Asset",   "type": "DataModelDescription",   "issued": "2015-09-01",   "modified": "2015-10-22",   "description": "犯罪発生情報を表現するための DMD",   "publisher": "http://example.org/user/0123456789",   "title": "犯罪発生情報 DMD",   "license": "http://creativecommons.org/licenses/by/3.0/",   "prev": "http://example.org/imins/00068573/",   "charCollection": 258,   "distribution": [     {       "class": "AssetDistribution",       "id": "header.json",       "type": "Header"     },     {       "class": "AssetDistribution",       "id": "crime.xsd",       "type": "XMLSchema"     },     {       "class": "AssetDistribution",       "id": "redefine.xsd",       "type": "XMLRedefineSchema"     },     {       "class": "AssetDistribution",       "id": "crime-shacl.ttl",       "type": "RDFDataShape"     }   ] } </pre> <p>(次ページへつづく)</p> | <p>RDF 版 DMD ヘッダーについては Turtle を廃止し、JSON-LD に変更する対応を検討させていただきます。</p> |

| No | ご意見  | 回答  |
|----|--|---|
| 5  | <p>(前ページからつづく)</p> <pre> }, {   "class" : "AssetDistribution",   "id": "item-data.json",   "type": "ItemData" }, {   "class" : "AssetDistribution",   "id": "document.pdf",   "type": "Documentation" }, {   "class" : "AssetDistribution",   "id": "document.docx",   "type": "Documentation" }, {   "class" : "AssetDistribution",   "id": "sample.jsonld",   "type": "Sample" }, {   "class" : "AssetDistribution",   "id": "mapping-info.json",   "type": "Mapping" }] } </pre> <p>※ context.jsonld を活用し、各種プロパティを整備したり @type や @id のエイリアスを作る想定</p> |   |
| 6  | <p>「3.4 項目データ」では、「項目データは、JSON 形式のファイルであり、全体は”itemList” メンバーのみをもつオブジェクトとして表現される。」と規定されています。 項目データにアプリケーションの固有データをいれたいと考えているのですが、現在の仕様では収録すると違反となってしまいます。仕様で定義されていない項目を認めないのではなく、入れた場合にエラーにならないような規定に変更できないでしょうか？ 「本仕様で規定されていないプロパティは無視される」など。 加えて、「_(アンダースコア)」で始まるプロパティは自由に使用してよい、など開発を意識したルールがあるとなおよいです。</p>   | <p>ご意見ありがとうございます。</p> <p>項目データ以外を”ItemList” に含めることも許容していることから、規定の変更なしに対応可能かと存じます。</p> <p>ルート要素直下にプロパティを追加することは仕様上不可としておりますが、itemList 配下であれば、現状の仕様でも valid となるため、規定変更なしで表現可能となります。</p> |

| No | ご意見  | 回答   |
|----|--|--|
| 7  | <p>「3.4 項目データ」の itemList 配下のオブジェクトは項目の名称と文字コレクションを持つことができると規定されています。ここで定義される項目オブジェクトと DMD のデータ構造を対応付ける手法として、構造化項目名記法を用いることを検討しています。この目的のために、項目オブジェクトに対して path プロパティを保持できるようにし、これを標準としてほしいです。以下は実際に構造化項目名を「項目データ」の一部として収録したものです。{</p> <pre> "itemList": [{   "name": "ID",   "path": "*場所" }, {   "name": "名前",   "path": "場所&gt;名称&gt;表記",   "charCollectionList": [256, 258] }, {   "name": "緯度",   "path": "場所&gt;地理座標&gt;緯度" }, {   "name": "経度",   "path": "場所&gt;地理座標&gt;経度" }, {   "name": "所在地",   "path": "場所&gt;住所&gt;表記",   "charCollectionList": [256, 258] }] </pre> | <p>ご意見を基に検討しました結果、構造化項目名を用いるには、「path」というキー名は不適切であると認識いたしました。「structuredLabel」というキー名を、標準として採用することを検討いたします。また、「structuredLabel」の採用と合わせてキー名「name」を「label」に改める対応を検討いたします。</p>                    |
| 8  | <p>「3.7 項目マッピング」では、「項目データの各項目と当該 DMD で定義されたデータモデルとの対応関係」を表現するために CSV on the Web が導入されています。ただ、CSVW は開発者にとっても読み解くのが困難で扱いづらいと感じています。一方、構造化項目名記法は「階層構造をもつデータの位置」を平易に指定することを目的としており、項目とデータ構造の対応付けに応用できるのではないのでしょうか？ 構造化項目名記法を使用した平易なマッピング表現を提案します。具体的には、「3.4 項目データ」で定義される各項目オブジェクトに対して、以下のように直接構造化項目名記法による位置を付与するというものです。</p> <p>(次ページへつづく)</p>   | <p>頂きましたご意見を基に検討させていただき、CSVW で表現していたマッピングを構造化項目名で置き換えること検討させていただきます。</p> <p>マッピングを構造化項目名で置き換えることにより、下位互換は失われてしまいますが、CSVW と構造化項目名の相互変換ツールを提供させて頂くことで DMD3.0 への移行をサポートできるよう対応を検討させていただきます。</p> |

| No | ご意見  | 回答 |
|----|--|----|
| 8  | <p>(前ページからつづく)</p> <p>場所情報の対応付け例</p> <pre>{   "itemList": [{     "name": "ID",     "path": "*場所"   }, {     "name": "名前",     "path": "場所&gt;名称&gt;表記"   }, {     "name": "緯度",     "path": "場所&gt;地理座標&gt;緯度"   }, {     "name": "経度",     "path": "場所&gt;地理座標&gt;経度"   }, {     "name": "所在地",     "path": "場所&gt;住所&gt;表記"   } ]</pre> <p>簡単な入籍情報の例</p> <pre>{   "itemList": [{     "name": "日付",     "path": "状況&gt;発生日&gt;表記"   }, {     "name": "夫",     "path": "状況&gt;関与[夫]&gt;氏名&gt;姓名"   }, {     "name": "妻",     "path": "状況&gt;関与[妻]&gt;氏名&gt;姓名"   } ]</pre> <p>(次ページへつづく)</p> |    |

| No | ご意見  | 回答   |
|----|--|--|
| 8  | <p>(前ページからつづく)</p> <p>ツールプロジェクトの内部では、この方式はシンプルでわかりやすい／データ構造を直感的に把握できる／手書きも可能ではないか、などの感想を得ており、おそらくは一般の開発者のみなさんからも支持を集めるのではないかと予想しております。また、ツールプロジェクトではこのような JSON オブジェクトから CSV on the Web の JSON を出力するツールの試作を行い、問題なく生成が可能であることを確認しています。仮にこの方式を採用していただけるのであれば、CSV on the Web はオプション扱い、または、廃止 (CSVW を必要とする利用者が適宜ツールで生成すればよい、という立場) とすることもご検討いただければと思います。</p>  |  |
| 9  | <p>「3.2」および「3.3」では DMD のデータ構造を定義するための SHACL/XSD に関して規定されています。ツールプロジェクトでは構造化項目名記法を使用した平易なデータ構造表現をもとに、SHACL や XSD を生成する方式を開発しています。同方式の公式採用を検討いただけないでしょうか？</p> <p>構造化項目名記法(+制約拡張)によるデータ構造定義</p> <p>「例 5: RDF データモデルの記述例」で表現されるデータモデルは以下のような形式で表現できます。</p> <p>場所&gt;名称{1..1}&gt;表記{1..1}</p> <p>場所&gt;名称{0..1}&gt;カナ表記{1..1}</p> <p>場所&gt;地理座標{0..1}&gt;緯度{1..1}</p> <p>場所&gt;地理座標{0..1}&gt;経度{1..1}</p> <p>場所&gt;住所{1..1}&gt;表記{1..1}</p> <p>・通常の構造化項目名に対して (カーリーブラケット／波括弧) による制約条件を拡張</p> <p>・ {0..1} {0..n} {1..1} {1..n} と書くことでカーディナリティを表現</p> <p>(次ページへつづく)</p> | <p>ご提示頂きましたデータ構造表現を基に検討をさせていただいた結果、制約条件の記述については、IMI の構文を表現する方法として採用を検討させていただきます。</p> <p>また採用検討のなかでは、XML と RDF 共通の仕様を記述できる方法としても検討をさせていただきます。</p> |

| No | ご意見   | 回答 |
|----|---|----|
| 9  | <p>(前ページからつづく)</p> <p>また、以下の例ではさらに「カナ表記」に対して正規表現制約を付与しています</p> <p>場所&gt;名称{1..1}&gt;表記{1..1}</p> <p>場所&gt;名称{0..1}&gt;カナ表記{1..1}{/[ァ-ヶ]+/}</p> <p>場所&gt;地理座標{0..1}&gt;緯度{1..1}</p> <p>場所&gt;地理座標{0..1}&gt;経度{1..1}</p> <p>場所&gt;住所{0..1}&gt;表記{1..1}</p> <ul style="list-style-type: none"> <li>・ {} は連結して書くことが可能</li> <li>・ {/regex/} の書式で、正規表現制約を付与</li> </ul> <p>リテラル値に対しては文字列(正規表現、最大文字数、最小文字数)、数値(&lt;, &gt;, &lt;=, &gt;=, ==, !=) といった制約を導入することを想定しています。また、以下のように型制約を表現できます。</p> <p>関与&gt;関与者{@人}&gt;氏名&gt;姓名</p> <p>重量&gt;数値{@xsd:nonNegativeInteger}</p> <p>上の例では関与者の値域を、デフォルトの実体型から人型に限定しています。下の例では、リテラルのデータ型をデフォルトの xsd:decimal から xsd:nonNegativeInteger に限定しています。</p> <p>ツールプロジェクトでは、上記のような記法を用いて作成したデータモデル定義から、SHACL や XSD を生成するツールを開発しており、実現性を確認しています。この記法はわかりやすいのでぜひ普及させたいと考えていますが、現状は 1 ツールの内部表現に過ぎません。公式採用をご検討いただければと思います。仮にこの方式を採用していただけるのであれば、XSD や SHACL はオプション扱いとすることもご検討いただければと思います。</p> |    |