

# 再起的記述を可能とする映像音声メディア・オントロジー

加藤 慎<sup>1</sup> 曾根 卓朗<sup>2,3</sup> 塚田 学<sup>1</sup> 江崎 浩<sup>1</sup>

**概要：**著者が所属する SDM (Software Defined Media) コンソーシアムでは、2016 年より三次元映像音声メディアを管理するためのプラットフォームとして SDM Ontology の提案を進めてきた。SDM Ontology は、メディアデータとともに様々な種類・粒度、かつ、膨大な量のメタデータを階層構造に整理して管理可能とする。従来の提案は、収録段階についての記述に焦点を当てた設計となっており、収録されたメディアデータの編集段階まで考慮していなかったため、編集作業に用いられるソフトウェアやミキサーなどのメディアプロセッサの情報を記述するための構造がなかった。また、メディア収録の情報は静的である一方で、メディアの編集作業は収録されたデータや編集されたデータをもとに繰り返しおこなわれることも多く、一般に動的といえるため従来設計では対応しきれない。このような問題を踏まえ、本稿では SDM Ontology の構造について整理し直し、修正を加えるとともに、動的構造を表現するために再帰的記述を導入し、メディアの収録・編集について記述可能な SDM Ontology Version 2.0 を提案する。

## Recursive description capable ontology for audio-visual media

SHIN KATO<sup>1</sup> TAKURO SONE<sup>2,3</sup> MANABU TSUKADA<sup>1</sup> HIROSHI ESAKI<sup>1</sup>

### 1. はじめに

インターネット技術は日々発展し続け、社会基盤として広く深く根付いている。スマートフォン・タブレット機器の普及に始まり、あらゆるものがインターネットに接続される IoT 機器も世の中に浸透し、益々日常生活とインターネットが密接に関係していくなか、著者が所属する SDM (Software Defined Media) コンソーシアム [1] [2]<sup>\*1</sup>では、2014 年よりインターネットを前提とした映像・音響メディアに関する研究を進めてきた。SDM コンソーシアムの主な研究目的は、メディアをオブジェクト単位で管理しソフトウェア制御することで新たなメディアの表現方法を開拓していくことである。これまでメディアの収録から、収録環境のメタデータを含むデータ群の管理、それらのデータを用いた仮想空間上の、あるいは、仮想空間と現実空間を組み合わせたメディア表現・演出方法の探求など、様々な研究活動をおこなってきた。

メディア管理の視点から、2016 年より SDM コンソーシアムでは三次元映像・音響メディアを管理するためのプラットフォームとして SDM Ontology [3] [4] の提案を進めてきた。収録の際にはメディアのデータファイルの他に、

収録環境や内容について様々かつ膨大なメタデータが発生する。SDM Ontology ではこれらのメタデータを階層構造に整理し、RDF (Resource Description Framework)<sup>\*2</sup>による記述、LOD (Linked Open Data) の作成によってオープンデータとしてアクセス可能となる。従来の提案では、カメラやマイクの収録機器とそれらによって生成される映像音声メディアファイルは 1 対 1 対応で用意されることが前提であったが、収録されたいくつかのメディアファイルを加工・編集することで新たなメディアファイルを生成する際に、生成されたメディアファイルに対応すべき収録機器が存在し得ないという問題が浮かび上がった。さらには、編集元となったメディアファイルや、作業において用いられたソフトウェアあるいはミキサーといったメディアプロセッサの情報を包含することが難しいという問題もある。そこで、本稿では SDM Ontology の構造について整理し直し、修正を加えるとともに、メディアオブジェクトや収録機器に再帰的な記述を可能とするよう拡張した SDM Ontology Version 2.0 を提案する。

本稿の構成は以下である。まず、2 節で RDF, LOD に関する基本事項を説明し、RDF 語彙の定義に関する研究として関連オントロジーを紹介する。3 節で本研究の目的を述べ、4 節で本研究の目的を実現するために修正・拡張した SDM Ontology について述べる。5 節では SDM Ontology に基づいて管理されたメディアデータとアプリケーションの連携例を紹介する。最後に、6 節で本稿のまとめと今後

<sup>1</sup> 東京大学  
The University of Tokyo

<sup>2</sup> 株式会社アプリックス  
Aplix Corporation

<sup>3</sup> 静岡大学  
Shizuoka University

<sup>\*1</sup> <https://sdm.wide.ad.jp/>

<sup>\*2</sup> <https://www.w3.org/RDF/>

の課題を述べる。

## 2. 関連技術・研究

関連技術である RDF や LOD についての基本事項と SDM Ontology に関連する既存オントロジーについて述べる。

### 2.1 関連技術

RDF (Resource Description Framework) とは Web 上にあるリソースを記述するためのモデルであり、1999 年 2 月に W3C (World Wide Web Consortium) によって規格化された。RDF では、トリプル (Triple) と呼ばれる主語 (Subject)、述語 (Predicate)、目的語 (Object) の 3 つの要素でリソースの関係を表現する。トリプルの集合を RDF のグラフと呼び、リソースの識別には URI (Uniform Resource Identifier) を用いる。リソースの種類やリソース同士の関係を表す語彙を RDF Vocabulary といい、基本的な RDF Vocabulary の拡張を RDF Schema と呼ぶ。RDF のデータを格納する専用データベースをトリプルストア (Triplestore) と呼び、RDF のクエリ言語 SPARQL<sup>\*3</sup>を用いてクエリを発行することができる。RDF を記述する方法として XML 形式を採用した RDF/XML や JSON 形式を採用した JSON-LD、簡潔で比較的可読性が高い Turtle、Turtle やその前身の Notation3 の派生であり簡潔性を排除する代わりに解析性を向上させた N-Triples などがある。SPARQL クエリには Turtle 記法を用いることができる。

Tim Berners-Lee は Web 上にデータを公開する方針として以下の 4 条件をみたす Linked Data を提唱した [5] [6]。

1. モノやコトに対する識別子に URI を使用する
  2. 誰もが参照できるよう HTTP の URI を使用する
  3. URI を参照した際には、RDF や SPARQL などを用いて有用な情報が提供される
  4. 多くの情報が得られるよう外部へのリンクを含める
- LOD (Linked Open Data) とはオープンライセンスのもとで公開され、無料で再利用が可能な Linked Data のことである。LOD および Linked Data は RDF を用いて作成することが可能であり、特に RDF データを、外部リンクを含みオープンライセンスのもとで公開することは LOD の作成を意味する。

RDF データや LOD は、Tim Berners-Lee が提案するオープンデータのための 5 段階評価の枠組み “5-star Open Data” [7] において、4 段階目、5 段階目と高い評価に位置づけられる。再利用性の高さや処理に特定のアプリケーションを必要としないこと、URI を用いてリソースを一意に識別できることが利点に挙げられ、さらに、LOD では外部と参照し合うことでデータの価値が高まる。

### 2.2 関連オントロジー

SDM Ontology に関連するオントロジーとして、音楽関連の RDF 語彙を提供する既存のオントロジーについて紹介する。

Music Ontology [8] [9] は、音楽関連の情報を記述するための統一されたフレームワークである。Music Ontology は、Timeline Ontology [10] や Event Ontology [11]、国際図書館連盟 (IFLA) による書誌レコードの機能要件 (FRBR) のオントロジー [12]、FOAF (Friend of a Friend) [13] などの既存オントロジーに基づいて構成されており、アーティスト名や楽曲タイトルの記述や、作曲からパフォーマンス、収録、リリースまで各段階を表現することが可能である。

音楽制作フローのうち、技術的範囲については Music Ontology では扱わず、Studio Ontology [14] が表現する。Studio Ontology では、マイクの配置や機器間の配線の接続方法、ミキシング、信号処理などの一連のオーディオ収録・編集フローを記述可能とする。特にオーディオエフェクトについては、より詳細な記述が Audio Effect Ontology [15] [16] として拡張定義されている。

楽器の分類に関するオントロジーを構築するための研究として、異なる 2 つの楽器分類手法に基づいたプロトタイプオントロジーを実装実験をおこなう研究 [17] や楽器の音色の特徴によって楽器の識別およびグルーピングをおこない、階層的な楽器オントロジーの自動生成を試みる研究 [18] が挙げられる。

Music Ontology と連携するオントロジーはいくつか存在する。Audio Features Ontology [19] [20] はオーディオの特徴抽出において、特徴抽出の計算ワークフローの記述や、様々な特徴データの形式に対応する表現構造を提供する。また、抽象度の高いオントロジーとして Audio Commons Ontology [21] [22] がある。様々なオンラインサービスが利用している音響や楽曲、合成サウンドなどは Web 上のリポジトリに格納されている。Audio Commons Ontology はオーディオコンテンツのリポジトリ統合やクライアントアプリケーションからリポジトリへのアクセスを容易にするために設計された。最後に、音楽に役立つデータを扱うコンピューティングデバイスのネットワークを IoT (Internet of Things) になぞらえて IoMusT (Internet of Musical Things) と呼び、IoMusT のエコシステムを表現するために Internet of Musical Things Ontology [23] [24] が提案された。

## 3. 本研究の目的

SDM コンソーシアムが提案する SDM Ontology は、メタデータとともに様々な種類・粒度、かつ、膨大な量のメタデータを階層構造に整理し管理することを可能とする。一般的なメディア制作の流れは図 1 に示すように収録、編集、配信、再生の段階に分かれる。従来の提案は、収

<sup>\*3</sup> [https://www.w3.org/2009/sparql/wiki/Main\\_Page](https://www.w3.org/2009/sparql/wiki/Main_Page)

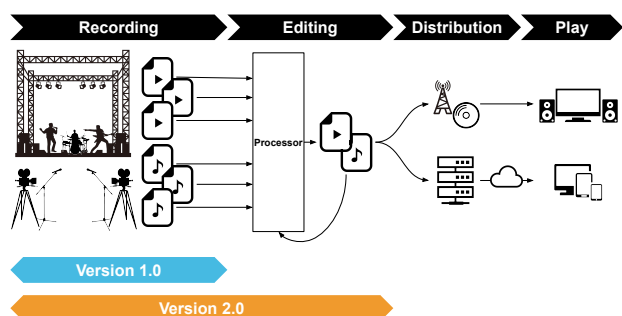


図 1: メディア制作フローの概要と SDM Ontology が記述可能な範囲。

録時の環境や内容についての記述を可能とする設計となっていたが、収録されたメディアデータの編集段階の記述までを考慮できていなかった。メディア収録の情報は静的であり、一度メタデータを階層構造化してしまえばアーカイブとして扱うことができる。一方で、メディアの編集作業は収録されたデータや編集されたデータをもとに繰り返しおこなわれることも多く、一般に動的といえるため従来設計では対応しきれなかった。また、カメラやマイクの収録機器とメディアファイルは1対1関係で対応付けられることが前提となっていたが、編集段階を考えると必ずしもメディアファイルは収録機器だけが生成するものではなく、編集用のソフトウェアやミキサーなどのメディアプロセッサから生成されることも多い。さらに、編集作業に用いられるメディアプロセッサの情報を記述するための構造が設計になかった。このような問題から、SDM Ontology の設計を修正・拡張していくことを本研究の目的とする。主な修正・拡張要件を以下に記す。

#### 再帰的記述：

収録機器の再帰的記述は、例えば、最小単位である単一のマイクからそれらを束ねたマイクアレイの表現を可能とし、このようなグループ化を階層的に記述することによって収録機器環境そのものの記述を可能とする。また、メディアの編集作業は多数の編集工程を経て行われることが一般的である。素材となるデータから「1次加工物」を生成し、これを素材として「2次加工物」を生成する。この過程を繰り返して生じた「 $n$ 次加工物」が完成版パッケージ（最終的な完成形）として世の中に出回っていくことが多い。SDM Ontology 上で再帰的記述を可能とすることで、メディアオブジェクトの再帰的記述は加工の度に生成される「 $n$ 次加工物」についての編集履歴を作成する。

#### 設計の抽象化：

不足していたメディア編集の構造を設計と並行して従来設計の見直しをおこなう。従来の設計では、音楽イベントに踏み込んだクラスや語彙も多く用意していたが、それぞれの分野・イベントに特化した記述は、既存

の外部のオントロジーと連携することで表現可能であると考え、SDM Ontology が主に扱う範囲をメディアの収録、編集の構造化に絞り、収録内容のプログラムや演奏者など、より詳細な情報の記述には適切な外部オントロジーを活用することとする。

## 4. SDM Ontology

3節で述べた本研究の目的や修正・拡張要件をもとに修正・拡張をおこない、SDM Ontology Version 2.0 を提案する。

### 4.1 SDM Ontology Version 2.0 の構造概要

SDM Ontology Version 2.0 の構造概要を図2に示す。図中の各ノードはクラスであり、赤字のクラスが基本クラス、青字のクラスがその他のクラスを表す。ノード内の述語は目的語にリテラルをとる。ノード間をつなぐ矢印の傍らに書かれた述語は矢印の根本にあるクラスがもつ述語であり、矢印が指すクラスのオブジェクトのURIを目的語にとる。SDM Ontology のベースURIは <http://sdm.hongo.wide.ad.jp/sdmo/> とし、クラスや述語は原則ベースURIからの相対URIで示している。ただし、外部オントロジーを利用している場合には「接頭辞 + 相対URI」の形で示してあり、例えば“schema:name”は、接頭辞“schema:”が <http://schema.org/> [25] を表し、相対URIは“name”であるので、<http://schema.org/name> を示す。

基本クラスを設計するにあたり次の考察をおこなった。現実空間内での出来事は収録・編集作業によってデジタルデータに記録され、仮想空間内で取り扱うことが可能となる。逆に、デジタルデータを用いて生成した仮想空間は再生システムによって現実空間内で再生することが可能となる。この考察から、収録環境や対象など現実空間内の事象を記述する Context クラス、現実空間と仮想空間の媒介を担う収録や編集、再生のシステムを記述する Recorder, Processor, Player クラス、仮想空間にて取り扱うことが可能なメディアデータを記述する Media クラスの5つを基本クラスとする。この他、収録に用いる時空間座標系を定義する CoordinateSystem クラスや、ここで定義された座標系での時空間座標を記述する Geometry クラス、メディア編集のためにメディアオブジェクトと時刻情報のペアを保持する MediaEvent クラスを補助クラスとして定義する。本研究では基本クラスおよび MediaEvent クラスの構造決定をおこない、外部オントロジーとの連携が必要なクラス、特に Context クラスや CoordinateSystem クラス、Geometry クラスの構造については現在議論段階である。

### 4.2 Context クラス

Context クラスは収録するイベント情報やプログラム、

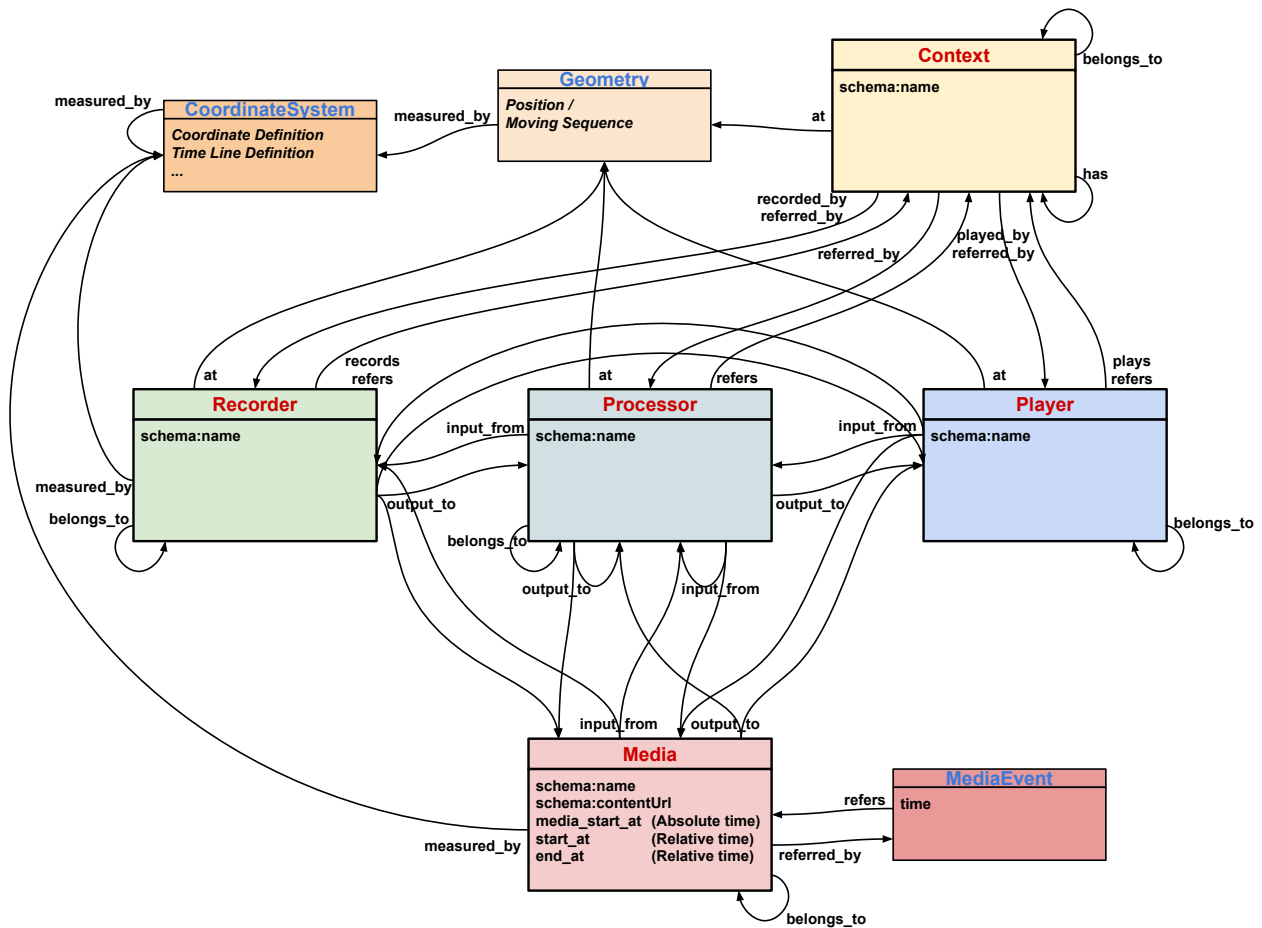


図 2: SDM Ontology Version 2.0 の構造概要.

人物や楽器などの対象、環境などを記述するためのサブクラスを用意する。図 3 の各ノードはクラスであり、オレンジの矢印はスーパークラスとサブクラスの包含関係を示す。また、黄色枠内が SDM Ontology の Context クラスの範囲で、外側には連携を検討している外部オントロジーのクラスが書かれている。外部オントロジーのクラスや連携するための述語関係は検討段階であり確定していない。

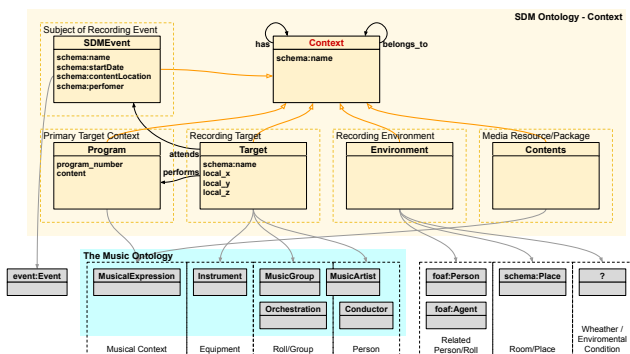


図 3: SDM Ontology Version 2.0 Context クラスの概要.

従来設計においては SDMEvent クラスや Target クラスが、Context クラスと並ぶ基本クラスとして定義されてい

たが、大局的に見るとこれらのクラスはいずれも現実空間内での事象を記述しており、前項 4.1 で述べた通りの SDM Ontology Version 2.0 基本クラス設計理念に基づいて、Context クラス 1 つをスーパークラスに置き、SDMEvent クラスや Target クラスをサブクラスとして位置づける。また、音楽イベントに特化したクラスなど具体的に詳細なクラスがいくつか用意されていたが、SDM Ontology Version 2.0 では取り扱う表現範囲の抽象度を高めて構造をシンプルにする、という方針があり、分野ごとの詳細クラスについてはそれぞれの分野専門の外部オントロジーと適切に連携することで表現したい。

#### 4.3 Recorder, Processor, Player クラス

Recorder, Processor, Player クラスはそれぞれサブクラスをもつ。サブクラスとスーパークラスの関係図を図 4 に示す。代表して Recorder クラスについて述べるとサブクラスには、より用途が詳細な DataRecorder, AudioRecorder, VideoRecorder クラスとそれらの複合を表現する CompositeRecorder クラスを定義する。例えば、ビデオカメラやスマートフォンなどの動画撮影では、一般に映像収録とステレオ (2ch) 音声収録が同時におこなわれている。従来設

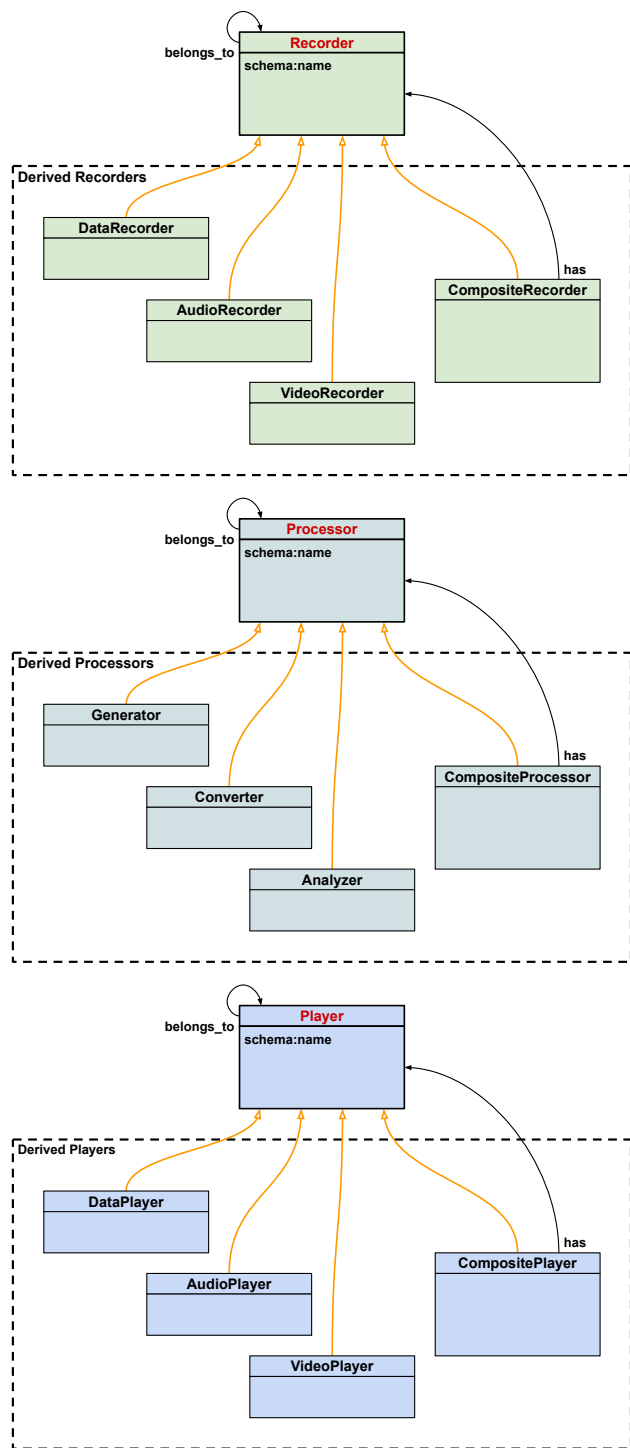


図 4: SDM Ontology Version 2.0 Recorder クラス, Processor クラス, Player クラスの概要.

計においてもビデオカメラやスマートフォンを Recorder クラスとして定義することは可能であったが, SDM Ontology Version 2.0 ではビデオカメラやスマートフォンがもつ映像収録機構を VideoRecorder クラス, ステレオ音声収録機構を 2ch まとめて, あるいは, 1ch ずつ AudioRecorder クラスとして定義し, ビデオカメラやスマートフォン自体を, 再帰的記述によって詳細な機構の複合体である CompositeRecorder クラスと定義することが可能となる. Processor,

Player クラスでも同様に, 再帰的記述によって編集・再生システムの最小単位から複合系までユースケースにあわせた自由な粒度で記述することが可能となる.

#### 4.4 Media クラス

Media クラスについても Recorder, Processor, Player クラスと同様にサブクラスをもつ. ただし, メディア編集の記述のために図 5 に示すように, Recorder, Processor, Player クラスの構造とは少し異なり MediaEvent クラスとの連携が発生する.

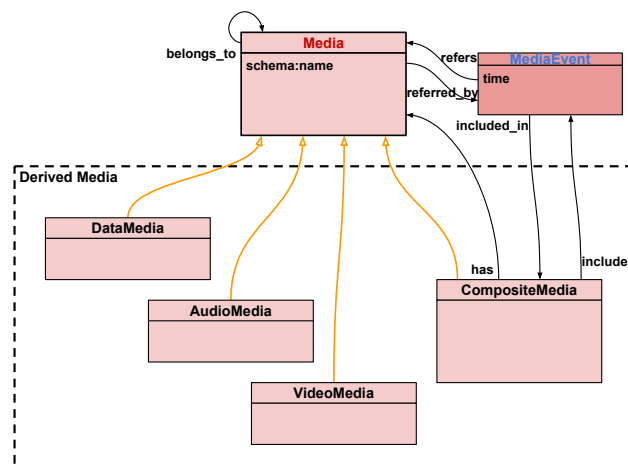


図 5: SDM Ontology Version 2.0 Media クラスの概要.

CompositeMedia クラスでは, 単なるメディアデータのグルーピングを表現するのではなく, 複数のメディアデータを時間軸上に切り貼りして編集された新たなメディアを表現することが可能である. 例えば, CD や DVD, Blu-ray ディスク上の音声・映像メディアデータを考える. メディアデータは複数のトラックあるいはチャプターに分かれて時間軸上に並んでいる. トラックやチャプターを単位に AudioMedia や VideoMedia など定義した後, CompositeMedia クラスの時間軸上にこれらを並べていくことで CD や DVD, Blu-ray ディスク全体のメディアデータを記述することが可能となる. ここで, CompositeMedia クラスは「貼り付けるメディアデータ」と「貼り付ける時刻」のペアを保持する必要があるため, メディアオブジェクトと時刻情報のペアを保持する MediaEvent クラスを定義し利用することとした.

CompositeMedia クラスには, MediaEvent クラスと連携するための includes/included.in のプロパティとは別に, MediaEvent クラスと連携せずに Media クラスの複合体を表現するための has プロパティが用意されている. 主に時間依存のないメディアについての複合体表現のために用意されたプロパティではあるが, CompositeMedia で定義するメディアが映像・音声などの時間に依存するメディ



アでありながら、包含関係の記述に has が使用されている場合には、時間軸始点 (0 秒の位置) に包含するすべてのメディアデータを貼り付けることと同等の記述であるとみなす。これにより、例えば、複数チャンネルの音声をチャンネルごとの音声の複合体として定義したいとき、この記述には複雑な時間軸上の情報を含める必要はなく、チャンネルごとの音声を AudioMedia クラスと定義し、Composite クラスには has プロパティで直接まとめることが可能となる。

## 5. SDM Ontology の活用

SDM Ontology Version 2.0 はメディアの収録および編集環境について階層的に記述する。記述されたデータはオープンデータとして公開されることが前提となっており、アプリケーションにこのデータを組み込んで活用できるようにしたい。本研究では、インタラクティブな 3D 視聴体験を提供する Web アプリケーション「Web360<sup>2</sup>」[26]\*<sup>4</sup> との連携を試作しローカル環境内で動作させた。

### 5.1 SDM Ontology のレポジトリ作成と簡易テスト

SDM Ontology Version 2.0 に基づいて、以下の 2 つの音楽イベントの収録・編集環境を RDF のデータに記述した。記述の形式には RDF グラフを簡潔に表現することが可能な Turtle<sup>\*5</sup>を採用した。

1. 2016 年 1 月 10 日に慶應義塾大学日吉キャンパス内に建つ藤原洋記念ホールで開催された慶應義塾大学コレギウム・ムジクム古楽アカデミーのコンサート
2. 2017 年 1 月 26 日に六本木ミッドタウン内にある Billboard Live Tokyo で開催された Musilogue Band のコンサート

それぞれのイベントの収録環境や収録方法は、[27] や [28] に記述されている。

SDM コンソーシアムでは GraphDB<sup>\*6</sup>を用いた SPARQL エンドポイントを <http://sdm.hongo.wide.ad.jp:7200/> で運営している。今回記述した Turtle ファイルを SPARQL エンドポイント上の web360square\_200425 リポジトリに格納した。GraphDB は Web 上に GUI を持ち、簡単な SPARQL クエリのテスト実行やグラフ生成をすることができる。<http://sdm.hongo.wide.ad.jp:7200/> にアクセスし web360square\_200425 リポジトリに対してコード 1 の SPARQL クエリを実行すると、Web360<sup>2</sup> で使用されている AudioSprite<sup>\*7</sup>形式の音源ファイルおよび HLS (HTTP Live Streaming) 形式の動画ファイルについての情報が得られるだろう。AudioSprite は複数の音声ファイルを時間

間隔をとりながら並べて、1 つの音声ファイルにまとめる技術であり、Turtle ファイル上では素材である各音源を AudioMedia クラスで定義し、AudioSprite をそれらの複合体として CompositeMedia クラスで定義している。HLS 動画ファイルも動画ファイルを小さなセグメントに分割しストリーミング配信する方式であるため、各セグメントの複合体と定義することは可能だが、今回のユースケースにおいてセグメントごとに分割して得たい情報がなかったため、セグメントを分割した定義はせず、HLS プレイリストファイルをもつ CompositeMedia として定義している。

```
1 PREFIX schema: <http://schema.org/>
2 PREFIX sdm: <http://sdm.hongo.wide.ad.jp/resource
  />
3 PREFIX sdmo: <http://sdm.hongo.wide.ad.jp/sdmo/>
4
5 SELECT DISTINCT ?event ?player_class ?url ?time ?
  start_at ?end_at ?x ?y ?z WHERE{
6   ?player
7   a ?player_class ;
8   schema:name "Web360Square" ;
9   sdmo:plays ?event ;
10  sdmo:input_from ?compositemedia.
11  ?compositemedia
12  schema:contentUrl ?url ;
13  sdmo:includes ?mediaevent .
14  ?mediaevent
15  sdmo:refers ?media ;
16  sdmo:time ?time .
17  ?media
18  sdmo:input_from ?recorder ;
19  sdmo:start_at ?start_at ;
20  sdmo:end_at ?end_at .
21  ?recorder
22  sdmo:local_x ?x ;
23  sdmo:local_y ?y ;
24  sdmo:local_z ?z .
25 }
```

コード 1: テスト用の SPARQL クエリ。Web360<sup>2</sup> で必要な情報が得られる。

クエリの実行結果には AudioSprite あるいは HLS 動画のコンテンツ URL および CompositeMedia クラスに包含されている各メディアの開始時間と再生区間、収録機器の位置情報が含まれる。現在公開されている Web360<sup>2</sup> ではこのクエリで得られたような情報はすべてソースコードに直接打ち込んである。よって、Web360<sup>2</sup> 上で SPARQL クエリを実行し、実行結果の値を適切に変数へ受け渡すことができれば Web360<sup>2</sup> と SDM Ontology の連携が実現できる。

\*4 <https://sdm-wg.github.io/web360square/>

\*5 <https://www.w3.org/TR/turtle/>

\*6 <http://graphdb.ontotext.com/>

\*7 <https://github.com/tonistiigi/audiosprite>

## 5.2 Web360<sup>2</sup> との連携

ローカル環境にて Web360<sup>2</sup> と SDM Ontology の連携を試作した。SPARQL エンドポイント上の web360square\_200425 リポジトリに対して SPARQL クエリを実行するには、[http://sdm.hongo.wide.ad.jp:7200/repositories/web360square\\_200425](http://sdm.hongo.wide.ad.jp:7200/repositories/web360square_200425) に向けて HTTP GET リクエストする際に、query パラメータとして SPARQL クエリを URI エンコードした値を渡せばよい。実行結果は JSON 形式で返ってくるので、適当なコードによって Web360<sup>2</sup> 上のメディアコンテンツ関連の変数に受け渡すことが可能である。図 6 はローカル環境にて Web360<sup>2</sup> と SDM Ontology の連携を動作させた際のスクリーンショットである。現在公開されている Web360<sup>2</sup> と同様の動作を確認することができた。



図 6: SDM Ontology と連携した Web360<sup>2</sup> の動作中スクリーンショット。

## 6. まとめと今後の課題

本稿では、メディアの収録・編集環境について階層的に記述する SDM Ontology Version 2.0 について述べた。従来の設計では、一般に動的な構造をとるメディア編集についての記述が難しい構造であったが、再帰的記述を取り入れる形に修正・拡張したことで動的な構造を取り扱い、ユースケースにあわせた自由な粒度の記述が可能となった。また、修正・拡張作業において、オントロジーの土台となるクラス設計を大きく見直し、より抽象度の高いシンプルな構造を目指した。抽象度を高める過程で SDM Ontology 単独では表現することが難しいような事象の記述にはそれぞれの分野専門の外部オントロジーと連携する方針を考えている。さらに、SDM Ontology Version 2.0 を活用したアプリケーション例として Web360<sup>2</sup> との連携について述べ、ローカル環境で正常に動作したことを報告した。

今後の課題としては、今回定義しきれなかった外部オントロジーとの連携部分についての構造決定が挙げられる。

Context クラスとの連携を考えている外部オントロジーは収録内容や対象、環境についての詳細情報を記述可能とし、実際の運用では必要不可欠なものが多いと考える。また、今回の Web360<sup>2</sup> との連携例において収録機器の位置座標は簡単のために座標系の定義なしに記述しているが、一般には収録における座標系とアプリケーションで活用する座標系は異なることが多い。時空間座標系と時空間座標を記述する CoordinateSystem クラス、Geometry クラスの議論についても進め、外部オントロジーとの連携を検討し、収録・編集における座標系とアプリケーションで活用する座標系とを参照し変換することができるような設計を考えていきたい。さらに、メディアの収録・編集段階の先にある配信・再生段階まで表現できるよう SDM Ontology の拡張の検討を続けていきたい。

## 参考文献

- [1] Tsukada, M., Ogawa, K., Ikeda, M., Sone, T., Niwa, K., Saito, S., Kasuya, T., Sunahara, H. and Esaki, H.: Software defined media: Virtualization of audio-visual services, *2017 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–7 (2017).
- [2] 塚田学, 小川景子, 池田雅弘, 曾根卓朗, 丹羽健太, 齊藤翔一郎, 粕谷貴司, 砂原秀樹, 江崎浩: Software Defined Media: 視聴空間サービスのソフトウェア制御, コンピュータソフトウェア, Vol. 34, No. 3, pp. 3.37–3.58 (2017).
- [3] Atarashi, R., Sone, T., Komohara, Y., Tsukada, M., Kasuya, T., Okumura, H., Ikeda, M. and Esaki, H.: The software defined media ontology for music events, *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music*, ACM, pp. 15–23 (2018).
- [4] 菰原裕, 塚田学, 江崎浩, 曾根卓朗, 池田雅弘, 高坂茂樹, 新麗, 新善文: SDM Ontology: Software Defined Media のメタデータ管理のための Ontology, マルチメディア, 分散協調とモバイルシンポジウム 2017 論文集, Vol. 2017, pp. 110–115 (2017).
- [5] Bizer, C., Heath, T. and Berners-Lee, T.: Linked data: The story so far, *Semantic services, interoperability and web applications: emerging concepts*, IGI Global, pp. 205–227 (2011).
- [6] Berners-Lee, T.: Linked Data - Design Issues (2009-06-18) (online), available from (<https://www.w3.org/DesignIssues/LinkedData.html>) (accessed 2020-04-26).
- [7] Berners-Lee, T.: 5-star Open Data (2015-08-31) (online), available from (<https://5stardata.info/en/>) (accessed 2020-04-26).
- [8] Raimond, Y., Abdallah, S. A., Sandler, M. B. and Giasson, F.: The Music Ontology., *ISMIR*, Vol. 2007, Cite-seer, p. 8th (2007).
- [9] Raimond, Y., Gängler, T., Giasson, F., Jacobson, K., Fazekas, G., Reinhardt, S. and Passant, A.: The Music Ontology (2013-07-22) (online), available from (<http://musicontology.com/>) (accessed 2020-04-26).
- [10] Raimond, Y. and Abdallah, S.: The Timeline Ontology (2007-10-29) (online), available from (<http://purl.org/NET/c4dm/timeline.owl>) (accessed 2020-04-26).
- [11] Raimond, Y. and Abdallah, S.: The Event Ontol-

- ogy (2007-10-25) (online), available from <http://purl.org/NET/c4dm/event.owl> (accessed 2020-04-26).
- [12] Davis, I. and Newman, R.: Expression of Core FRBR Concepts in RDF (2005-08-10) (online), available from <https://vocab.org/frbr/core> (accessed 2020-04-26).
  - [13] Brickley, D. and Miller, L.: FOAF Vocabulary Specification 0.99 (2014-01-14) (online), available from <http://xmlns.com/foaf/spec/> (accessed 2020-04-26).
  - [14] Fazekas, G. and Sandler, M. B.: The Studio Ontology Framework., *ISMIR*, pp. 471–476 (2011).
  - [15] Wilmering, T., Fazekas, G. and Sandler, M. B.: The Audio Effects Ontology., *ISMIR*, pp. 215–220 (2013).
  - [16] Wilmering, T. and Fazekas, G.: The Audio Effect Ontology (2016-06-12) (online), available from <https://w3id.org/aufx/ontology/1.0#> (accessed 2020-04-26).
  - [17] Kolozali, S., Barthet, M., Fazekas, G. and Sandler, M. B.: Knowledge Representation Issues in Musical Instrument Ontology Design., *ISMIR*, pp. 465–470 (2011).
  - [18] Kolozali, S., Fazekas, G., Barthet, M. and Sandler, M.: A framework for automatic ontology generation based on semantic audio analysis, *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Audio Engineering Society (2014).
  - [19] Allik, A., Fazekas, G. and Sandler, M. B.: An Ontology for Audio Features., *ISMIR*, pp. 73–79 (2016).
  - [20] Fazekas, G. and Allik, A.: Audio Features Ontology (2017-09-14) (online), available from <https://semantic-audio.github.io/afo/> (accessed 2020-04-26).
  - [21] Ceriani, M. and Fazekas, G.: Audio Commons ontology: a data model for an audio content ecosystem, *International Semantic Web Conference*, Springer, pp. 20–35 (2018).
  - [22] Fazekas, G. and Ceriani, M.: The Audio Commons Ontology (2019-01-29) (online), available from <https://w3id.org/ac-ontology/aco#> (accessed 2020-04-26).
  - [23] Turchet, L., Antoniazzi, F., Viola, F., Giunchiglia, F. and Fazekas, G.: The Internet of Musical Things Ontology, *Journal of Web Semantics*, Vol. 60, p. 100548 (2020).
  - [24] Antoniazzi, F.: Internet of Musical Things Ontology (IoMusT) (2020-02-05) (online), available from <https://fr4ncidir.github.io/IoMusT/> (accessed 2020-04-26).
  - [25] Guha, R. V., Brickley, D. and Macbeth, S.: Schema.org: evolution of structured data on the web, *Communications of the ACM*, Vol. 59, No. 2, pp. 44–51 (2016).
  - [26] 加藤慎, 池田友洋, 川守田光昭, 塚田学, 江崎浩: Web360<sup>2</sup>: インタラクティブな3D視聴体験を提供するWebアプリケーション, 研究報告デジタルコンテンツクリエーション (DCC), Vol. 2019, No. 14, pp. 1–8 (2019).
  - [27] 塚田学, 菰原裕, 粕谷貴司, 新居英明, 高坂茂樹, 小川景子, 江崎浩: SDM360<sup>2</sup>: インタラクティブ3Dコンテンツの自由視聴点再生, 情報処理学会論文誌デジタルコンテンツ (DCON), Vol. 6, No. 2, pp. 10–23 (2018).
  - [28] Ikeda, M., Sone, T., Niwa, K., Saito, S., Tsukada, M. and Esaki, H.: New recording application for software defined media, *Audio Engineering Society Convention 141*, Audio Engineering Society (2016).