東京大学
情報理工学系研究科 電子情報学専攻
修士論文

時空間メディアを管理するオントロジーと インタラクティブ視聴体験への応用

An Ontology for Spatio-Temporal Media Management and Application to Interactive 3D Audio-Visual Viewer

48-196416

加藤 慎

Shin Kato

指導教員　江崎 浩 教授

2021 年 1 月

概要

メディアコンテンツの制作は収録, 編集・加工, 配信, 再生の四段階に分割して考えることができる．著者が所属する SDM (Software Defined Media) コンソーシアムでは，三次元映像・音声メディアを管理するための構造フレームワークとして SDM Ontology の提案を進めてきた．SDM Ontology は制作フローの収録段階を対象として，様々かつ膨大なメタデータを階層構造に整理し，RDF (Resource Description Framework) 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を可能にした．本研究では，メディア収録段階より先の制作フローに焦点を当て，特にメディア編集・加工段階を SDM Ontology の記述対象範囲に加えるとともに，従来の構造について考察し直し，音楽イベントに限らず時空間情報をもつメディアを管理する，より汎用的なフレームワークとして SDM Ontology Version 2 を提案する．また，SDM Ontology Version 2 の活用例として，インタラクティブに 360° 動画および立体音響を視聴体験できる Web アプリケーション「Web360^2」を開発した．Web360^2 は SDM Ontology Version 2 に準拠するクラシックコンサートおよびジャズセッションイベントのデータセットを問い合わせて動作する．このアプリケーションについて，Interop Tokyo 2019 での対面アンケート，2020 年 12 月より開始した Web アンケートの 2 種類の主観評価を実施し，インタラクティブ視聴体験の有効性や異なる視聴環境が及ぼす影響を検証・考察した．さらに，SDM Ontology Version 2 の普及活動の一環として，オントロジーの語彙体系を説明する Web ドキュメントを作成および公開し，LOD の普及促進を目的に毎年開催される「LOD チャレンジ」のアプリケーション部門に Web360^2，データセット部門に SDM Ontology Version 2 準拠のデータセットを提出した．LOD チャレンジにおいては，データセット部門への応募作品で最優秀賞およびスポンサー賞を受賞した．

Abstract

The SDM (Software Defined Media) consortium, to which the author belongs, has been proposing the SDM Ontology for managing 3D video and audio media. The production of media content can be divided into four stages: recording, editing/processing, distribution, and playback. The SDM Ontology organizes various and huge amounts of metadata into a hierarchical structure for the recording stage of the production flow and makes it accessible as open data. This data is LOD (Linked Open Data) described by RDF (Resource Description Framework). In this study, we propose the SDM Ontology Version 2 as a more general framework by modifying and extending Version 1. In the SDM Ontology Version 2, we focus on the editing/processing stage beyond the recording stage and make it possible to manage spatio-temporal media, not limited to music events. We also designed and implemented a Web application, called "Web360^2," which plays 360-degree video and object-based 3D sounds interactively on the Web. Web360^2 plays a dataset of classical concert and jazz session events following the SDM Ontology Version 2. We conducted two types of subjective evaluations about this application: a face-to-face survey at Interop Tokyo 2019 and a web survey that started in December 2020. In other work to promote the SDM Ontology Version 2, we created and published a Web document describing the vocabularies. We submitted a dateset following the SDM Ontology Version 2 and Web360^2 to the LOD Challenge Japan, which is held annually to promote LOD, and won the first prize and the sponsor prize.

目次

第 1 章　序論 1
1.1　はじめに 1
1.2　本研究について 2
1.3　本論文の構成 3

第 2 章　関連技術・研究 4
2.1　RDF/LOD 4
2.2　関連オントロジー 5
2.3　立体音響技術 6
2.4　三次元映像・音声を用いたインタラクティブアプリケーション 7

第 3 章　SDM Ontology 8
3.1　要求事項 9
3.2　SDM Ontology Version 2 の概要 10

第 4 章　SDM Ontology の応用—Web360^2 17
4.1　要求事項 17
4.2　設計 18
4.3　素材データ 19
4.4　実装 22

第 5 章　評価・普及活動 33
5.1　SDM Ontology の評価・普及活動 33
5.2　Web360^2 の評価 37

第 6 章　結論 46
6.1　まとめ 46
6.2　今後の課題 46

発表文献 48

参考文献 49

謝辞 53

付録 A SDM Ontology Version 2 準拠のデータ例 55

付録 B SDM Ontology Version 2 の語彙一覧 58

B.1 クラス一覧 59

B.2 オブジェクトプロパティ一覧 66

B.3 データ型プロパティ一覧 71

図目次

1.1 メディアコンテンツ制作フローの概要 2

3.1 メディア制作フローの概要と SDM Ontology の記述可能な範囲 8
3.2 SDM Ontology Version 2 の構造概要 11
3.3 SDM Ontology Version 2 の Context クラスの概要 13
3.4 SDM Ontology Version 2 の Recorder クラス，Processor クラス，Player クラスの概要 14
3.5 SDM Ontology Version 2 の Media クラスの概要 15

4.1 Web360^2 のシステム設計 19
4.2 SDM Ontology Version 2 に準拠するデータ作成の概要 20
4.3 素材データ収録時の収録機器配置 21
4.4 素材データ収録を SDM Ontology Version 2 に準拠するように記述した RDF データ（一部抜粋） 21
4.5 Web360^2 の実装概要 22
4.6 視聴体験中の Web360^2 のスクリーンショット 23
4.7 Web360^2 のイベント選択画面のスクリーンショット 24
4.8 Web360^2 におけるイベント情報取得クエリ 24
4.9 Web360^2 におけるイベント情報取得の概要 25
4.10 Web360^2 における視聴情報取得クエリ 26
4.11 Web360^2 における視聴情報取得の概要 27
4.12 音声を可視化する処理 29
4.13 死角にある棘状オブジェクトの非表示 31

5.1 Interop Tokyo 2019 での対面調査におけるアンケートの回答比率 39
5.2 オンライン調査におけるアンケートの回答比率 41
5.3 デバイスの種類による評価の比較 42
5.4 タブレット・スマートフォンにおける OS による評価の比較 42
5.5 ジャイロセンサの使用に関する評価の比較 43
5.6 ヘッドホンの使用に関する評価の比較 43

A.1 最小限の収録形態の例 55
A.2 SDM Ontology Version 2 準拠の単純な RDF データの例 57

表目次

4.1 Web360^2 と関連システムとの比較 18

5.1 知識の網羅性と人気度に関する指標 34

B.1 RDF の名前空間 58

第 1 章

序論

1.1 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][*1]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 [3] [4]（SDM Ontology Version 1 と呼ぶ）の提案を進めてきた．メディアコンテンツの制作は**図 1.1** のように，収録，編集・加工，配信，再生の四段階に分割して考えることができる．この制作フローにおいて，収録や編集・加工したメディアのデータファイルのほかに，収録環境や収録対象の詳細情報，編集・加工の環境および手順など，様々かつ膨大なメタデータが発生するが，これまで，メディアのメタデータには確立された管理方法が存在せず，コンテンツ制作をおこなう各組織・各団体内で独自に管理されてきた．SDM Ontology Version 1 は，収録段階を対象として，収録環境や収録対象についてのメタデータを階層構造に整理し，RDF (Resource Description Framework)[*2] 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を可能にした．

[*1] https://sdm.wide.ad.jp/
[*2] https://www.w3.org/RDF/

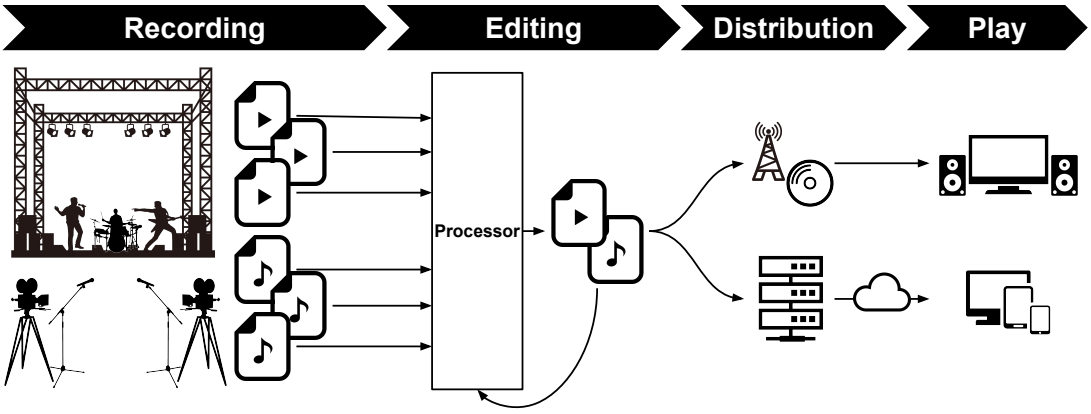


図 1.1. メディアコンテンツ制作フローの概要．収録，編集・加工，配信，再生の四段階に分割して考えることができる．

1.2　本研究について

SDM Ontology Version 1 により，収録に携わっていない第三者でも収録の内容や状況を正しく把握することができ，誰でも自由にコンテンツの制作やアプリケーションの開発をおこなうことができる開かれた環境が生まれ，新たなメディアの表現方法が開拓されていくことを期待してきた．しかしながら，コンテンツの制作やアプリケーションの開発において，収録段階のメディアデータやメタデータの参照だけでなく，編集・加工されたメディアデータ，メタデータも参照したいという要求は一般的であるにも関わらず，SDM Ontology Version 1 で記述可能な範囲は収録段階にとどまり，編集・加工段階は対象外であった．

本研究では SDM Ontology の対象範囲にメディア編集・加工段階を加えるとともに，従来の構造について考察し直し，修正および拡張した SDM Ontology Version 2 を提案する．また，SDM Ontology Version 2 の普及を促進するための活動をおこなった．本研究の貢献を以下にまとめる．

SDM Ontology Version 2 の提案

Version 1 では三次元映像・音声メディアのうち，特にライブやコンサートなどの音楽イベントを前提としており，音楽イベントに特化した語彙の定義もおこなっていたが，Version 2 では音楽イベントに限らず利用できるように，SDM Ontology が提供する語彙はメディアコンテンツの制作フローの核となる部分にとどめ，イベント内容などの具体的な情報は適宜，外部のオントロジーと組み合わせて記述することとする．さらに，三次元映像・音声メディアに限らず，センサー群によるデータメディアなども対象可能となるように，より抽象度の高い枠組みである「時空間情報をもつメディア」を扱うことを意識してオントロジーを拡張した．

SDM Ontology Version 2 の普及活動

1. SDM Ontology Version 2 準拠のオープンデータを用いたアプリケーションの開発
   SDM コンソーシアムが過去に収録したクラシックコンサートとジャズセッションイベントをメディア編集・加工の工程も含めて，SDM Ontology Version 2 準拠のオープンデータとして記述し，このデータを問い合わせて動作するアプリケーション「Web360^2」を開発した．Web360^2 はインタラクティブに 360° 動画および立体音響を視聴体験できる Web アプリケーションである．また，Web360^2 の開発は SDM Ontology Version 2 の普及目的だけでなく，オントロジーの構造を考察する上でコンテンツ制作者視点の要求を取り入れる意図もあった．
2. Web ドキュメントの公開
   オントロジーは OWL (Web Ontology Language)[*3]というオントロジー記述言語によって記述され，人間が容易に読むことのできる形式になっていない．SDM Ontology Version 2 の語彙体系は，OWL ファイルを Web ドキュメントに変換するプロジェクト WIDOCO (WIzard for DOCumenting Ontologies) [5] を利用して，Web ドキュメント[*4]として公開されている．
3. LOD チャレンジ 2020 への応募
   LOD チャレンジ[*5]とは LOD の普及促進を目的に 2011 年から毎年開催されているイベントである．今回，LOD に興味・関心のある人々を中心に我々の活動を周知してもらう目的で，アプリケーション部門に Web360^2，データセット部門に SDM Ontology Version 2 準拠のオープンデータを提出し，データセット部門において最優秀賞およびスポンサー賞を受賞した．

1.3 本論文の構成

本論文では，SDM Ontology Version 2 とその応用として開発したインタラクティブ Web アプリケーションの Web360^2 を中心に述べていく．2 章では，オントロジーの基本事項である RDF/LOD や SDM Ontology に関連するオントロジー，さらに，三次元音声を扱うための立体音響の手法，三次元映像・音声を用いたインタラクティブアプリケーションについて紹介する．3 章で SDM Ontology Version 2 の概要説明をおこない，4 章で SDM Ontology Version 2 との連携について述べつつ Web360^2 の設計・実装について説明する．5 章では，SDM Ontology Version 2 と Web360^2 の評価を述べる．また，オントロジーは世間に広く普及して活用されることが評価の上で特に重要であるため，ここで SDM Ontology Version 2 を普及するためにおこなった活動を紹介する．最後に，6 章で本研究のまとめを述べる．

[*3] https://www.w3.org/OWL/
[*4] https://sdm.wide.ad.jp/sdmo/
[*5] https://2020.lodc.jp/

第 2 章

関連技術・研究

本章では，オントロジーの関連技術である 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．さらに，三次元映像・音声を用いたアプリケーションに関連して立体音響技術と既存のインタラクティブアプリケーションを紹介する．

2.1 RDF/LOD

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[*1]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 [6] [7].

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 スキームの 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

[*1] https://www.w3.org/2009/sparql/wiki/Main_Page

LOD (Linked Open Data) とはオープンライセンスのもとで公開され，無料での再利用が可能な 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 段階評価の枠組みである 5-star Open Data [8] 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 Version 2 は記述対象を音楽イベントに限らない構造に拡張・修正されたが，依然として音楽イベントとの関わりは大きい．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 [9] [10] 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 [11] や Event Ontology [12]，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 [13]，FOAF (Friend of a Friend) [14] 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 [15] 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 [16] [17] 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 [18] 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 [19] 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 [20] [21] 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 [22] [23] 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 [24] [25] が提案された．

2.3　立体音響技術

立体音響技術には，様々な方法が存在する．現在，国際標準化機構 (ISO) および国際電気標準会議 (IEC) のワーキンググループである Moving Picture Experts Group (MPEG) が標準化を進めている MPEG-H において，MPEG-H Part 3 として 3D Audio（立体音響）に関する項目がある [26]．MPEG-H 3D Audio が許容する立体音響の手法は，チャンネルベース，オブジェクトベース，HOA（Higher-order Ambisonics，高次アンビソニックス）の 3 つである [27]．

チャンネルベースオーディオは，2.0 チャンネルのステレオサウンドから 5.1 チャンネルを代表とする多チャンネルのサラウンドサウンドやイマーシブサウンドへと発展している．チャンネルベースオーディオはスピーカーの数や視聴者に対するスピーカーの位置関係の情報をもとに，再生環境に紐付けて音響データを記録・編集する．したがって，再生環境の変更に対しては環境に紐付けられた別の音響データが必要になる．あるいは，もとの音響データを改変する必要がある [28]．また，視聴者の動きに追従して音響に動きを加えることも難しく，柔軟性やインタラクティブ性に欠ける．

オブジェクトベースオーディオでは，音響データとともに 3 次元の位置情報がメタデータとして記録される．再生環境を事前に想定する必要はなく，スピーカーの数や位置に適した音響を柔軟にレンダリングしながら再生することができる．また，音響データがもつ 3 次元位置情報の時間変化に対応しながらレンダリングすることが可能であるため音響オブジェクトの動きを表現できるほか，視聴者の位置移動や頭部の回転動作に音響を追従させることもできる．コンテンツ制作において，コンテンツに含まれる楽器の演奏や音声，効果音にはパンニングツールによって 3 次元位置情報が割り当てられており，今日のスタジオには，すでに膨大な数のオブジェクトベースコンテンツが存在している．大部分は配信される際にオブジェクトベースオーディオとしてではなく，従来のチャンネルベースオーディオとしてミキシングされてしまう [28] が，主に映画館やホームシアターへの導入が進む Dolby Atmos [29] [30] や DTS:X [31]，AuroMax [32] ではオブジェクトベースオーディオを採用，あるいは，従来のチャンネルベースオーディオとともに併用している．

HOA は音響を，1 点を中心に広がる球状の音場として記録する手法である．数学的には音場を球調和関数展開を利用して表現する方法であり，従来より研究されてきたアンビソニックスでは球調和関数の 0 次および 1 次係数にのみ着目するが，HOA では 2 次以上の係数も考慮することで，より高品質な音場を表現できるようになった [33]．HOA/アンビソニックスのもつ空間的な情報はオブジェクトベースオーディオのような明示的な幾何学的メタデータではないため，音場内のそれぞれの音響オブジェクトへアクセスすることは難しい [28]．HOA/アンビソニックスの収録には，360° 全方位を収音するためにアンビソニックマイクと呼ばれる特殊なマイクが必要となる．収録されたデータはマイクを中心とする球状の音場を再現するた

め，視聴者の頭部回転動作に対しては音場の回転によって容易に追従することができる．一方で，視聴者は常に音場の球の中心に位置することが求められるため，位置移動への追従は難しい．全天球カメラによって撮影された 360° 動画も同様であり，映像の回転は容易だが中心点から移動することはできない．このように，HOA/アンビソニックスと全天球カメラは同じ特性をもち，相性がよいと言える．最近では，RICOH THETA Z1, THETA V [34] などの民生用全天球カメラにもアンビソニックマイクが搭載されるようになり，YouTube や Facebook の 360° 動画サービスでも HOA/アンビソニックスに対応している．

2.4　三次元映像・音声を用いたインタラクティブアプリケーション

著者が所属する SDM コンソーシアム [1] [2][*2]の先行研究 [35] では，タブレット端末を前提とした収録済みコンテンツの再生アプリケーション SDM360[2] を開発し， [36] [37] では，HMD (Head Mounted Display) を前提としたライブ配信コンテンツ対応のアプリケーション LiVRation を開発した．360° 動画を投影し，オブジェクトベースオーディオの手法をもとに音響オブジェクトをマッピングした仮想空間内で，視聴者の頭部動作に応じた映像・音響の追従を行う．また，視聴者の操作によって仮想空間内を移動することができる．さらに，各音源がオブジェクトとして独立していることを利用して，音源ごとに ON/OFF を切り替えたり，音量調節を行うことができるなどインタラクティブな 3D コンテンツの再生を実現している．どちらも Unity を用いて開発されており，開発・実行には Unity が必要である．

Google Creative Lab と Song Exploder が 2017 年に公開した実験サイト Inside Music [38] は，WebVR と Web Audio の技術を用いており，Web 上で仮想空間におけるオブジェクトベースオーディオによる立体音響を体験できる．仮想空間内には 360° 動画の投影はされず，音響オブジェクトなどのコンポーネントのみが投影される．空間内は自在に水平移動することができ，視聴位置や視聴角度の変化に音響が追従する．また，各音源オブジェクトをクリックすることで ON/OFF の切り替えができる．専用のアプリケーションのインストールは不要で，Web ブラウザを利用することができれば，HMD や PC，モバイル端末から非常に手軽に体験できる．ソースコードは GitHub[*3]に公開されており，Apache License, Version 2.0 [39] に則り，コードの利用を提示することで Inside Music のシステムを利用したインタラクティブな VR 体験アプリケーションの制作が可能になる．

[*2] https://sdm.wide.ad.jp/

[*3] https://github.com/googlecreativelab/inside-music

第 3 章

SDM Ontology

SDM コンソーシアムが提案する 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収録，編集，配信，再生の段階に分かれ，従来の提案である SDM Ontology Version 1 [3] [4] では，メディアの収録段階における環境や内容，対象に関する記述を可能とする設計となっていたが，収録段階に続く，編集，配信，再生の段階については記述対象外であった．本研究では主にメディア編集・加工段階に関する記述も対象範囲に含むように SDM Ontology Version 1 の構造を見直し，修正および拡張した SDM Ontology Version 2 [40] を提案する．本論文では [40] の提案をもとに語彙の構造や命名規則を整理し，オントロジーを洗練させた．SDM Ontology の記述可能な範囲の概要を**図 3.1** に示す．

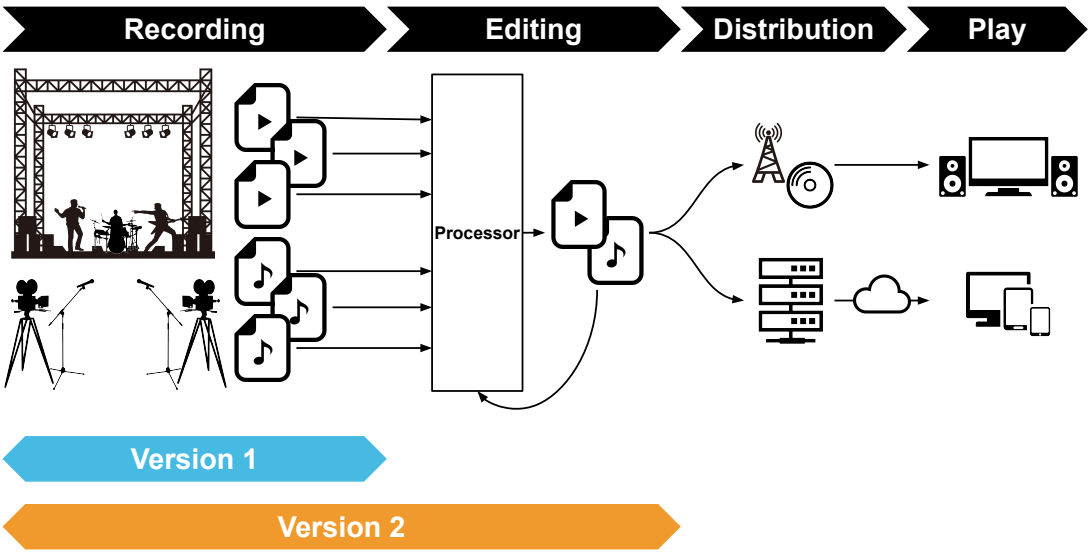


図 3.1. メディア制作フローの概要と SDM Ontology の記述可能な範囲．SDM Ontology Version 1 では収録段階まで，SDM Ontology Version 2 では収録段階および編集・加工段階までを主な対象範囲とする．

3.1 要求事項

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．したがって，収録段階のみを対象としていた SDM Ontology Version 1 では，静的データを記述することが可能な構造に設計されていた．一方で，メディアの編集・加工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の静的データを記述するための構造設計では対応しきれない．

また，Version 1 では，カメラやマイクなど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場合もある．さらには，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い．

このような問題を解消するために SDM Ontology Version 1 の設計を見直して，オントロジーを修正・拡張する必要がある．修正・拡張に関する主な要求事項を以下に記す．

動的な情報を記述するための再帰構造の導入

メディアの編集・加工作業は多数の編集工程を経ておこなわれる場合が多い．素材となる収録データを加工して「1 次加工物」を生成し，これを素材としてさらに加工することで「2 次加工物」を生成する．この過程を繰り返して生まれた「n 次加工物」がメディアコンテンツの最終的な完成形として世の中に出回っていくことが一般的である．繰り返しによる動的情報は再帰的な構造によって記述することが可能であり，メディアデータの編集履歴を作成することが可能となる．また，再帰的記述は動的な構造の表現だけでなく包含関係の表現も可能とする．包含関係の表現は静的情報の記述に対しても非常に有効であり，Version 1 では記述できなかった詳細な情報構造を表現できるようになる．例えば，マイクアレイとは，複数のマイクを配置し単一のマイクだけでは得られない音の空間情報を収録する収録機器であり，マイクアレイ自体を 1 つの収録機器と見ることもできれば，マイクアレイがもつ単一のマイクを収録機器の最小単位としてマイクアレイはそれらを包含する収録機器である，と記述することもできる．逆に，同じ対象を収録する収録機器群をグループ化して収録機器グループを記述したり，収録機器グループ群をさらにグループ化することで収録機器環境そのものを記述することも可能となる．このように再帰的記述の導入は，メディアの編集・加工段階の表現だけでなく詳細な情報の表現にも貢献し，SDM Ontology Version 2 における大きな変更点の 1 つである．

時空間メディアへの抽象化

不足していたメディア編集・加工段階の構造設計と並行して従来設計の見直しをおこなう．SDM Ontology Version 1 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る．Version 2 では，オントロジーの表現対象はメディアコンテンツの制作フローの核となる部分にとどめ，分野ごとの専門的な知識やイベント内容などの特に具体的な情報は適宜外部オントロジーと組み合わせて記述することとする．さらに，対象メディアを三次元映像・音声メディア前提だったものから「時空間情報をもつメディア」へと抽象度を高め，センサー群によるデータメディアなども表現対象となるようにオントロジーを拡張した．このような抽象化によって SDM Ontology はより幅広い種類のメディアを取り扱い，メディアコンテンツの制作を管理する汎用性の高いオントロジーとなる．

3.2　SDM Ontology Version 2 の概要

3.1 節で述べた要求事項をもとにオントロジーの修正・拡張をおこない，SDM Ontology Version 2 を提案する．

以下の各項では，オントロジーの構造を**図 3.2** から**図 3.3**，**図 3.4**，**図 3.5** までに示すように図示しながら説明する．図中の各ノードはオントロジーのクラスであり，赤字で記されたクラスは SDM Ontology において特に重要な，オントロジー構造の核である基本クラスを表し，青字のクラスはその他のクラスを表す．クラスノード内に書かれた内容は Datatype Property（データ型プロパティ）と呼ばれるオントロジーの語彙（述語）であり，目的語に文字列や数値，URI などのリテラルをとる．ノード間をつなぐ黒色矢印の傍らに書かれているのは Object Property（オブジェクトプロパティ）と呼ばれる語彙（述語）であり，矢印の根本側のクラスがもつ述語である．目的語には矢印が指す先のクラスインスタンスをとり，クラスインスタンス間の関係性を表す．一方で，**図 3.3** 以降に描かれるノード間をつなぐオレンジ色矢印はクラスの親子関係を表し，矢印根本のクラスは矢印が指すクラスのサブクラスである．

それぞれの語彙は原則として，SDM Ontology のベース URI である `http://sdm.hongo.wide.ad.jp/sdmo/` からの相対 URI で示している．**図 3.2** の Media クラスではノードに “Media” と記されているが，正式には `http://sdm.hongo.wide.ad.jp/sdmo/Media` という絶対 URI によって識別される．また，Media クラスのノード内の “startAt” というデータ型プロパティは `http://sdm.hongo.wide.ad.jp/sdmo/startAt` で識別される．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，接頭辞 “schema:” が `http://schema.org/` [41] を表し，相対 URI が “name” である “schema:name” は，`http://schema.org/name` と等価である．ベース URI や接頭辞の情報は図中では省略しているが，通常，オントロジー定義を記述する OWL ファイルや RDF データのファイルの先頭にまとめて記述される．

3.2.1　構造概要

SDM Ontology Version 2 の構造概要を**図 3.2** に示す．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

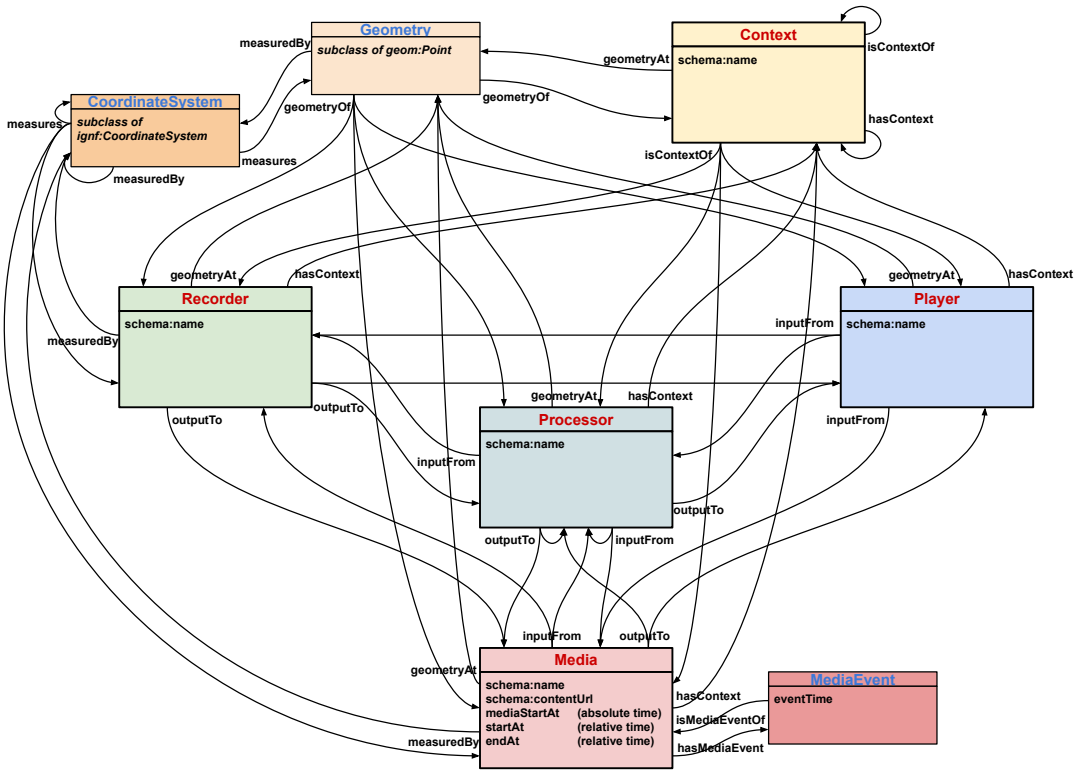


図 3.2. SDM Ontology Version 2 の構造概要．各ノードはクラスであり，赤字は基本クラスを表し，青字はその他のクラスを表す．ノード内に書かれた語彙はデータ型プロパティであり，ノード間をつなぐ黒色矢印はオブジェクトプロパティである．

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．SDM Ontology はここでいうデジタルデータを Media と呼び，Media の収録，編集・加工，再生[*1]の流れを包括的に記述するために設計したオントロジーである．現実空間の事象を Media に記録するシステムを Recorder，Media を編集・加工，または，分析，生成するシステムを Processor，Media を再生するシステムを Player と呼ぶ．また，Media やこれらのシステムは様々なメタデータをもち，これを Context と呼ぶ．ここで述べた要素をすべてメディア制作において重要であると考え，SDM Ontology Version 2 ではこれらの要素を表現するための Context クラス，Recorder クラス，Processor クラス，Player クラス，Media クラスの 5 つを基本クラスとする．基本クラスの詳細は 3.2.2 項から 3.2.4 項にかけて説明する．

3.1 節で述べたように，SDM Ontology Version 2 では「時空間情報をもつメディア」を取り扱うことを前提とするため，メタデータの中でも特に時空間情報は重要な意味をもつと考え，時空間メタデータの扱いに特化した CoordinateSystem クラス，Geometry クラス

[*1] SDM Ontology Version 2 は特に編集・加工段階の表現に注力して設計したが，後々必要になってくるだろう再生段階の記述に関しても最低限導入している．

を定義し，Context クラスとは分離して表現する．CoordinateSystem クラスで時空間座標系を定義し，Geometry クラスによって時空間座標系上の点（座標）を表現する．例えば，CoordinateSystem クラスで右手系の xyz 空間座標系を定義し，Geometry クラスではこの座標系における点 $(1, 2, 3)$ を表現するなどして運用する．CoordinateSystem クラスのインスタンスと Geometry クラスのインスタンスは measures/measuredBy プロパティで相互参照し座標系定義をおこない，Geometry クラスのインスタンスは基本クラスの各インスタンスと geometryOf/geometryAt プロパティのリンクによって時空間メタデータを与える．CoordinateSystem クラスと Geometry クラスはそれぞれ，ignf:CoordinateSystem [42][*2]，geom:Point [43][*3]のサブクラスとして定義しているが，SDM Ontology を用いたメディア管理の実運用時には座標系定義や座標記述よりも多くの時空間情報を表現しなければならないことが予想され，表現力の不足が懸念される．CoordinateSystem クラスと Geometry クラスの詳細な構造の検討や整備については今後の課題としたい．

図 3.2 右下にある MediaEvent クラスは Media クラスの包含関係を表現するための補助クラスであり，詳細は 3.2.4 項で述べる．

3.2.2　Context クラス

Context クラスは SDM Ontology の基本クラスがもつ様々なメタデータを包括的に記述するためのクラスである．メタデータの種類を大別して SDMEvent クラス，Target クラス，Program クラスをサブクラスとしてもつ．Context クラス，および，そのサブクラスに関する構造図を**図 3.3** に示す．

SDMEvent クラスは収録対象である現実空間内での事象（イベント）に関する日時や場所をはじめとする様々な収録環境のメタデータを記述し，Target クラスは収録対象である人物やモノを記述する．収録対象のイベントがもついくつかの意味をもつ事象のまとまりをプログラムと呼び，Program クラスではこれを記述する．例えば，あるコンサートを記述する場合，コンサート名や開催日時・場所は SDMEvent クラス，指揮者や演奏者，楽器は Target クラス，演目は Program クラスのように整理して記述可能である．ただし，プログラムを収録対象のモノであるという見方をすれば，Program クラスの内容は Target クラスで記述することも可能であるため，将来的に Program クラスは廃止される可能性がある．

SDM Ontology Version 1 では SDMEvent クラスや Target クラスは，Context クラスと並ぶ基本クラスとして定義されていたが，いずれも大局的に見るとメタデータの記述をおこなう共通の働きをもつことから SDMEvent クラスや Target クラスを Context クラスのサブクラスとして定義し直した．また，Version 1 は音楽イベントのメディア収録を前提として設計したオントロジーであったため，音楽イベントに特化したクラスなど具体的で詳細なクラスがいくつか用意されていたが，Version 2 では音楽イベントに限らず「時空間情報をもつメディア」を取り扱うことのできるよう具体的なクラス・プロパティは削除し構造をシンプル

[*2] http://data.ign.fr/def/ignf#CoordinateSystem
[*3] http://data.ign.fr/def/geometrie#Point

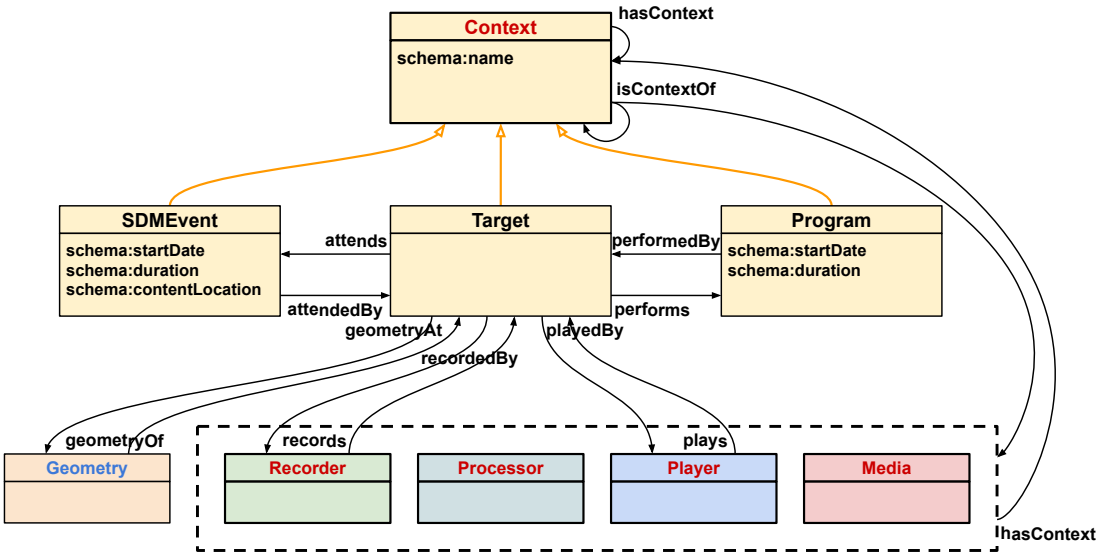


図 3.3. SDM Ontology Version 2 の Context クラスの概要．各ノードはクラスであり，ノード内に書かれた語彙はデータ型プロパティ，ノード間をつなぐ黒色矢印はオブジェクトプロパティである．また，ノード間をつなぐオレンジ色矢印はクラスの親子関係を表す．

に，抽象度を高めた．したがって，Version 2 を用いて音楽イベントのデータを記述する場合には SDM Ontology でメディア制作の核の部分を記述し，演奏された楽曲や演奏者，楽器などの情報は音楽を専門とする外部のオントロジーと連携して記述することになる．

3.2.3 Recorder クラス，Processor クラス，Player クラス

Recorder クラス，Processor クラス，Player クラスはそれぞれ**図 3.4** に示すように用途を詳細に表現できるサブクラスと包含関係を記述するためのサブクラスをもつ．

Recorder クラス，Processor クラス，Player クラスのそれぞれで同じ構造をとるため，ここでは代表して Recorder クラスについて述べる．Recorder クラスのサブクラスにはより用途が詳細な DataRecorder クラス, AudioRecorder クラス, VideoRecorder クラスと包含関係を表現する CompositeRecoder 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での動画撮影では，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る．SDM Ontology Version 1 ではこれらを Recorder クラス，あるいは，映像収録がメインである場合は VideoRecorder クラスのインスタンスとして記述することは可能であった．Version 2 では従来通り単純に Recorder クラスあるいは VideoRecorder クラスのインスタンスと記述することもできれば，ビデオカメラやスマートフォンがもつ映像収録機構を切り出して VideoRecorder クラスのインスタンス，ステレオ音声収録機構を 2ch まとめて，あるいは，1ch ずつ切り出して AudioRecorder クラスのインスタンスとして定義し，ビデオカメラやスマートフォン自体はこれらの VideoRecorder クラス，AudioRecorder クラスインスタンスを

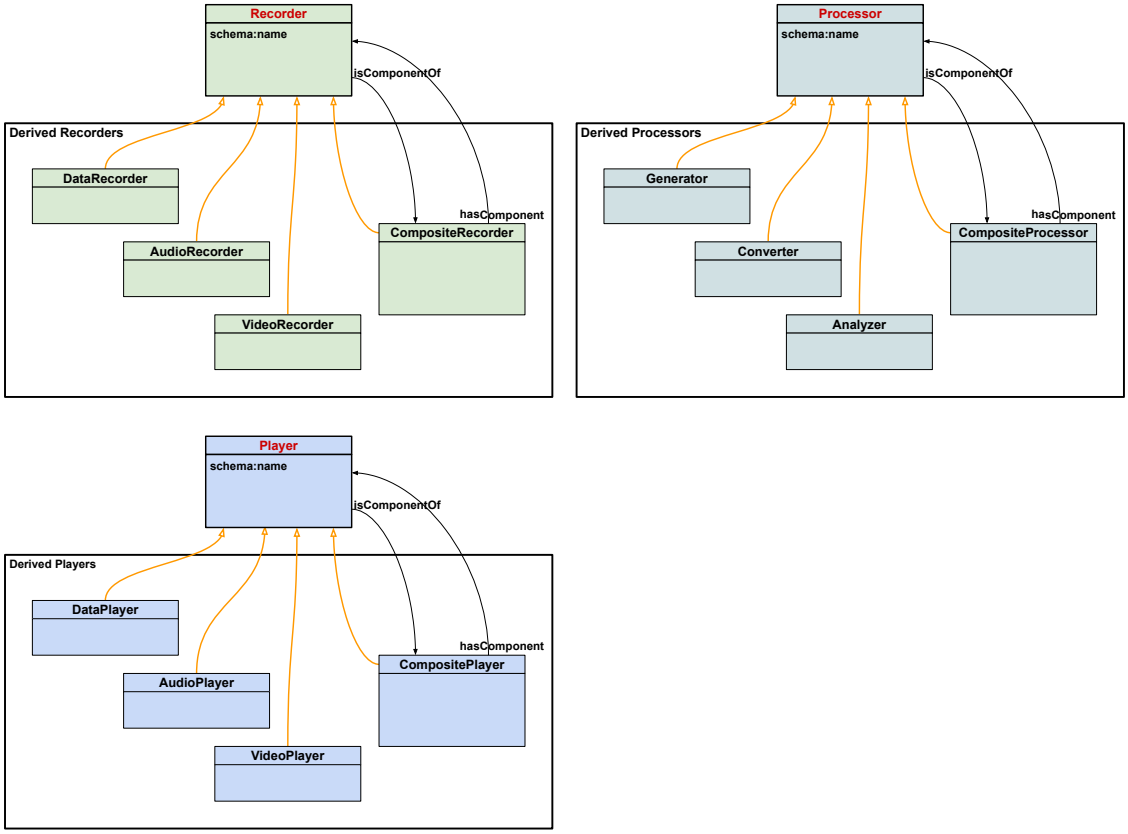


図 3.4. SDM Ontology Version 2 の Recorder クラス，Processor クラス，Player クラスの概要．各ノードはクラスであり，ノード内に書かれた語彙はデータ型プロパティ，ノード間をつなぐ黒色矢印はオブジェクトプロパティである．また，ノード間をつなぐオレンジ色矢印はクラスの親子関係を表す．

包含する CompositeRecorder クラスのインスタンスとして記述することも可能となる．3.1 節で述べたマイクアレイの例も同様の方法で記述することができる．また，カメラやマイクあるいはセンサなどのそれぞれ独立な収録機器が同じ対象を収録していた場合，これらをグループ化したい場合がある．この場合にも CompositeRecorder クラスによる包含関係の記述によって収録機器グループの記述が可能であり，再帰的な包含関係の記述によって収録環境全体を 1 つのグループとして記述することも可能となる．Processor クラス，Player クラスでも同様であり，包含関係を記述する “Composite” クラスの再帰的な特性によって収録，編集・加工，再生システムはそれぞれ機器の最小単位（ミクロ）から複合系（マクロ）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．

3.2.4　Media クラス

Media クラスについても Recorder クラス，Processor クラス，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 3.5** に示すように，Recorder クラス，Processor クラス，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

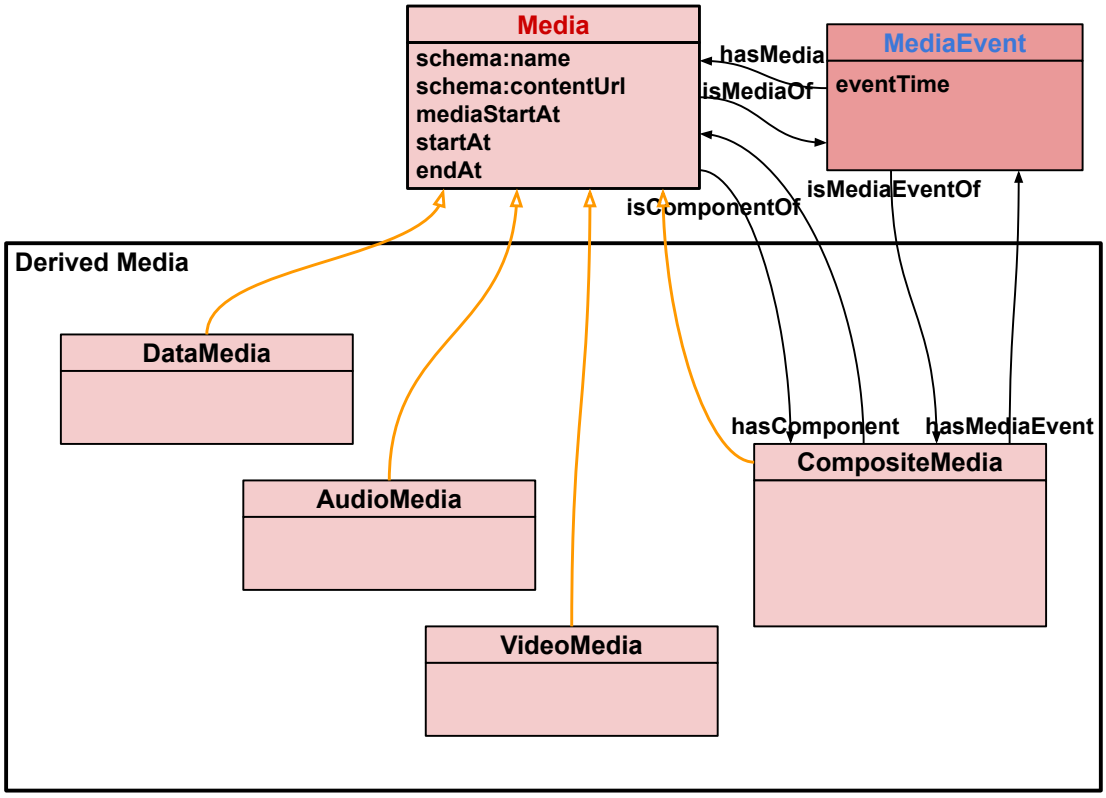


図 3.5. SDM Ontology Version 2 の Media クラスの概要．各ノードはクラスであり，ノード内に書かれた語彙はデータ型プロパティ，ノード間をつなぐ黒色矢印はオブジェクトプロパティである．また，ノード間をつなぐオレンジ色矢印はクラスの親子関係を表す．

する．

メディア管理においては，メディアデータの記録時刻を管理するための絶対的な時間軸（通常は現実空間内の時間軸）のほかに，記録されたメディアデータの再生区間を表す相対的な時間軸など，複数の時間軸を意識する必要があり，時間情報の取り扱いは重要である．Media クラスには時間情報を扱うプロパティとして，メディアデータの記録を開始した時刻を表す mediaStartAt プロパティ，メディアファイルの有効な再生区間を示すための startAt/endAt プロパティを用意した．mediaStartAt プロパティが絶対時刻を記述するのに対して，startAt/endAt プロパティはメディアファイルの先頭を原点とする相対時刻を指定することに注意されたい．画像データなど時間情報を区間としてもたないメディアデータについては startAt/endAt プロパティをともに 0 とし，再生時間を 0 と定義することで表現することができる．

メディアのグループ化においては，Recorder クラス，Processor クラス，Player クラスの包含関係とは異なり，グループ化した各メディアの時間的な配置も重要になる場合が多いと考える．CompositeMedia クラスを用いてメディアの包含関係を記述する際には，原則として各

メディアの配置位置を記す MediaEvent クラスのインスタンスを包含対象として指定することで，包含するメディアとその時間的な配置の組み合わせを表現することができる．例えば，CD や DVD，Blu-ray ディスク上の音声や映像，字幕情報などの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配置されている．これらのメディアデータをトラックやチャプターを単位として適切な AudioMedia/VideoMedia/DataMedia クラスのインスタンスとして定義したのち，MediaEvent クラスのインスタンスにそれぞれのメディアが配置されるべき時間位置（トラックやチャプターの開始位置）を指定し，CompositeMedia クラスにグループ化していくことで 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．

一方で，MediaEvent クラスを介することなく直接 Media クラスのインスタンスを包含するような記述も可能であり，この場合は時間的な配置を気にしない単なる包含関係を示すことができるほか，暗黙的に包含するすべてのメディアを再生区間の時刻 0（原点）の位置に配置することを表す．この表現方法では，同じメディアデータ内に映像と音声が同時に記録されている場合や複数のチャンネルをもつ音声を，映像・音声ごとや音声チャンネルごとに分離して個々のメディアインスタンスとして定義したのち，CompositeMedia クラスで包含関係を表現するなどの運用方法が考えられる．

第 4 章

SDM Ontology の応用—Web360^2

本研究では，SDM Ontology Version 2 の普及，および，コンテンツ制作者の視点からオントロジーの構造を考察する目的で，SDM Ontology Version 2 準拠のデータをもとに動作するアプリケーション Web360^2 の開発し，実験を通して評価をおこなった．Web360^2 [44] [45] は，インタラクティブに 360° 動画および立体音響を視聴体験できる Web アプリケーションである．具体的には Web ブラウザ上で，収録済みの 360° 動画を用いた自由視点視聴，および，視聴者の動作に追従し，個別の音源をインタラクティブに制御することができる立体音響の提示を目指す．そのため，立体音響の手法として，従来より広く用いられているチャンネルベースや 360° 動画との相性がよいとされる HOA/アンビソニックスベースではなく，個別の音源の制御が容易なオブジェクトベース方式を採用する．[44] [45] を発表した時点では，視聴体験可能なコンテンツの情報やメディアデータはすべてソースコードに埋め込まれていたが，現行バージョンでは，SDM Ontology Version 2 との連携によって LOD から情報を取得するよう実装されている．また，JavaScript のフレームワークである Vue.js[*1]を用いて UI (User Interface)/UX (User Experience) デザインを含めて刷新した．

Web360^2 は Github Pages[*2]にて公開している．2 章で紹介した関連システムと Web360^2 の比較概要を**表 4.1** にまとめる．現行バージョンでの視聴位置の移動は実験段階の β 機能であり，視聴位置の切り替えにはネットワーク環境に依存した遅延が起こる．

4.1 要求事項

先に述べた目的を踏まえて Web360^2 の要求事項として以下を想定した．

SDM Ontology Version 2 準拠のデータの活用：
SDM Ontology Version 2 の普及が前提にあるため，SDM Ontology Version 2 準拠のデータの活用は必須事項である．

[*1] https://vuejs.org/
[*2] https://sdm-wg.github.io/web360square-vue/

表 4.1. Web360^2 と関連システムとの比較.

システム	環境	360° 動画	視聴位置操作	視聴角度操作	各音源操作	ライブ配信
SDM360^2	Unity	✓	✓	✓	✓	×
LiVRation	Unity + HMD	✓	✓	✓	✓	✓
Inside Music	Web	×	✓	✓	✓	×
Web360^2	Web	✓	✓（β 機能）	✓	✓	×

Web アプリケーション：

専用のアプリケーションをインストールすることなく，Web ブラウザから直接利用可能な手軽さを目指す．SDM Ontology Version 2 の普及において，誰でも手軽に体験できることは望ましいと考える．

自由な映像音声の視聴：

視聴者が自由に視聴位置や視聴角度を決めることができ，位置・角度および視聴オブジェクトがもつ 3 次元位置情報をもとに自動で適切な映像・音声をレンダリングして提示できる．

インタラクティブ性：

ジャイロセンサ搭載デバイスでは，インタラクティブな視聴角度の変化を体験することができる．また，視聴者の操作によって，各音源のオブジェクトに個別にアクセスし ON/OFF の切り替えができる．これらの機能により，視聴者の興味関心に基づいた自由度の高い視聴体験が可能となる．

ストリーミング配信：

360° 動画を HTTP Live Streaming (HLS) でストリーミング配信する．ストリーミング配信に対応することで，再生開始までの待機時間の短縮や余計な通信を省くことが可能となる．

4.2　設計

Web360^2 のシステム設計概要を**図 4.1** に示す．視聴者の動作や音声オブジェクトへのタッチ操作を入力として受け付け，操作に応じた映像や音声をシステム内でリアルタイムにレンダリングして出力することでインタラクティブ性を実現する．映像・音声のメディアファイルはそれぞれ独立しており，システム内で同期をとりながら再生する．映像ファイルに 360° 動画，音声ファイルには AudioSprite[*3]によって各音源ファイルを 1 つにまとめた音源群が収録されている．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，再生の際のリクエスト数を減らすことができるため読み込み時間を短縮することができる．また，すべての音声が 1 つのファイルにまとまっていることで，音

[*3] https://github.com/tonistiigi/audiosprite

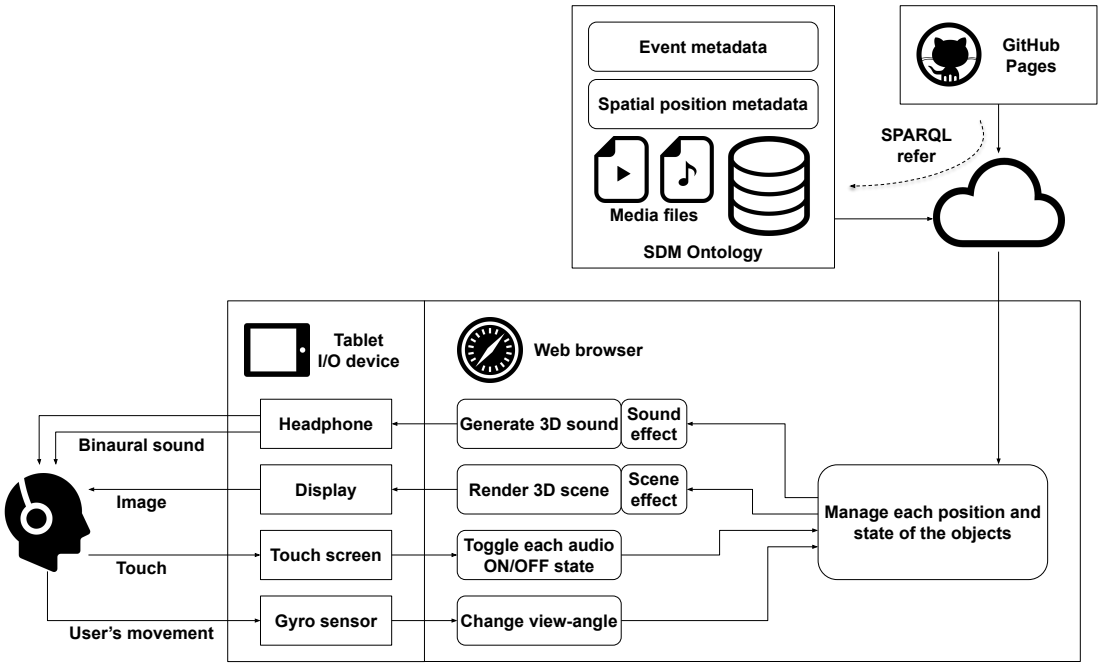


図 4.1. Web360^2 のシステム設計．ユーザのインタラクティブな操作に応じてリアルタイムに映像・音声をレンダリングする．視聴コンテンツの情報は SPARQL クエリを用いて RDF ストアに問い合わせて取得する．

声ごとの読み込み遅延を気にする必要がなくなり，音声間の同期がとりやすいことも利点として挙げることができる．

360° 動画は HLS でストリーミング配信され，視聴者を中心とする仮想的な球面内側に投影して再生される．また，音声オブジェクトのもつ 3 次元位置情報に基づいて，動画を投影する仮想的な球内部の各地点に音声可視化オブジェクトを描画し，各音声の強弱を表現したり，タッチ操作によって ON/OFF を切り替えることができる．視聴者の動作は端末に搭載されたジャイロセンサを通してシステムに入力され視聴角度が変更される．視聴角度と ON 状態にある各音声オブジェクトの位置関係から聴こえるべき音声を生成し出力する．

4.3 素材データ

Web360^2 の開発に用いた素材データは，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の収録データ，および，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 のジャズセッションの収録データである．いずれも SDM コンソーシアムの活動の一環で過去に収録したデータであり，[3] や [46] に，それぞれの収録環境や収録方法が記述されている．

Web360^2 で活用するために，収録当時に記録していた収録に関する環境や内容の情報をも

とに，SDM Ontology Version 2 準拠の RDF データファイルを，簡潔で可読性の高い Turtle 記法で作成・整理し，SDM コンソーシアムが運用する GraphDB[*4]で構築された RDF ストア[*5]にデータを登録した．例えば，**図 4.2** に示すように収録対象や収録機器，生成されるメディアファイルとその編集・加工の過程など，様々な情報を整理・分類する．**図 4.3** は各素材データ収録時の収録機器の配置を表し，このような収録に関する情報を整理して**図 4.4** に表す RDF データを作成および登録した．Web360² は，RDF ストアに Web API を通じて SPARQL クエリを発行することでアプリケーションの動作に必要なデータを取得し，活用することができる．

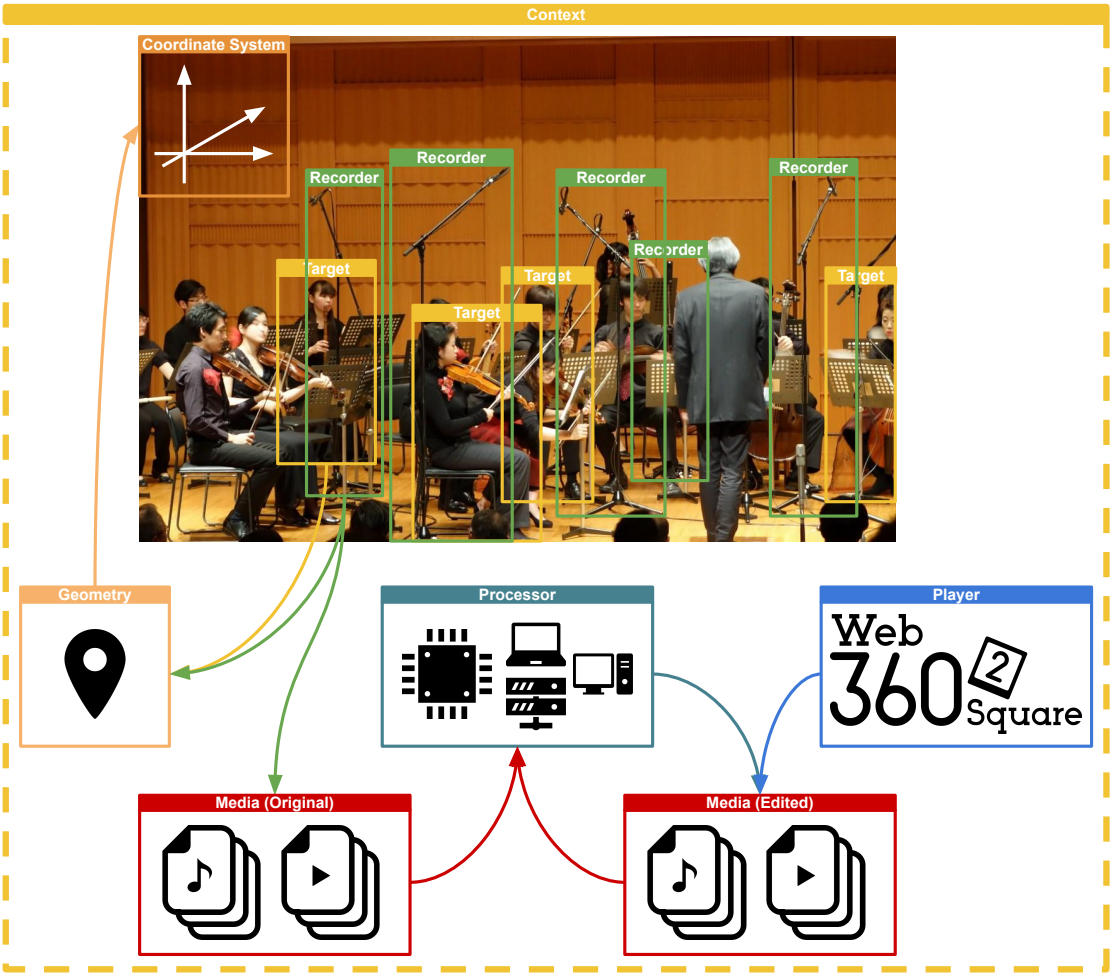


図 4.2. SDM Ontology Version 2 に準拠するデータ作成の概要．メディアの収録や編集・加工について，SDM Ontology Version 2 が定義するクラスやプロパティを用いて整理・分類し，RDF データを作成する．

[*4] http://graphdb.ontotext.com/

[*5] 本論文では，RDF のデータを保存・格納するトリプルストア，および，RDF のクエリ言語である SPARQL を用いてデータの問い合わせをする SPARQL エンドポイントの機能を有するサービスのことを指す．https://sdm.hongo.wide.ad.jp:7200/ で運用している．

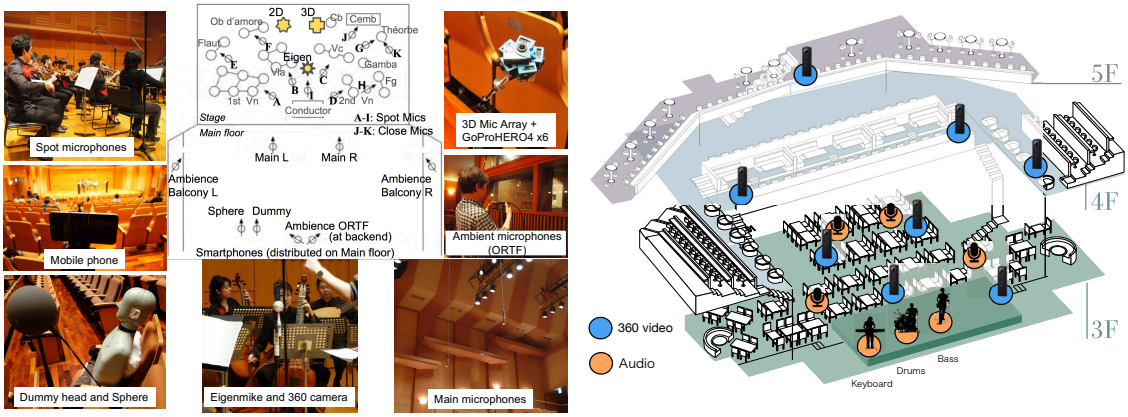


図 4.3. 慶應義塾大学コレギウム・ムジクム古楽アカデミーのコンサート（左），および，Musilogue Band のジャズセッション（右）の収録機器の配置を表す [3].

```
# prefix definition
@prefix sdm: <http://sdm.hongo.wide.ad.jp/resource/> .
@prefix sdmo: <http://sdm.hongo.wide.ad.jp/sdmo/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix geom: <http://data.ign.fr/def/geometrie#> .

# ...

# an instance of microphone coordinate
sdm:Kmc160110AudioRecorderGeometry1
    a sdmo:Geometry ;

    # coordinate values
    geom:coordX "3.05"^^xsd:double ;
    geom:coordY "1.0"^^xsd:double ;
    geom:coordZ "-7.33"^^xsd:double ;

    # link to a microphone instance
    sdmo:geometryOf sdm:Kmc160110AudioRecorder1 ;

    # link to a coordinate system instance
    sdmo:measuredBy sdm:Kmc160110CoordinateSystem1 .

# ...
```

図 4.4. 素材データ収録を SDM Ontology Version 2 に準拠するように記述した RDF データ．例として収録機器配置に関する記述の一部を抜粋している．データは Turtle 記法で記述されている．

4.4　実装

Web360² は，JavaScript のフレームワークである Vue.js をベースに，WebVR のフレームワークである A-Frame[*6]と Web 上で高度なオーディオ処理などを実行できる Web Audio API[*7]を用いて実装した．コンテンツ視聴部分の実装概要を**図 4.5** に示し，視聴体験中のスクリーンショットを**図 4.6** に示す．

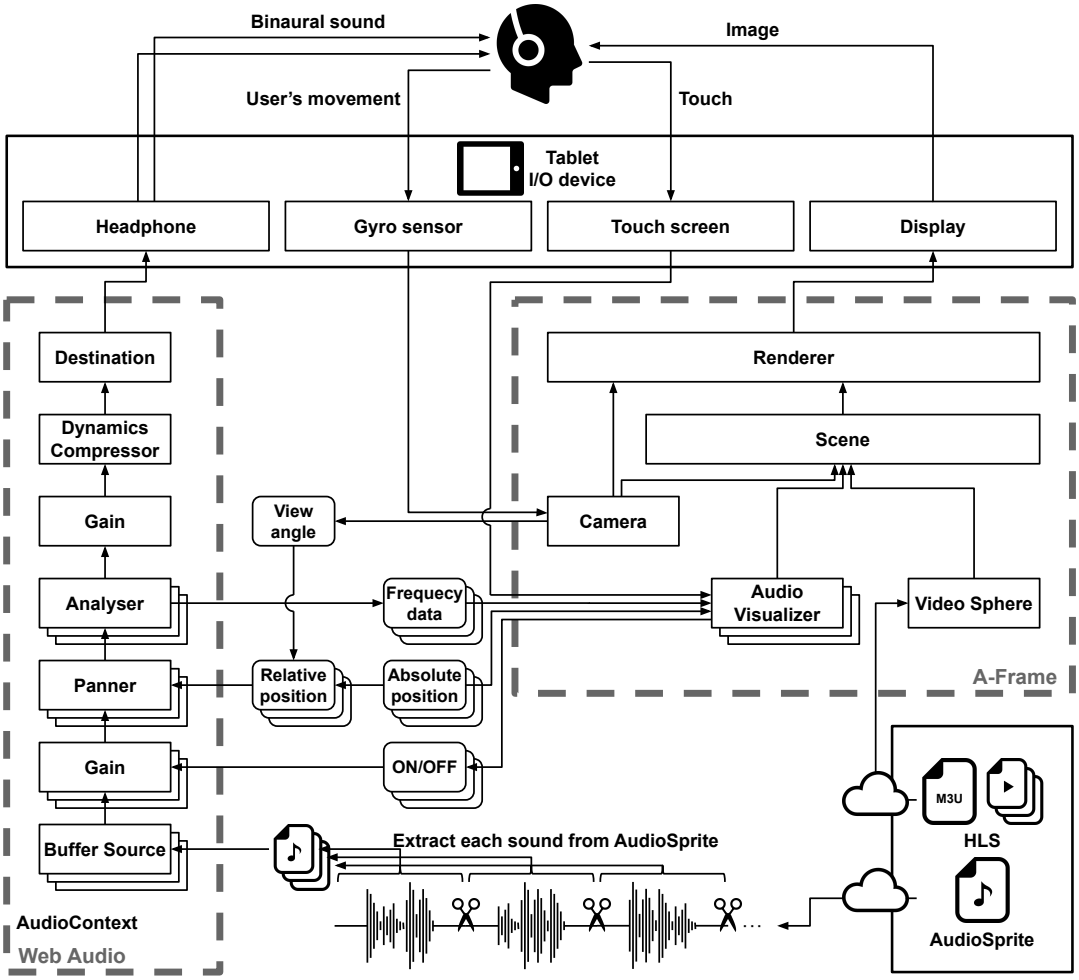


図 4.5. Web360² の実装概要．Web Audio API を用いた処理（左点線枠）と A-Frame を用いた処理（右点線枠）を連携して実装している．

[*6] https://aframe.io/

[*7] https://www.w3.org/TR/webaudio/

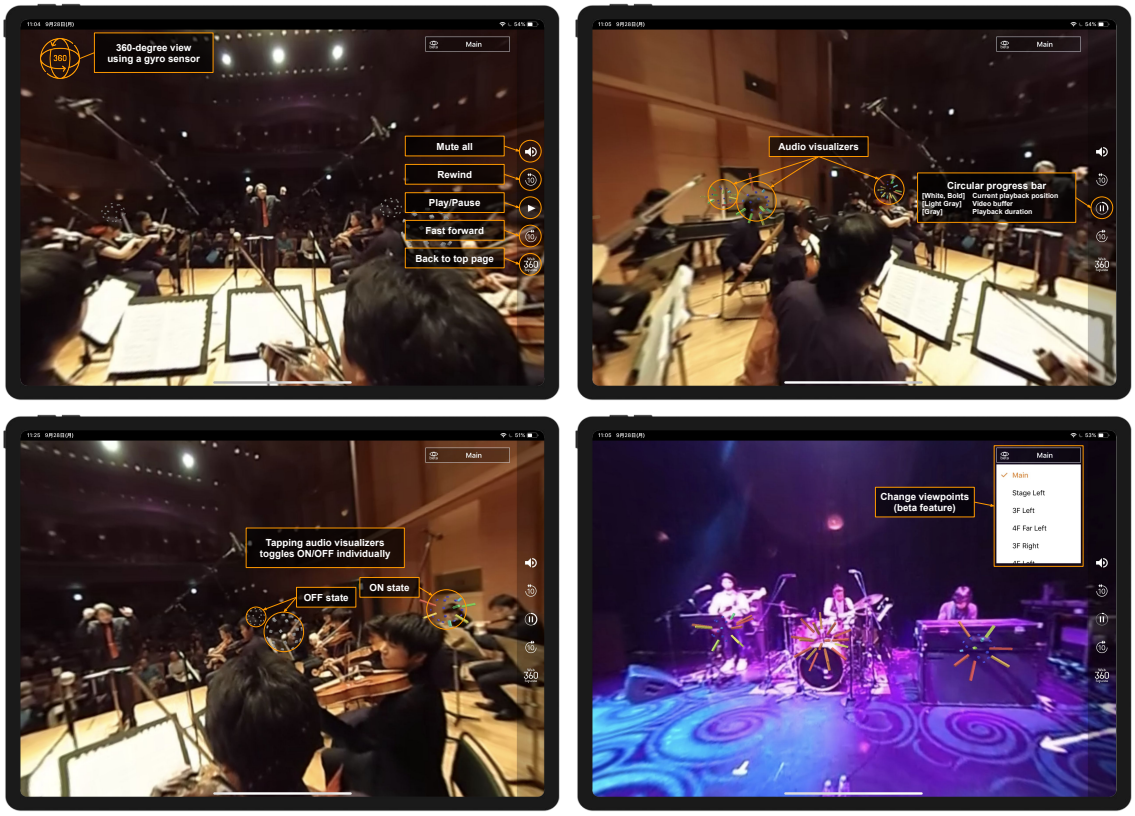


図 4.6. 視聴体験中の Web360^2 のスクリーンショット．画面右側にコントロールパネルがあり，再生や一時停止，早送り，巻き戻しなどの基本操作ができる．ジャイロセンサ搭載デバイスでは，動作にあわせて視聴角度が変動する．音声は個別管理されていて，タップ（クリック）操作によってそれぞれ独立に ON/OFF を切り替えることができる．視聴位置の変更操作は実験的な機能である．

4.4.1 SPARQL クエリによる動的なデータ取得

Web360^2 は，4.3 節で言及したように，収録に関する様々な情報をまとめた RDF データを格納する RDF ストアに SPARQL クエリを発行してアプリケーションの動作に必要なデータを取得する．現状，発行するクエリは以下の 2 種類である．

1. 視聴体験可能なイベントの取得
2. あるイベントに関する映像・音声メディアファイルや座標情報などの取得

視聴体験可能なイベントの取得

Web360^2 にアクセスすると，はじめに**図 4.7** のようなイベント選択画面が表示される．この画面に表示されるイベントの各情報は，アプリケーションから RDF ストアに向けて**図 4.8** の SPARQL クエリを発行し，**図 4.9** に表すような探索を経て取得される．



図 4.7. Web360² のイベント選択画面のスクリーンショット．イベントの情報は**図 4.8** の SPARQL クエリによって**図 4.9** のように RDF ストアから動的に取得する．

```
PREFIX sdm: <http://sdm.hongo.wide.ad.jp/resource/>
PREFIX sdmo: <http://sdm.hongo.wide.ad.jp/sdmo/>
PREFIX schema: <http://schema.org/>

SELECT DISTINCT ?event ?eventName ?eventDate ?eventPlaceName ?eventPlaceAddress WHERE {
  VALUES ?playerClass {sdmo:Player sdmo:DataPlayer sdmo:AudioPlayer sdmo:VideoPlayer sdmo:
      CompositePlayer}
  ?player
    a ?playerClass ;
    schema:name "Web360Square" ;
    sdmo:plays ?event .
  ?event
    a sdmo:SDMEvent ;
    schema:name ?eventName ;
    schema:startDate ?eventDate ;
    schema:contentLocation ?eventPlace .
  ?eventPlace
    a schema:Place ;
    schema:name ?eventPlaceName ;
    schema:address ?eventPlaceAddress .
}
```

図 4.8.　Web360² におけるイベント情報取得クエリ．**図 4.9** に示すように探索される．

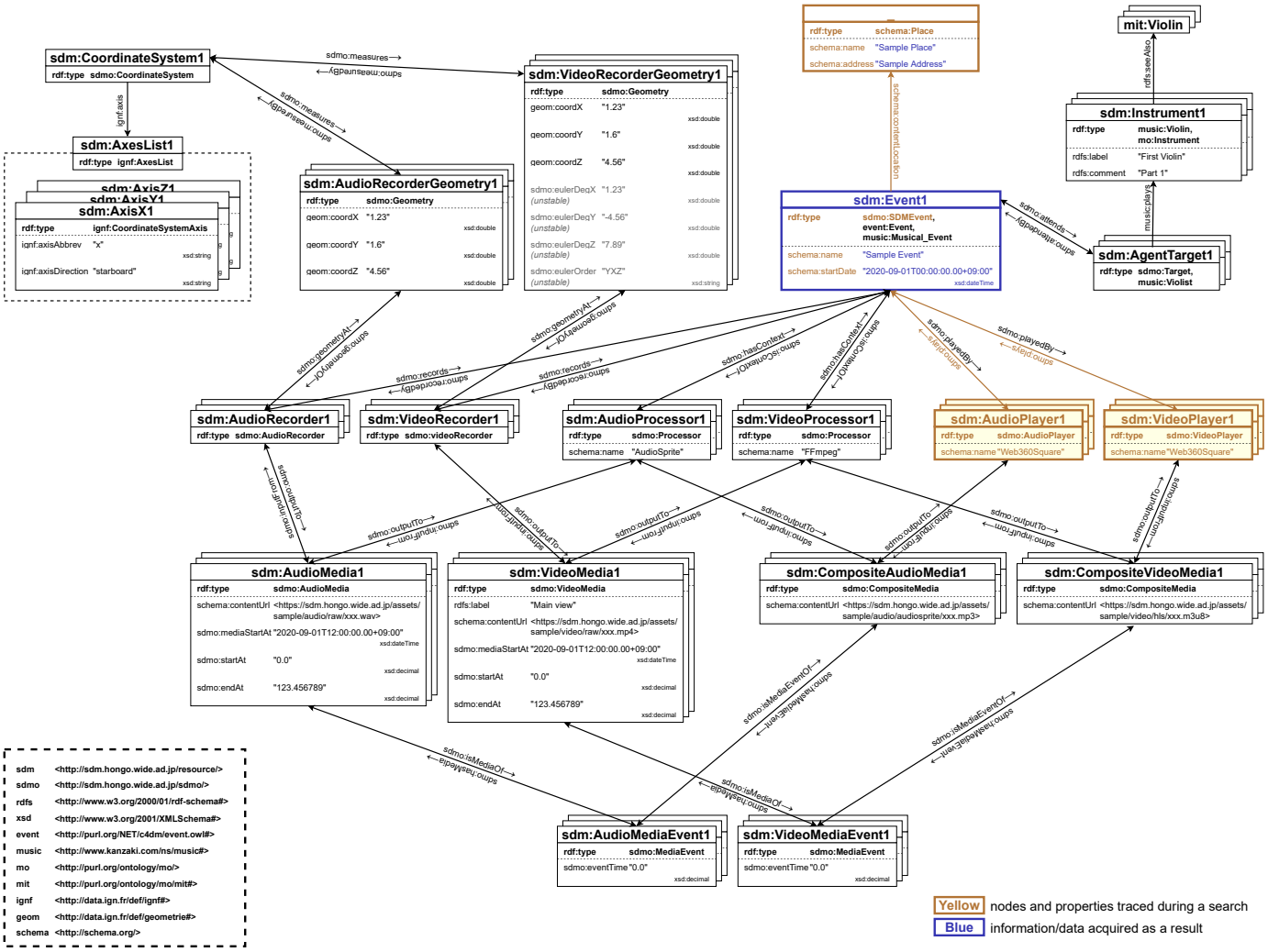


図 4.9. Web360^2 におけるイベント情報取得の概要．**図 4.8** のクエリが発行されると RDF データ内の Web360^2 を Player として指定する Player クラスインスタンス（右中段にある背景が薄黄色の 2 つのノード）を基点に探索する．黄色は探索中に辿ったノードやプロパティを表し，青色は探索結果として取得する情報を表す．

あるイベントに関する映像・音声メディアファイルや座標情報などの取得

イベント選択画面から 1 つのイベントを選択してタップあるいはクリックすると，**図 4.6** のような視聴画面に遷移する．視聴画面では視点に対して音声可視化オブジェクトがそれぞれ正しい配置でレンダリングされる必要があるため，360° 動画や各音声のメディア以外に，それらの座標情報が必要となる．また，映像・音声はそれぞれ独立したメディアファイルを用いるため映像と音声の同期処理が必要になる．同期処理自体の詳細は 4.4.5 項に記すが，同期のためにそれぞれのメディアファイルの長さの情報なども必要である．このように，視聴画面をレンダリングするためには選択されたあるイベントに対して様々な情報が必要であり，Web360^2 では，**図 4.10** の SPARQL クエリを用いて収集する．**図 4.11** に視聴情報取得の過程を示す．

```
PREFIX sdm: <http://sdm.hongo.wide.ad.jp/resource/>
PREFIX sdmo: <http://sdm.hongo.wide.ad.jp/sdmo/>
PREFIX geom: <http://data.ign.fr/def/geometrie#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>

SELECT DISTINCT ?playerClass ?contentUrl ?eventTime ?viewLabel ?startAt ?endAt ?x ?y ?z ?eulerDegX ?
    eulerDegY ?eulerDegZ ?eulerOrder WHERE {
  ?player
    a ?playerClass ;
    schema:name "Web360Square" ;
    sdmo:plays sdm:Billboard170126Event1 ;
    sdmo:inputFrom ?appMedia.
  VALUES ?appMediaClass {sdmo:Media sdmo:DataMedia sdmo:AudioMedia sdmo:VideoMedia sdmo:CompositeMedia
      }
  ?appMedia
    a ?appMediaClass ;
    schema:contentUrl ?contentUrl ;
    sdmo:inputFrom ?processor .
  VALUES ?plocessorClass {sdmo:Processor sdmo:Generator sdmo:Converter sdmo:Analyzer sdmo:
      CompositeProcessor}
  ?processor
    a ?plocessorClass ;
    sdmo:inputFrom ?originalMedia .
  OPTIONAL {
    ?appMedia
      sdmo:hasMediaEvent ?mediaEvent .
    ?mediaEvent
      a sdmo:MediaEvent ;
      sdmo:hasMedia ?originalMedia ;
      sdmo:eventTime ?eventTime .
  }
  VALUES ?originalMediaClass {sdmo:Media sdmo:DataMedia sdmo:AudioMedia sdmo:VideoMedia sdmo:
      CompositeMedia}
  ?originalMedia
    a ?originalMediaClass ;
    sdmo:inputFrom ?recorder ;
    sdmo:startAt ?startAt ;
    sdmo:endAt ?endAt .
  OPTIONAL {
    ?originalMedia
      rdfs:label ?viewLabel .
  }
  VALUES ?recorderClass {sdmo:Recorder sdmo:DataRecorder sdmo:AudioRecorder sdmo:VideoRecorder sdmo:
      CompositeRecorder}
  ?recorder
    a ?recorderClass ;
    sdmo:geometryAt ?geometry .
  ?geometry
    a sdmo:Geometry ;
    geom:coordX ?x ;
    geom:coordY ?y ;
    geom:coordZ ?z .
  OPTIONAL {
    ?geometry
      sdmo:eulerDegX ?eulerDegX ;
      sdmo:eulerDegY ?eulerDegY ;
      sdmo:eulerDegZ ?eulerDegZ ;
      sdmo:eulerOrder ?eulerOrder .
  }
}
```

図 4.10.　Web360² における視聴情報取得クエリ．**図 4.11** に示すように探索される．

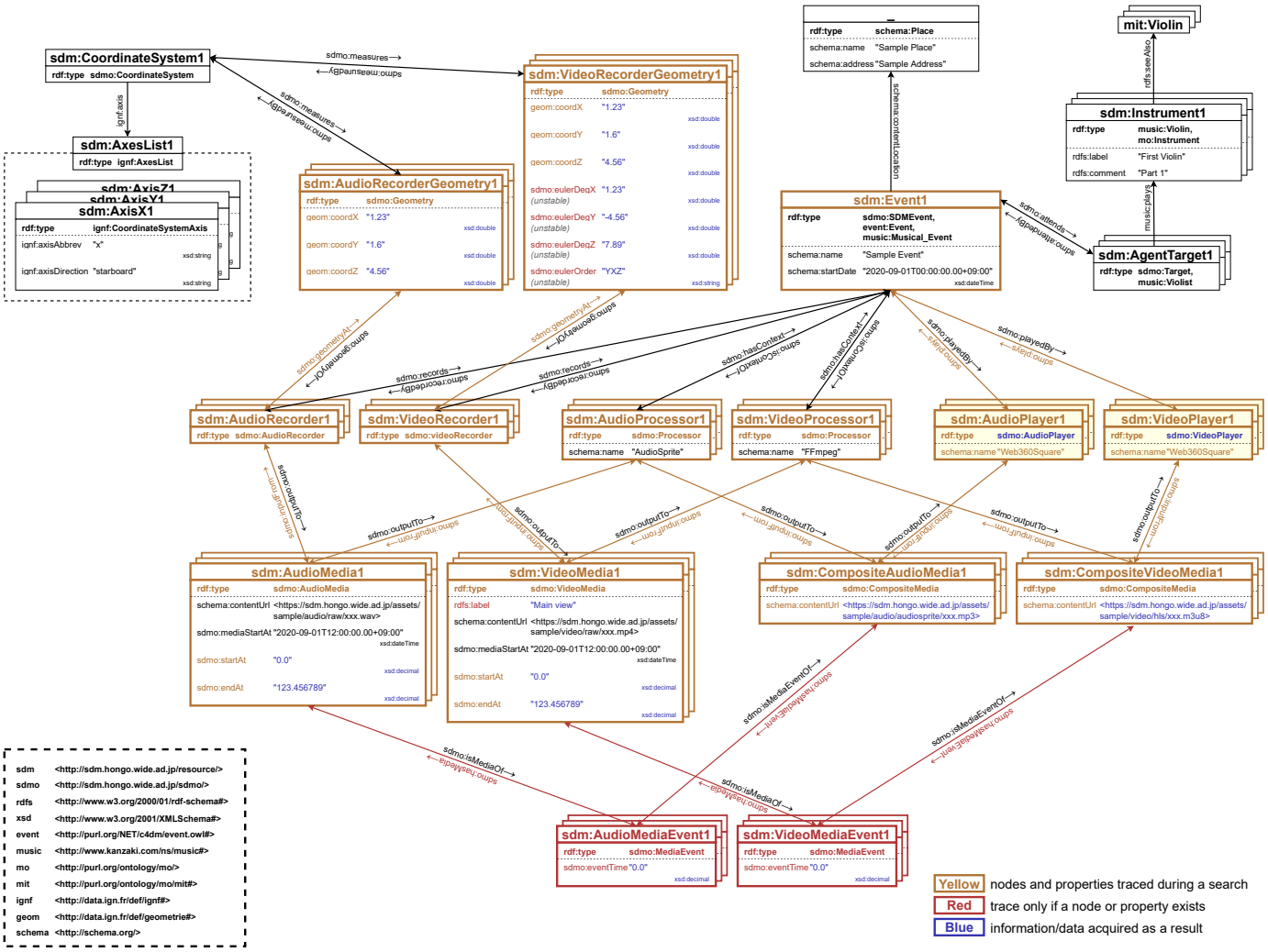


図 4.11. Web360^2 における視聴情報取得の概要．**図 4.10** のクエリが発行されると RDF データ内の Web360^2 を Player として指定する Player クラスインスタンス（右中段にある背景が薄黄色の 2 つのノード）を基点に探索する．黄色は探索中に辿ったノードやプロパティを表し，赤色は該当のノード・プロパティが存在するときにのみ辿ることを表す．また，青色は探索結果として取得する情報である．

4.4.2 A-Frame を用いた処理概要

A-Frame を用いて，**図 4.5** の右点線枠の手順で 360° 動画の投影や音声可視化オブジェクトの描画をおこなう．360° 動画は HLS でストリーミング配信され，A-Frame の `a-videosphere` オブジェクトを用いて，カメラの座標を基点に広がる球面内部に映像を投影する．HLS は Safari や Edge，一部のモバイル端末用ブラウザではネイティブサポートされているが，Chrome や Firefox，IE などサポートしていないブラウザも多く存在するため，

hls.js[*8]を用いて対応した．音声の可視化は Web Audio の Analyser ノードから提供される各音声の周波数領域データを用いて表現している．

また，視聴者のインタラクティブな操作は主に A-Frame が担う処理を介して入力される．視聴者の頭部動作は端末に搭載されたジャイロセンサに検知され，A-Frame のカメラオブジェクトに入力されることで視聴角度が変更される．タッチ操作は A-Frame でレンダリングしたオブジェクトにタッチすることで入力として認識され，各音声の ON/OFF 状態を変更することができる．

4.4.3　Web Audio API を用いた処理概要

図 4.5 の左点線枠に示すように，Web Audio の AudioContext 上で各ノードを鎖状に繋いでいき，視聴位置・角度と ON 状態の各音声オブジェクトから聴こえるべき音声をリアルタイムレンダリングする．はじめに各音源が 1 つにまとめられた AudioSprite のファイルデータから各音源を抽出し，Buffer Source ノードに入力する．次の Gain ノードでは，OFF 状態の音声のゲインを 0 にすることで音声の ON/OFF 状態を表現する．映像はある視聴位置に対して視聴角度と各オブジェクトの絶対座標に応じて，A-Frame が自動的に適切な相対位置を計算しながらレンダリングするが，Web Audio 上で音声オブジェクトは視聴角度の変動を自動追従できないため，カメラの視聴角度と絶対座標から相対座標を随時計算して Panner ノードに受け渡すことで音声の追従性を実現する．例えば，視聴角度に対して音声オブジェクトが右側に存在すれば右側で音が鳴っているような音声を生成することができる．また，各音声の周波数領域データは Analyser ノードで計算され，A-Frame の音声の可視化処理に用いる．以上の処理までは各音源ごとにされ，最後にマスターゲインとなる Gain ノードや Dynamics Compressor ノードでまとめられバイノーラル出力される．

4.4.4　音声の可視化

音声可視化オブジェクトは音声の周波数領域データに対応する棘状オブジェクトをもち，対応する周波数データの大きさによって長さや色を変化する．この棘状オブジェクトは [47] のアルゴリズムを参考に球面上に可能な限り均等に配置している．ただし，[47] では球面の極周辺における分布のずれの補正に関する言及があるが，配置に関してそこまでの精度を必要としないことから補正項の導入はおこなっていない．音声可視化オブジェクトがもつ棘の数は 32 としているため，**図 4.12** に示す対応する音声の周波数領域データを 32 個の値に変換する処理が必要である．

Web Audio の Analyser ノードでは，時間領域データや FFT（Fast Fourier Transform，高速フーリエ変換）による周波数領域データを取得することができ，`AnalyserNode.getByteFrequencyData()` メソッドを用いて周波数データを取得する．周波数データの取得は音声可視化オブジェクトを構成する A-Frame コンポーネントの `tick` ハ

[*8] `https://github.com/video-dev/hls.js/`

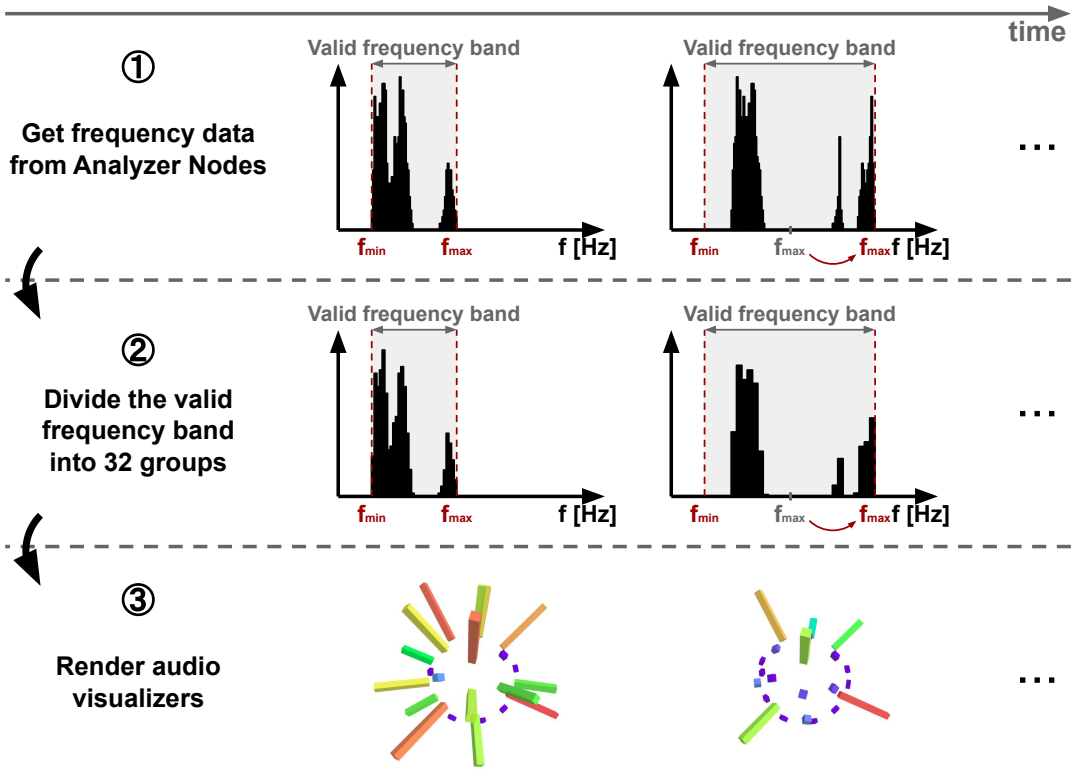


図 4.12. 音声を可視化する処理．1) 周波数データを取得して有効周波数帯を更新する．2) 有効周波数帯を可能な限り要素数が均等な 32 グループに分けて，平均値を求める．3) 32 個の平均値と音声可視化オブジェクトの棘を対応させてレンダリングする．以上，1) から 3) までの手順を毎秒 20 回程度のオーダーで繰り返しおこなう．

ンドラを用いて毎秒 20 回程度のオーダーで繰り返しおこなわれる．Web Audio のデフォルトのサンプリング周波数は OS やブラウザに依存する値で，一般には 44 100 Hz または 48 000 Hz であり，`AnalyserNode.getByteFrequencyData()` メソッドによってナイキスト周波数 22 050 Hz または 24 000 Hz 以下の周波数データが得られるが，極端な高周波数帯・低周波数帯にはあまりデータが存在せず，有効な値が含まれる周波数の範囲は限られていることが多い．そこで，一度でも有効な値が現れたことがある周波数の上限・下限をとり，この範囲を有効周波数帯とする．有効周波数帯は再生開始から周波数データが取得される度に更新される．有効周波数帯の中に含まれるデータを周波数の高さの順に可能な限り要素数が均等な 32 グループに分けて，グループ内で平均値を計算する．得られた 32 グループそれぞれの平均値は音声可視化オブジェクトの 32 本の棘の長さと色に変換されて描画される．周波数の高さ順と音声可視化オブジェクトの棘の配置順を対応させて描画してみたところ，近接する周波数グループ同士に相関があるために部分的に棘が盛り上がるような不格好な形状となったため，周波数の 32 グループと音声可視化オブジェクトの 32 本の棘の配置の対応関係はランダムに定まるようにした．また，OFF 状態の音声可視化オブジェクトの棘の色はグレーであり，ON

状態では平均値が大きくなるにつれて青，水色，緑，黄，オレンジ，赤と変化していく．

負荷軽減

音源 1 つにつき 32 本の棘状オブジェクトをもつ音声可視化オブジェクトを生成し，さらにそれらがすべて独立に長さや色を変化させるため，デバイスがおこなうべき計算処理は非常に大きくなる．使用するデバイスの計算能力によっては，レンダリングのカクつきが起こり QoE (Quality of Experience) の低下につながる可能性が高い．Web360^2 では，以下の 3 つの負荷軽減のための対策を導入している．

1. 音声可視化処理の FPS (Frames Per Second) の低減
   A-Frame コンポーネントの `tick` ハンドラはデフォルトでは毎秒 60 回から 120 回程度のオーダーで実行されるが，音声可視化オブジェクトにおける `tick` ハンドラは毎秒 20 回程度のオーダーで実行されるように実行回数を低減している．様々なオーダーを試してみたが，音声可視化オブジェクトが十分滑らかに更新されるためには毎秒 20 回程度実行される必要があると主観的に判断した．
2. 死角にある棘状オブジェクトの非表示
   音声可視化オブジェクトがもつ棘状オブジェクトは中心から全方位に均等配置で伸びているため，視聴位置の反対方向に伸びている棘は常に視聴者の死角にある．死角にある棘状オブジェクトを**図 4.13** のように非表示にすることで，32 本の棘のうちレンダリングすべき棘はおおよそ 20 本程度になり計算処理を軽減できる．
3. 音声可視化オブジェクトの球体セグメント数の削減
   音声可視化オブジェクトは中心に透明な球体をもち，この球体にタップ（クリック）による音声の ON/OFF 切り替え操作を紐付けている．球体自体は透明であるため球体に滑らかさは不要であり，セグメント数を削減することでポリゴン構造を荒くすることができる．ただし，あまりに構造が荒いとタップ（クリック）の反応範囲に違和感が現れるため，A-Frame の `a-sphere` オブジェクトのデフォルトセグメント数の半数に設定している．

4.4.5　映像と音声の同期処理

映像・音声のメディアファイルはそれぞれ独立しているため同期をとりながら再生する必要がある．映像の再生時間は JavaScript の `HTMLMediaElement.currentTime` プロパティによって取得・設定することができる．一方，音声については Web Audio API に読み取り専用の `AudioContext.currentTime` プロパティが用意されているが，これは音声の再生時間ではなく Web Audio の AudioContext が生成されてからの経過時間を示す．音声が停止中であっても `AudioContext.currentTime` プロパティの値は単調に増加し続けるため，システム内では音声の再生・停止を管理する Web Audio API の `AudioBufferSourceNode.start()` メソッドおよび `AudioBufferSourceNode.stop()` メ

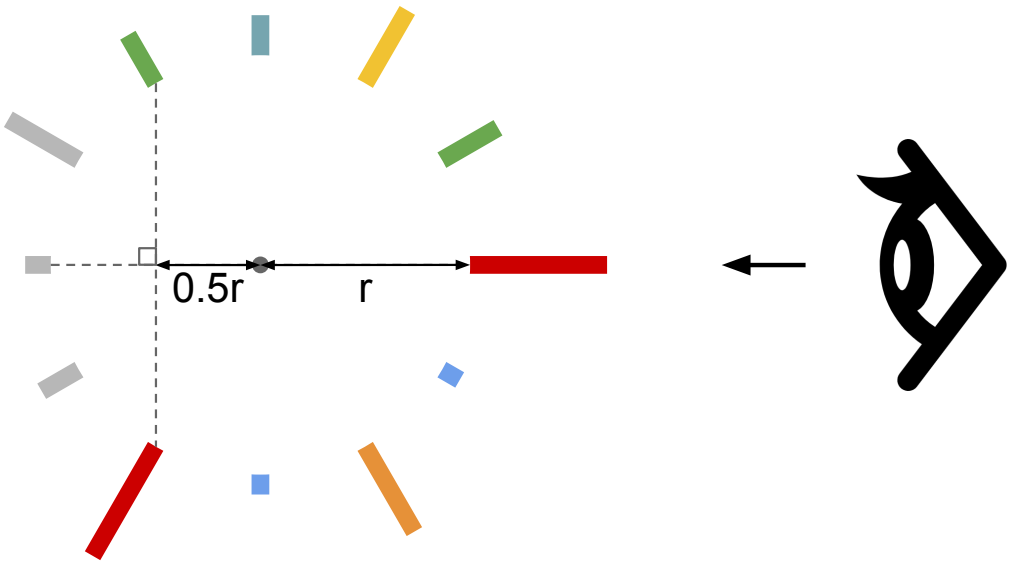


図 4.13. 死角にある棘状オブジェクトを非表示にする処理の概要．視聴位置から音声可視化オブジェクトの中心へと向かうベクトルを法線ベクトルとするような平面を音声可視化オブジェクトの中心から半径の半分だけ後方に置く．棘状オブジェクトの根本がこの平面よりも後方にある場合，死角にあると判定して非表示にする．

ソッドが呼び出される際に `AudioContext.currentTime` プロパティの値を記録するなどして音声の再生時間を算出している．

再生開始時やループ再生により先頭に戻った際には算出された音声の再生時間をもとに映像の再生時間を強制同期させる．また，視点を管理する A-Frame のカメラオブジェクトがもつ `tick` ハンドラによって毎秒 60 回から 120 回程度のオーダーで現在の音声の再生時間を算出し，映像の再生時間との差分を管理する．差分値に対して閾値を設定し，閾値を超えたときに映像の再生時間を音声の再生時間に強制的に同期させる対応は可能ではあるが，同期処理が入るたびに映像再生の滑らかさが失われ，QoE を損なってしまう．そこで，現行バージョンでは差分の閾値を 0.1 秒として，閾値よりも映像が遅れたら映像の再生速度を 2 倍にし，閾値よりも映像が先行したら映像の再生速度を 0.5 倍にすることで，映像と音声の再生時間のずれが徐々に閾値以内に収束するような比較的制約のゆるい同期をとっている．再生速度の変調は多少の違和感を生じてしまうが，映像再生の滑らかさを保つことができる．映像の再生速度は JavaScript の `HTMLMediaElement.playbackRate` プロパティによって設定する．ここで，映像と音声が，映像の速度変調によるゆるい同期では時間を要するほど大きくずれてしまった場合，視聴者に与える違和感も大きくなってしまうため，やむを得ず映像の再生時間を音声の再生時間に強制的に同期させる．強制同期を発動する再生時間差分の閾値は 1 秒としている．

4.4.6　視聴位置の移動

視聴位置の移動は実験段階の β 機能である．現行バージョンでは，再生中の視聴位置以外の 360° 動画は読み込まず，視聴位置移動の操作があってから動画の読み込みが始まるためネットワーク環境に依存した遅延が起こる．この問題はすべての視聴位置の動画を事前に読み込んでおくことで解決することができるが，視点位置の数に比例して通信量が大きくなってしまう．このように視点移動の滑らかさと通信量はトレードオフである．

4.4.1 項でも述べた通り，音声可視化オブジェクトをはじめとする視聴オブジェクトを正しい位置にレンダリングするために，映像・音声の座標情報などを取得している．4.4.2 項で触れた 360° 動画を内部に投影する A-Frame の `a-videosphere` オブジェクトは常に視点を管理するカメラオブジェクトを中心とする性質がある．視聴位置の移動の実装の方法として 2 つの方針が考えられる．

1. カメラオブジェクトを映像の座標まで移動して，音声可視化オブジェクトは取得した座標情報通り配置する
2. カメラオブジェクトを固定して，音声可視化オブジェクトを映像の座標情報を原点とする相対座標で配置する

前者はカメラオブジェクトのみを移動すればよく単純であるが，映像の位置情報だけでなく姿勢情報も含めてカメラオブジェクトに適用する必要がある．カメラオブジェクトの姿勢が過度に傾いている場合，ジャイロセンサによる視聴角度操作が滑らかにおこなわれないことがあったため，Web360^2 では，カメラオブジェクトを固定する後者の音声可視化オブジェクトの相対座標配置を採用している．

第 5 章

評価・普及活動

5.1 SDM Ontology の評価・普及活動

第 3 章で紹介した SDM Ontology Version 2 について評価をおこないたい．5.1.1 項では，既存の指標値やオンラインサービスを用いた評価をまとめる．しかしながら，オントロジーには製作者の主観的な思想が色濃く現れる場合が多く，指標値やオントロジー構造のみをもとに，その良し悪しを正確に測ることは一般には難しいとされる．オントロジーが世間に周知され，利用されるようになってはじめて，そのオントロジーが「良い」かどうかわかってくる面もあるため，オントロジーの普及活動は非常に重要であると考える．本研究では，SDM Ontology Version 2 の普及を目指していくつかの活動をおこなっている．普及活動の詳細について 5.1.2 項で述べる．

5.1.1 評価

Fernández ら [48] は，オントロジーの評価測定のための 12 の指標を定義しており，Turchet らは Internet of Musical Things Ontology [24] において，これらの指標の一部を修正しつつオントロジー評価に適用している．本節では，SDM Ontology に対して Turchet らが用いた以下の指標を適用する．ただし，いずれの指標についてもオントロジーを評価するための有意な統計的意味をもっているとの報告はされておらず，あくまで参考としての値であることに注意されたい．

a) 知識の網羅性と人気度に関する指標（表 5.1）

- **クラス数：**
  対象のオントロジーに含まれるクラスの数.
- **プロパティ数：**
  対象のオントロジーに含まれるプロパティの数.
- **個体数：**
  対象のオントロジーに含まれる個体の数.

表 5.1. 知識の網羅性と人気度に関する指標．Fernández ら [48] が定義した指標をもとに，Turchet らが Internet of Musical Things Ontology [24] に適用した指標値を用いる．

指標	値
クラス数	30
プロパティ数	31
- オブジェクトプロパティ	23
- データ型プロパティ	8
個体数	0
外部オントロジーに利用されている数	0
外部オントロジーを利用している数：	
- 外部オントロジーの利用数	3
- 外部クラスの割合	10.0%
- 外部プロパティの割合	16.1%

- **外部オントロジーに利用されている数：**
  対象のオントロジーが外部のオントロジーに利用されている数．一般に成熟したオントロジーほど，この指標の値は大きくなるが，SDM Ontology は構造が確立し，外部に向けて公開してからの期間が短いため指標値は小さい．
- **外部オントロジーを利用している数：**
  対象のオントロジーが外部のオントロジーを利用している数．この指標は他の知識体系との相互運用性や対象のオントロジーの新規性を測定する．

b) データ構造に関する指標

- **最小のトリプル数：**
  対象のオントロジーがあるデータを記述するために必要なトリプルの最小数．この指標値はデータを使用する際のリソースの使用量やパフォーマンスに関係する場合がある．付録 A に示すような最小限の収録形態を記述するためには 60 程度のトリプルが必要である．
- **最大のトリプル数：**
  対象のオントロジーがあるデータを記述するために使用可能なトリプルの最大数．この指標値は SDM Ontology によって制限されるものではない．

また，SDM Ontology について OOPS! (OntOlogy Pitfall Scanner!) [49] による評価をおこなった．OOPS!は既存の文献に基づいてオントロジーの欠陥につながりうる一般的な危険性を検出する RESTful Web サービスである．SDM Ontology では，いくつかの重要な (Important)，または，細かな (Minor) 危険性が検出されているが，極めて重大な (Critical) 危険性は検出されていない．検出された危険性を以下にまとめる．

Important pitfalls

1. **要素間における互いに素な関係が定義されていない：**
   同時に所属し得ない互いに素なクラスや，同時に持つことができない互いに素なプロパティが存在する場合があり，矛盾した関係の記述を防止するためにはオントロジー側で互いに素な関係を定義する必要がある．SDM Ontology にも互いに素な要素関係は存在するが，細かく議論できていないため定義していない．今後，議論を重ねて定義していくべきである．
2. **プロパティの主語または目的語が定義されていない：**
   この警告はすべて外部オントロジーから参照したプロパティについて発生している．外部オントロジーのプロパティについては元のオントロジー側で主語・目的語を含め関係性が定義されているはずであり，SDM Ontology 内で再記述する必要はないため，この警告は無視している．
3. **再帰的定義の使用：**
   自分自身を用いた再帰的な定義をすると不都合が起こる場合があるため警告が発生しているが，SDM Ontology（特に Version 2）において再帰性はオントロジーの特性上必要であるため，この警告は無視している．

Minor pitfalls

1. **独立なオントロジー要素の存在：**
   他のクラスやプロパティとの関係性が定義されていない独立な要素に対する警告であり，親クラスとして参照した外部オントロジーのクラス要素に対して発生している．SDM Ontology 内で外部オントロジーのクラスがもつ細かな関係性を再記述する必要はないため，この警告は無視している．
2. **逆関係が定義されていない：**
   オブジェクトプロパティは一般に順関係・逆関係を対称的に定義することがよいとされるため，逆関係が明示的に定義されていないと，この警告が発生する．外部オントロジーから参照したプロパティについて発生しており，SDM Ontology 内で再記述する必要はないため，この警告は無視している．
3. **ラベル・注釈が存在しない：**
   人間にも読みやすいオントロジーを作成するために，各要素にはラベル・注釈のプロパティをつけることが推奨されている．この警告はすべて外部オントロジーから参照したプロパティについて発生しており，SDM Ontology 内で再記述する必要はないため無視している．
4. **ラベル・注釈の誤用：**
   人間が読むためのラベル・注釈のプロパティに誤用があった場合に発生する．SDM Ontology では不要なラベルづけをおこなっている要素において，この警告が発生しているが，他の要素との対称性を考えて無視している．

5.1.2　普及活動

SDM Ontology Version 2 の普及を目的として以下の活動をおこなった.

Web360^2 の開発

第 4 章で述べた通り，Web360^2 は SDM Ontology Version 2 準拠のオープンデータを用いて動作するインタラクティブ Web アプリケーションである．オントロジーと連携したコンテンツ・アプリケーションの存在は，第三者のオントロジーへの興味・関心を得るために大いに役立つものであると期待している．また，オントロジー開発と並行してアプリケーション開発をおこなうことで，オントロジー構造に対するコンテンツ・アプリケーション制作者側の要求を知ることができ，SDM Ontology Version 2 の構造を考察する上で非常に有益であった.

Web ドキュメントの公開

SDM Ontology の開発にはオントロジー開発支援のためのソフトウェア Protégé [50] を用いており，OWL (Web Ontology Language)[*1]というオントロジー記述言語のファイルを生成する．OWL ファイルは人間が読みやすい形式にはなっていないため，オントロジー普及のためには人間に読みやすい形式に変換して公開するとよい．Garijo らが開発した WIDOCO (WIzard for DOCumenting Ontologies) [5] は，オントロジーの各要素に定義したラベルや注釈のプロパティをもとに OWL ファイルを半自動的に Web ドキュメントに変換する．SDM Ontology Version 2 の語彙体系は，WIDOCO を利用して Web ドキュメント[*2]として公開されている.

LOD チャレンジ 2020 への応募

LOD チャレンジ[*3]は LOD の普及促進を目的に 2011 年から毎年開催されているイベントであり，LOD チャレンジ 2020 ではアイデア部門，アプリケーション部門，データセット部門，データ分析・可視化部門，基盤技術部門の 5 つの部門について，2020 年 7 月 1 日から 10 月 18 日まで作品募集しており，本研究からは LOD に興味・関心のある人々を中心に我々の活動を周知してもらう目的で，アプリケーション部門に Web360^2，データセット部門に SDM Ontology Version 2 準拠のオープンデータを提出し，データセット部門において最優秀賞およびスポンサー賞を受賞した.

[*1] https://www.w3.org/OWL/
[*2] https://sdm.wide.ad.jp/sdmo/
[*3] https://2020.lodc.jp/

5.2 Web360^2 の評価

Web360^2 については，アンケート調査による主観評価を二度おこなった．一度目は 2019 年 6 月 12 日から 6 月 14 日までの 3 日間にわたって開催された Interop Tokyo 2019 の SDM コンソーシアムブースにおけるデモンストレーション展示の一環として対面調査をおこなった．当時の Web360^2 には SDM Ontology との連携機能が実装されておらず，三次元映像・音声をインタラクティブに視聴体験するための最低限の機能のみをもたせたアプリケーションであった．二度目は一通りの機能を実装し終えた状態の Web360^2 に対して，2020 年 12 月中旬から 2021 年 1 月中旬までの期間に研究団体への告知や SNS などを用いてアンケート回答の協力を仰ぎ，Web ベースのアンケート回答フォームを用いてオンラインで実施した．

また，Web ブラウザや WebVR のフレームワークである A-Frame の開発用機能を用いて，Web360^2 を動作させるための要求性能を簡単に評価する．

5.2.1 Interop Tokyo 2019 での対面調査

調査概要

2019 年 6 月 12 日から 6 月 14 日までの 3 日間にわたって開催された Interop Tokyo 2019 の SDM コンソーシアムブースでデモンストレーション展示をおこなった際に，Web360^2 を体験した来場者にアンケートを回答してもらい，調査を実施した．体験では，2 台のヘッドホン (Sony WH-1000XM2) と 2 台のタブレット (Apple iPad Pro 12.9-inch (2018), Apple iPad Pro 11-inch (2019)) をそれぞれ有線接続して，Wi-Fi で Web360^2 にアクセスした．

はじめにアプリケーション概要と操作方法を実際に操作する様子を見せながら説明し，その後，体験者に自由に操作してもらった．体験終了後，アンケートの回答をお願いしてデータを取得した．

視聴可能なコンテンツは 4.3 節で述べたコンサートとジャズセッションの 2 つがあり，これらのコンテンツはそれぞれ異なる方法で収録したため，コンサートの収録データでは各音源に複数の楽器の音色が混ざって収録されているが，ジャズセッションの収録データでは楽器ごとの個別の音色を明確に聞き取りやすく収録されている．Interop Tokyo 2019 における対面調査では体験者に Web360^2 の機能をわかりやすく説明するために，個別の楽器の音がより聞き取りやすいジャズセッションのコンテンツを優先して操作説明をおこなった．

被験者

被験者は Web360^2 を体験した Interop Tokyo 2019 の来場者であり，男性 81 人，女性 9 人，無回答 3 人の合計 93 人でおこなった．被験者の年齢構成は 10 代 1 人，20 代 24 人，30 代 26 人，40 代 18 人，50 代 17 人，60 代以上 6 人，無回答 1 人である．Interop Tokyo 2019 には様々な企業から来場しており，被験者はビジネス寄りの視点をもつ人が多いと考えられる．

質問項目

アンケートの質問項目は，以下に示す設問 Q1 から Q6 までの 6 つであり，それぞれ 1 から 7 までの 7 段階のリッカート尺度を用いて回答してもらった．それぞれの回答について，最低評価を 1，中間を 4，最高評価を 7 とする目安を記載した．

Q1 音響は映像の方角と比べて正しい位置で鳴っているように聴こえましたか？
Q2 映像が動いたとき，音響も追従して動いたと感じられましたか？
Q3 音響の立体感は感じられましたか？
Q4 インタラクティブな視聴体験の操作は簡単にできましたか？
Q5 音量可視化による音響オブジェクトの ON/OFF 操作は直感的でしたか？
Q6 音響オブジェクトの ON/OFF 操作によって個別の楽器の音色を聞くことができましたか？

設問 Q1 および Q2 は映像と音響の組み合わせが違和感なく一体として知覚できるかを問う設問である．設問 Q1 は映像上の音響オブジェクトの位置と出力される音響の鳴っている方角・位置の知覚に差異がないかを問い，設問 Q2 は頭部動作による視聴角度の動きに対する音響の追従性を問う．設問 Q3 はシステム内で合成された音響に立体感を感じるかを問う設問である．また，設問 Q4 および Q5 はタブレットをもった状態での頭部動作やタッチ動作によるインタラクティブな視聴体験についての設問であり，設問 Q4 は全般的な操作の容易さを問い，設問 Q5 は音響オブジェクトの可視化と ON/OFF 操作について問う．設問 Q6 は各音響の ON/OFF 操作を通して，全体の音響を分解し個別の楽器の音色がを聞くことができたかを問う．

アンケートの末尾に「感想・要望・その他」という自由記述欄を用意して，Web360^2 の体験に関するコメントを得た．

評価結果

アンケートの調査結果を**図 5.1** に示す．グラフの横軸は 1 から 7 までの 7 段階の回答の比率を表し，それぞれ合計すると 100% になる．中間評価を示す 4 の回答比率の中間を横軸の原点 0% に置き，正の方向に高評価を示す 5, 6, 7 の回答比率を，負の方向に低評価を示す 3, 2, 1 の回答比率を伸ばす．つまり，高評価 (5, 6, 7) の回答が多いほどグラフは正の方向に偏り，逆に低評価 (3, 2, 1) の回答が多いほどグラフは負の方向に偏っていく．グラフ縦軸は前述の各設問 (Q1–Q6) であり，それぞれ末尾の括弧内に有効回答数を示す．

すべての設問に対して概ね高評価を得ることができ，いずれも最高評価の 7 と次点評価の 6 の回答割合のみで半数を上回った．特に後半の 3 つの設問 (Q4–Q6) については，最高評価の 7 のみで半数を超えている．一方で，音響の立体感を問う設問 Q3 については，評価が分散しており，あまり実感できなかったという評価（低評価の 3, 2, 1）も 8.7% とほかの設問に比べて多い結果となった．音響の立体感を構成する要素は音像の方角や奥行き，音源の臨場感，空間の臨場感など複数あると考えられるが，設問 Q3 は表現が曖昧であり，質問がどの要素を問

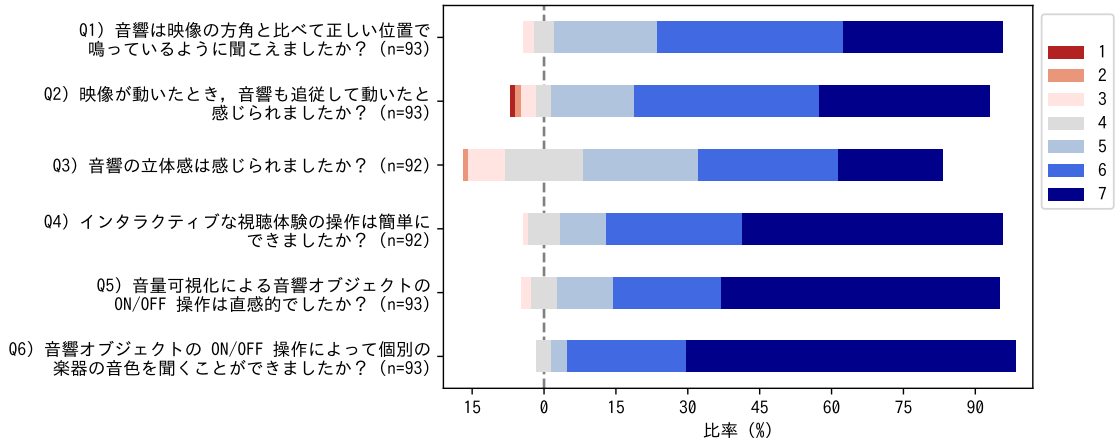


図 5.1. Interop Tokyo 2019 での対面調査におけるアンケートの回答比率．被験者には Q1–Q6 までの 6 つの質問に対して最低評価を 1，中間を 4，最高評価を 7 とする 7 段階のリッカート尺度を用いて回答してもらった．中間評価を示す 4 の回答比率の中心を横軸の原点 0% に置き，正の方向に高評価を示す 5, 6, 7 の回答比率を，負の方向に低評価を示す 3, 2, 1 の回答比率を伸ばしている．

うているのか，被験者がどの要素について応えているのか，明確にわからないという反省点が挙げられる．また，調査概要でも言及したように，視聴体験の説明には個別の音色が聞き取りやすい特性をもつジャズセッションのコンテンツを優先して用いたためか設問 Q6 では非常に高評価を得ているが，コンサートのコンテンツを用いていた場合には結果が大きく変わってくることが予想される．

「感想・要望・その他」の自由記述欄では各設問の評価を補足するようなコメントとして，直感的な操作性や個別の音響を消す機能，音響の追従によりその場にいるような没入感が得られたことへの好感触な感想がある一方で，360° で音響が鳴っている様子があまり実感できないという意見もあった．また，仮想空間内の移動への要望があったが，現行バージョンでは複数の 360° 動画を用いて視聴位置を移動する実験的な機能を実装している．

5.2.2 オンライン調査

調査概要

2020 年 12 月中旬から 2021 年 1 月中旬までの期間に研究団体のオンライン会合や研究発表の場，SNS などを用いて広報し，Web ベースのアンケート回答フォームを用いてオンライン調査を実施した．被験者には，テキストで送った手順や注意点に従って各自のデバイスやネットワーク環境のもと Web360^2 にアクセスしてもらい，視聴体験後に Web アンケートに回答してもらった．

被験者ごとに視聴環境が異なることを踏まえて，デバイス名や OS の種類，使用した Web ブラウザの種類，ネットワーク環境などの情報，さらに，ジャイロセンサやヘッドホンの有無などもあわせて入力してもらった．また，収録方法に差異があるためコンテンツによって，ア

ンケートの回答結果に影響が出ることを考慮して視聴したコンテンツの情報も取得した.

被験者

被験者は男性 28 人，女性 3 人の合計 31 人であり，年齢構成は 10 代 1 人，20 代 10 人，30 代 2 人，40 代 8 人，50 代 7 人，60 代以上 3 人である．研究団体中心に広報した経緯を踏まえて，情報技術に精通している人が多い.

質問項目

アンケートの質問項目は，以下に示す設問 Q1 から Q6 までの 6 つであり，Interop Tokyo 2019 での対面調査（5.2.1 項）の設問を一部改変して問うている．それぞれの設問に対して，最低評価を 1，中間を 4，最高評価を 7 とする 1 から 7 までの 7 段階のリッカート尺度を用いて回答してもらった.

Q1 音響は映像の方向と比べて正しく聞こえましたか？
Q2 音響は映像の奥行き（距離感）と比べて正しく聞こえましたか？
Q3 映像が動いたとき、音響も追従して動いたと感じられましたか？
Q4 インタラクティブな視聴体験の操作は簡単でわかりやすいものでしたか？
Q5 音響オブジェクトの ON/OFF 操作は直感的でしたか？
Q6 音響オブジェクトの ON/OFF 操作によって個別の楽器の音色を聞くことはできましたか？

設問 Q1，Q2 については，Interop Tokyo 2019 で実施した調査の反省を踏まえて，音響の立体感を方向と奥行きに分けて問うている．設問 Q3 から Q6 は表現を少し変更した部分があるものの，以前の設問内容と同じである.

この他に，対面調査とは異なり被験者ごとの視聴体験環境の差異が大きいため以下の環境情報を取得した.

- デバイス名
- OS の種類・バージョン
- Web ブラウザの種類・バージョン
- ネットワーク環境
- ジャイロセンサの使用
- ヘッドホンの使用
- 視聴体験したコンテンツ

また，アンケートの末尾に「感想・要望・気になったこと」,「不具合報告」の自由記述欄を用意した.

Web360[2] はマルチデバイス対応の Web アプリケーションを目指して開発しているが，開発中のテストには主に Linux，Mac，iPad，iPhone を用いたため，テストに用いていないデバイスや OS を中心に不具合が存在する可能性がある．また，動作が確認されたデバイスでも異

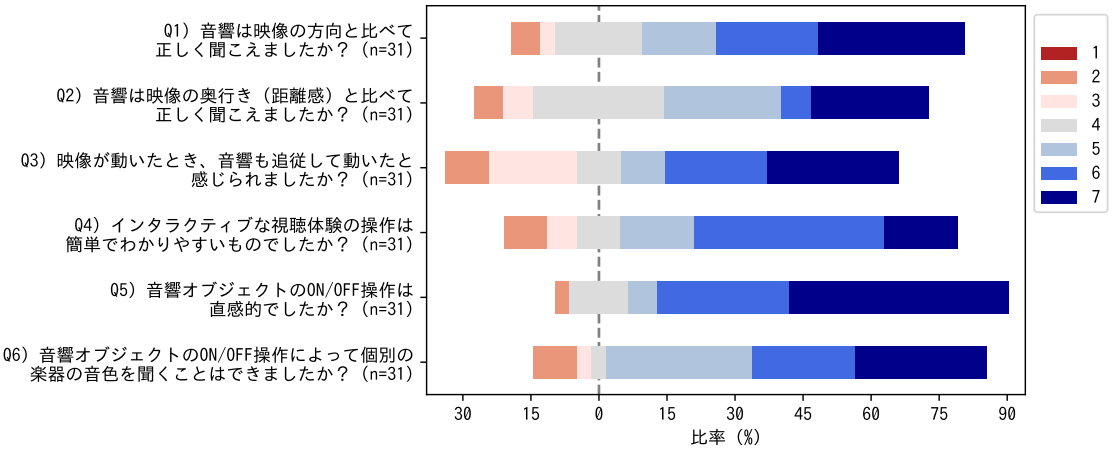


図 5.2. オンライン調査におけるアンケートの回答比率．被験者には Q1–Q6 までの 6 つの質問に対して最低評価を 1，中間を 4，最高評価を 7 とする 7 段階のリッカート尺度を用いて回答してもらった．中間評価を示す 4 の回答比率の中心を横軸の原点 0% に置き，正の方向に高評価を示す 5, 6, 7 の回答比率を，負の方向に低評価を示す 3, 2, 1 の回答比率を伸ばしている．

なる Web ブラウザでは不具合が起こる場合があり，上の環境情報はアンケート評価のためだけでなくデバッグに役立つためという意図も込めて取得した．

評価結果

アンケートの調査結果を**図 5.2** に示す．グラフの見方は**図 5.1** と同様である．Interop Tokyo 2019 での対面調査に比べて回答のばらつきや低評価の増加が目立つ結果となった．**図 5.3** では，被験者が使用したデバイスの種類を PC，タブレット・スマートフォンに大別して比較した．Web360^2 はマルチデバイス対応アプリケーションではあるものの，特にジャイロセンサを搭載したタブレット・スマートフォンでのインタラクティブな視聴体験を意識して開発している．しかしながら，すべての設問においてタブレット・スマートフォンのグループの評価が悪い結果となった．タブレット・スマートフォンのグループの自由記述回答には，アプリケーションのレンダリング処理が十分におこなえなかったことが書かれているものもあり，PC に比べて性能不足であった可能性がある．また，一般に画面サイズが小さいスマートフォンデバイスでは音声可視化オブジェクトの ON/OFF をタップ操作で切り替えることは難しいと考えられる．

次に，タブレット・スマートフォンのグループに焦点を当てて，OS の種類による評価の比較を**図 5.4** に示す．評価のサンプル数が小さいため図から明確に判断することはできないが，Android 系 OS を搭載したデバイスを使用した被験者の方が，iOS デバイス使用者に比べて低評価をつけている傾向があるようである．性能の最低水準が比較的高く安定している iOS デバイスに比べて，廉価モデルも多い Android 系 OS 搭載デバイスでは性能に大きな差があり，視聴体験をおこなうだけの性能が不足している可能性も高くなる．

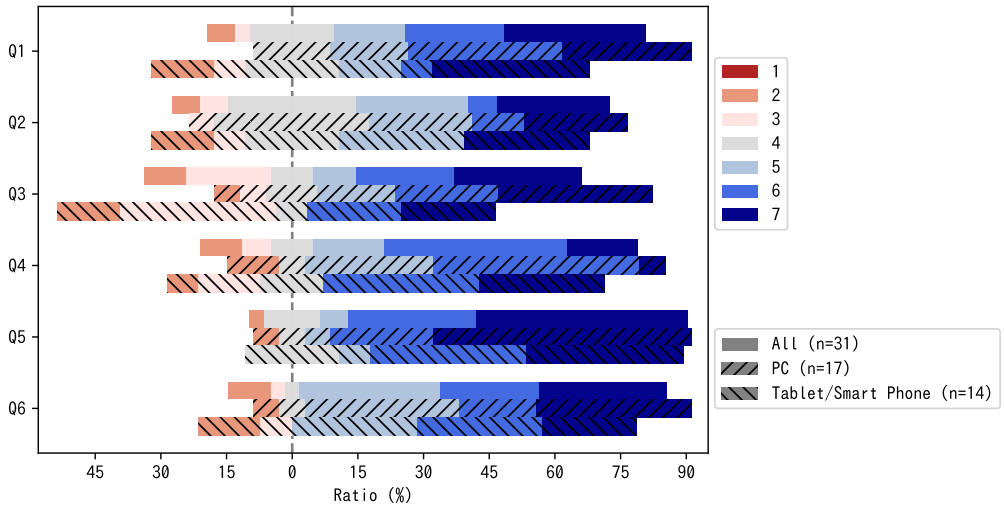


図 5.3. 被験者が使用したデバイスの種類を PC，タブレット・スマートフォンに大別して評価を比較する．

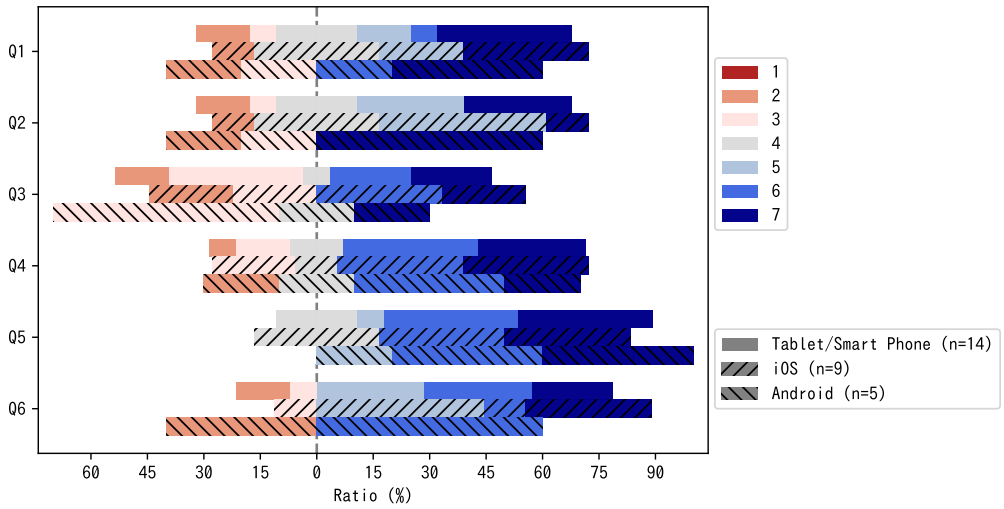


図 5.4. タブレット・スマートフォン使用者のうち，iOS 系デバイス使用者と Android 系デバイス使用者の評価を比較する．

ジャイロセンサの使用に関する評価の比較を**図 5.5** に示す．ジャイロセンサの使用によって設問 Q4 のインタラクティブな操作に関する評価に影響が出ることを期待したが，全体的に低評価の傾向が目立つ結果となった．これはジャイロセンサ搭載デバイスの大半がタブレット・スマートフォンであるため，**図 5.3** で示したデバイスの種類による評価の差が大きく現れてしまっていることが考えられる．

図 5.6 はヘッドホンの使用に関する評価の比較であり，Web Audio による立体音響はヘッドホンを使用したときの方がより聞き取りやすいと考えられるため，設問 Q1，Q2，Q3，Q6

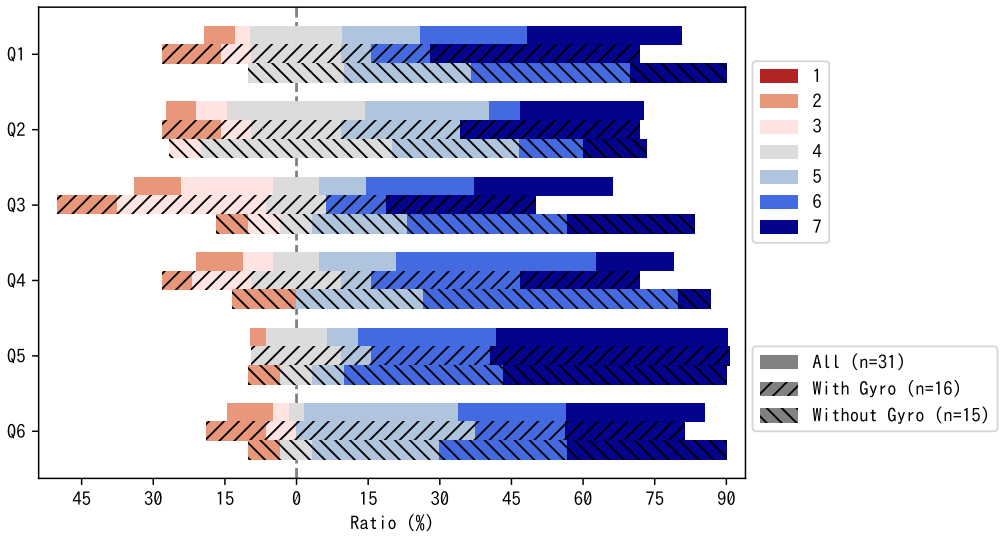


図 5.5. ジャイロセンサの使用に関する評価の比較.

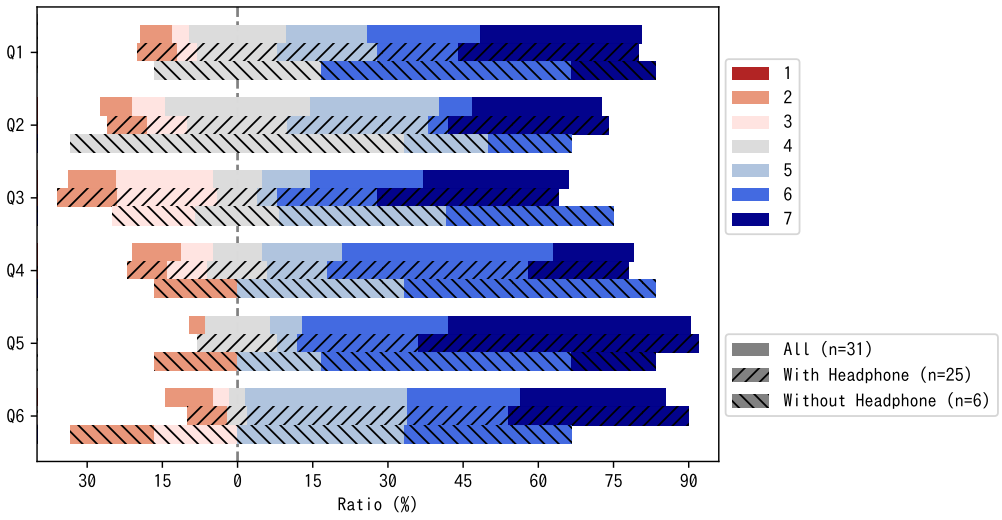


図 5.6. ヘッドホンの使用に関する評価の比較.

の回答に影響が出ると予想できる．ヘッドホンを使用していないサンプルの数が少ないため断定することはできないが，ヘッドホンを使用している被験者の方が最高評価に近い評価の比率が高い傾向が見られる．特に個別の音色を聞き取る設問 Q6 の回答はヘッドホンの使用によって大きく影響されるようである．一方で，ヘッドホン使用者の回答の中には片耳でしか聞いていないことを自由記述欄に記入しているものもあり，ヘッドホンを使用していてもステレオで音を聞いていない可能性があることには注意が必要である．

全体的な評価のばらつきには視聴コンテンツの種類も関係していると考えられる．今回の調査においては合計 31 名のうち，29 名がコンサートとジャズセッションの両方のコンテンツを

視聴している．コンサートの音響データはジャズセッションに比べて，音の方向感・奥行きがわかりにくく，各音源に複数の楽器の音色が混ざって収録されており，個別の音色も聞き取りづらい特性をもつ．これにより，説明のわかりやすさのためにジャズセッションを優先して視聴してもらった Interop Tokyo 2019 での対面調査（5.2.1 項）と比べて音の聞こえ方に関する評価に大きく影響が出ることが予想される．また，オンライン調査においては被験者の視聴体験環境だけではなく対面調査とは異なり，操作方法を直接説明することができないという課題が挙げられる．操作方法や注意点をテキストで指示したが，回答の中には操作方法がよくわからなかったというコメントもあり，低評価が目立った．

5.2.3　要求性能評価

ネットワーク性能

Web360^2 ではコンテンツの視聴のために音響データおよび最大 40 秒前後の映像データを事前に読み込む．映像のみにストリーミング再生を適用しており，視聴を開始すると 10 秒前後の映像データを随時取得する．コンサートおよびジャズセッションの両コンテンツともに約 10 MB の音響データと約 30 MB の 40 秒分の映像データ，合計約 40 MB のメディアデータを事前に読み込み，再生中はおおよそ 10 秒ごとに 5–9MB ほどの 10 秒の映像データを取得し続ける．したがって，継続的に視聴するためには最低でも 8 Mbps 前後の通信速度は必要になる．ただし，両コンテンツともに通信容量削減のために収録音源を低いビットレート (64 kbps) に変換した音響データを扱っている．また，映像データの解像度は 1960×960 であり視聴体験してみるとかなり画質が荒く感じられる．音質や画質をより向上させたい場合，用意するメディアデータの容量は上述の値に比べて大きくなり，視聴体験のためには，より安定かつ高速な通信環境が必要になるだろう．

描画性能

3 次元シーンを管理する A-Frame の `a-scene` オブジェクトは `stats` プロパティをもち，指定することで 3 次元描画に関する統計情報を表示することができる．Web360^2 では，三角形メッシュ数 (Triangles) および描画命令数 (Calls) が描画性能に影響を及ぼしていると考えられる．360° 動画を投影する A-Frame の `a-videosphere` オブジェクトの三角形メッシュ数は 3968 であり，画面内に存在する音声可視化オブジェクトの本体の球に 306，1 つの棘につき 12 だけ三角形メッシュが使われる．4.4.4 項の負荷軽減処理において述べた通り，音声可視化オブジェクトの棘は約 20 本であるので，音声可視化オブジェクトオブジェクト 1 つにつき三角形メッシュ数は 550 程度である．つまり，音声可視化オブジェクトが映っていない視野では三角形メッシュ数は `a-videosphere` オブジェクトの 3968 のみで，音声可視化オブジェクトが画面に映るごとに三角形メッシュ数は約 550 だけ増加し，両コンテンツともに最大 7500 前後の三角形メッシュが描画される．描画命令数は画面内のオブジェクト 1 つにつき 1 だけ増加し，両コンテンツともに最大約 150 以下の描画命令が出される．

`stats` プロパティはデフォルトで三角形メッシュ数が 1000 を超えると警告を示す．これ

は様々なデバイスにおいて快適に動作するための基準であると考えられる．しかしながら，a-videosphere オブジェクトのみの三角形メッシュ数で 3968 だけあり基準値の 1000 を大きく超えてしまっている．音声可視化オブジェクトによる増加分を可能な限り減少させるために死角にある棘状オブジェクトの非表示（4.4.4 項）にしており，開発中の実機テストにおいて用いたスマートフォン (Apple iPhone SE (2020)) では，この処理により動作が軽くなった．ただし，現状の負荷軽減処理でも再生を開始すると明確にわかる程度の FPS の減少が確認される．

stats プロパティにおいて描画命令数 150 程度は問題ないとされているが，5.2.2 項のオンライン調査の回答に性能不足により満足に動作しなかったとのコメントがあったことから，三角形メッシュ数や描画命令数をさらに調整するべきかもしれない．また，今回は単純にオブジェクトや描画更新頻度を減らす程度の負荷軽減処理しかおこなっていないが，他の手法を試すのもよいだろう．

第 6 章

結論

6.1 まとめ

本研究では，三次元映像・音声メディアの収録に関する管理フレームワークである SDM Ontology Version 1 を修正・拡張して，時空間情報をもつメディアを主に編集・加工段階まで，構造的に管理可能な SDM Ontology Version 2 を提案した．また，SDM Ontology Version 2 の普及を促進するために，オントロジー準拠のオープンデータによって動作するインタラクティブ Web アプリケーション「Web360^2」の開発やオントロジーを説明する Web ドキュメントの作成および公開，LOD チャレンジ 2020 への作品応募をおこなった．LOD チャレンジ 2020 では，オントロジー準拠のオープンデータセットを作品として応募したデータセット部門において最優秀賞およびスポンサー賞を受賞した．SDM Ontology Version 2 は既存の指標値やオンラインサービスによる評価を適用し，さらに「良い」オントロジーの評価を得るためには普及活動の重要であることを述べた．Web360^2 は、Interop Tokyo 2019 での対面調査，2020 年 12 月中旬から 2021 年 1 月中旬にかけてのオンライン調査の 2 種類の主観評価をおこない，Web 開発機能を用いたアプリケーション動作に関する要求性能を簡単に調査した．要求性能を十分にみたす環境においては，Web360^2 が提供する三次元映像・音声のインタラクティブな視聴体験は有効であるものの，様々な環境においては課題も多いことが判明した．また，視聴するコンテンツの種類が評価に大きな影響を与える可能性もわかった．

6.2 今後の課題

SDM Ontology Version 2 では主にメディアの編集・加工段階までを記述可能にするべくオントロジー構造を決定していったが，議論段階の構造部分も多く残っている．特に様々なメタデータを扱う Context クラスとそのサブクラスの詳細構造については，Version 1 の名残で残している構造も存在し，今後変更を加える必要があると考えられる．座標関連を記述する Geometry クラスおよび CoordinateSystem クラスについては，外部オントロジーのクラスで

ある ignf:CoordinateSystem[*1]，geom:Point[*2]のサブクラスとして定義しているが，現状では座標系原点の定義や軸の方向に関する明確な定義，移動点座標の定義などの記述をおこなうことはできない．高精度な位置情報の記述と移動点の記述は SDM Ontology の表現範囲（表現力）に大きく影響するため，議論を進めていくべき点である．また，Version 2 では再生系に関する Player クラスを追加しているものの，その表現能力は乏しく，今後はメディア制作の収録，編集・加工，配信，再生の四段階のうち，後半の配信段階，再生段階についても必要なオントロジー構造を考えていくべきである．以上のオントロジー構造上の課題に加えて，継続的な普及活動も必須である．

Web360[2] については，Web アプリケーションとしてマルチデバイス対応のアプリケーションを目指して，より幅広いデバイス性能で視聴体験可能となるように修正した方がよいと考えられる．また，オンライン調査において視聴操作がわかりづらいという意見が一定数あったことから，UI/UX デザインを向上する必要もあるだろう．現行バージョンでは実験機能となっている視聴位置の変更については，通信容量と操作応答はトレードオフであるため慎重に考えなければならない点である．

[*1] http://data.ign.fr/def/ignf#CoordinateSystem
[*2] http://data.ign.fr/def/geometrie#Point

発表文献

[1] **加藤慎**, 池田友洋, 川守田光昭, 塚田学, 江崎浩. Web360^2: 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.

[2] **Shin Kato**, Tomohiro Ikeda, Mitsuaki Kawamorita, Manabu Tsukada, and Hiroshi Esaki. Web360^2: An interactive web application for viewing 3d audio-visual contents. In *17th Sound and Music Computing Conference (SMC)*, 2020.

[3] **加藤慎**, 曽根卓朗, 塚田学, 江崎浩. 再帰的記述を可能とする映像音声メディア・オントロジー. マルチメディア, 分散, 協調とモバイル (DICOMO2020) シンポジウム, 2020. 優秀プレゼンテーション賞.

参考文献

[1] Manabu Tsukada, Keiko Ogawa, Masahiro Ikeda, Takuro Sone, Kenta Niwa, Shoichiro Saito, Takashi Kasuya, Hideki Sunahara, and Hiroshi Esaki. Software defined media: Virtualization of audio-visual services. In *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2017.

[2] 塚田学, 小川景子, 池田雅弘, 曽根卓朗, 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] Ray Atarashi, Takuro Sone, Yu Komohara, Manabu Tsukada, Takashi Kasuya, Hiraku Okumura, Masahiro Ikeda, and Hiroshi Esaki. The software defined media ontology for music events. In *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music*, pp. 15–23. ACM, 2018.

[4] 菰原裕, 塚田学, 江崎浩, 曽根卓朗, 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] Daniel Garijo. Widoco: a wizard for documenting ontologies. In *International Semantic Web Conference*, pp. 94–102. Springer, Cham, 2017.

[6] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pp. 205–227. IGI Global, 2011.

[7] Tim Berners-Lee. Linked data - design issues. (online), available from ⟨`https://www.w3.org/DesignIssues/LinkedData.html`⟩. (accessed 2020-04-26).

[8] Tim Berners-Lee. 5-star open data. (online), available from ⟨`https://5stardata.info/en/`⟩. (accessed 2020-04-26).

[9] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. The music ontology. In *ISMIR*, Vol. 2007, p. 8th. Citeseer, 2007.

[10] Yves Raimond, Thomas Gängler, Frederick Giasson, Kurt Jacobson, György Fazekas, Simon Reinhardt, and Alexandre Passant. The music ontology. (online), available from ⟨`http://musicontology.com/`⟩. (accessed 2020-04-26).

[11] Yves Raimond and Samer Abdallah. The timeline ontology. (online), available from

⟨http://purl.org/NET/c4dm/timeline.owl⟩. (accessed 2020-04-26).

[12] Yves Raimond and Samer Abdallah. The event ontology. (online), available from ⟨http://purl.org/NET/c4dm/event.owl⟩. (accessed 2020-04-26).

[13] Ian Davis and Richard Newman. Expression of core frbr concepts in rdf. (online), available from ⟨https://vocab.org/frbr/core⟩. (accessed 2020-04-26).

[14] Dan Brickley and Libby Miller. Foaf vocabulary specification 0.99. (online), available from ⟨http://xmlns.com/foaf/spec/⟩. (accessed 2020-04-26).

[15] György Fazekas and Mark B Sandler. The studio ontology framework. In *ISMIR*, pp. 471–476, 2011.

[16] Thomas Wilmering, György Fazekas, and Mark B Sandler. The audio effects ontology. In *ISMIR*, pp. 215–220, 2013.

[17] Thomas Wilmering and György Fazekas. The audio effect ontology. (online), available from ⟨https://w3id.org/aufx/ontology/1.0#⟩. (accessed 2020-04-26).

[18] Sefki Kolozali, Mathieu Barthet, György Fazekas, and Mark B Sandler. Knowledge representation issues in musical instrument ontology design. In *ISMIR*, pp. 465–470, 2011.

[19] Sefki Kolozali, György Fazekas, Mathieu Barthet, and Mark Sandler. A framework for automatic ontology generation based on semantic audio analysis. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[20] Alo Allik, György Fazekas, and Mark B Sandler. An ontology for audio features. In *ISMIR*, pp. 73–79, 2016.

[21] György Fazekas and Alo Allik. Audio features ontology. (online), available from ⟨https://semantic-audio.github.io/afo/⟩. (accessed 2020-04-26).

[22] Miguel Ceriani and György Fazekas. Audio commons ontology: a data model for an audio content ecosystem. In *International Semantic Web Conference*, pp. 20–35. Springer, 2018.

[23] György Fazekas and Miguel Ceriani. The audio commons ontology. (online), available from ⟨https://w3id.org/ac-ontology/aco#⟩. (accessed 2020-04-26).

[24] Luca Turchet, Francesco Antoniazzi, Fabio Viola, Fausto Giunchiglia, and György Fazekas. The Internet of Musical Things Ontology. *Journal of Web Semantics*, Vol. 60, p. 100548, 2020.

[25] Francesco Antoniazzi. Internet of musical things ontology (iomust). (online), available from ⟨https://fr4ncidir.github.io/IoMusT/⟩. (accessed 2020-04-26).

[26] ISO/IEC. 23008-3:2019, Information technology – High efficiency coding and media delivery in heterogeneous environments – Part3: 3D audio. Standard, ISO/IEC, 2019.

[27] Seungkwon Beack, Jongmo Sung, Jeongil Seo, and Taejin Lee. MPEG Surround Extension Technique for MPEG-H 3D Audio. *ETRI Journal*, Vol. 38, No. 5, pp.

829–837, 2016.

[28] Jürgen Herre, Johannes Hilpert, Achim Kuntz, and Jan Plogsties. MPEG-H 3D audio —The new standard for coding of immersive spatial audio. *IEEE Journal of selected topics in signal processing*, Vol. 9, No. 5, pp. 770–779, 2015.

[29] Dolby Laboratories. Dolby Atmos®Specifications. Isuue 3, Dolby Laboratories, 2015.

[30] Dolby Laboratories. Dolby Atmos®Home Theater Installation Guidelines. Technical report, Dolby Laboratories, 2018.

[31] DTS, Inc. Home Theater Sound Gets Real. (online), available from ⟨`https://dts.com/dtsx`⟩. (accessed 2019-09-05).

[32] Auro Technologies. AUROMAX®Next generation Immersive Sound system. Technical report, Auro Technologies, 2015.

[33] Matthias Frank, Franz Zotter, and Alois Sontacchi. Producing 3D audio in ambisonics. In *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology–Cinema, Television and the Internet*. Audio Engineering Society, 2015.

[34] Ricoh Company, Ltd. 360-degree camera RICOH THETA. (online), available from ⟨`https://theta360.com/`⟩. (accessed 2019-09-05).

[35] 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. SDM360^2: 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.

[36] Takashi Kasuya, Manabu Tsukada, Yu Komohara, Shigeki Takasaka, Takuhiro Mizuno, Yoshitaka Nomura, Yuta Ueda, and Hiroshi Esaki. Livration: Remote vr live platform with interactive 3d audio-visual service. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 1–7. IEEE, 2019.

[37] 粕谷貴司, 塚田学, 菰原裕, 高坂茂樹, 水野拓宏, 野村譲誉, 上田雄太, 江崎浩. LiVRation: VR による自由視聴点映像音声のインタラクティブ再生. マルチメディア, 分散協調とモバイルシンポジウム 2018 論文集, Vol. 2018, pp. 514–522, 2018.

[38] Google Creative Lab and Song Exploder. Song Exploder Presents: Inside Music. (online), available from ⟨`https://experiments.withgoogle.com/webvr/inside-music/view/`⟩. (accessed 2019-09-06).

[39] Apache Software Foundation. Apache License, Version 2.0. (online), available from ⟨`https://www.apache.org/licenses/LICENSE-2.0`⟩. (accessed 2019-10-14).

[40] 加藤慎, 曽根卓朗, 塚田学, 江崎浩. 再帰的記述を可能とする映像音声メディア・オントロジー. マルチメディア, 分散, 協調とモバイル (DICOMO2020) シンポジウム, 2020.

[41] Ramanathan V Guha, Dan Brickley, and Steve Macbeth. Schema.org: evolution of structured data on the web. *Communications of the ACM*, Vol. 59, No. 2, pp. 44–51, 2016.

[42] Nathalie Abadie and Ghislain Atemezing. Ontology of coordinates reference systems. (online), available from ⟨http://data.ign.fr/def/ignf/20190213.en.htm⟩. (accessed 2020-11-28).

[43] Nathalie Abadie and Ghislain Atemezing. Ontology of geometric primitives. (online), available from ⟨http://data.ign.fr/def/geometrie/20190212.en.htm⟩. (accessed 2020-11-28).

[44] Shin Kato, Tomohiro Ikeda, Mitsuaki Kawamorita, Manabu Tsukada, and Hiroshi Esaki. Web360^2: An interactive web application for viewing 3d audio-visual contents. In *17th Sound and Music Computing Conference (SMC)*, 2020.

[45] 加藤慎, 池田友洋, 川守田光昭, 塚田学, 江崎浩. Web360^2: 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.

[46] Masahiro Ikeda, Takuro Sone, Kenta Niwa, Shoichiro Saito, Manabu Tsukada, and Hiroshi Esaki. New recording application for software defined media. In *Audio Engineering Society Convention 141*. Audio Engineering Society, 2016.

[47] 山路敦. 多数の点を球面上に一様分布させるソフトウェア gss generator. 情報地質, Vol. 12, No. 1, pp. 3–12, 2001.

[48] Miriam Fernández, Chwhynny Overbeeke, Marta Sabou, and Enrico Motta. What makes a good ontology? a case-study in fine-grained knowledge reuse. In *Asian Semantic Web Conference*, pp. 61–75. Springer, 2009.

[49] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 10, No. 2, pp. 7–34, 2014.

[50] Mark A Musen. The protégé project: a look back and a look forward. *AI matters*, Vol. 1, No. 4, pp. 4–12, 2015.

謝辞

本論文の執筆にあたり多くの方々にご指導，ご協力をいただきました．心より感謝申し上げます．指導教員である江崎浩教授には，いつも大局的な視点から研究内容や発表に関するご指摘をいただきました．また，本研究における主観評価調査を実施する際には教授自ら広報してくださり，お忙しい中お力添えいただいたこと御礼申し上げます．塚田学准教授にはご多忙の中，定期的にミーティングを開いていただき，研究の細かな方針や論文内容に関する様々なご指導を親身にしてくださりましたことに深く感謝申し上げます．落合秀也准教授，山本成一助教授には研究室のミーティングにおいて的確なアドバイスをいただきました．Software Defined Media コンソーシアムの皆様には本研究へ協力していただいたほか，商業的な観点からご意見やご指摘をいただきました．研究生活においてとても新鮮でよい刺激になっておりました．研究室生活について様々なことをご教示してくださった研究室の先輩方，切磋琢磨し，励まし合い，刺激を与えてくれた同期一同，後輩一同に感謝いたします．研究室の秘書である岩井愛映子さん，高橋富美さんには諸事務を通じて大変お世話になりました．最後に，これまでの学生生活をあらゆる面から支えてくださった家族，友人，先生方，すべての皆様に深く感謝申し上げます．

2021 年 1 月 28 日

加藤 慎

付録 A

SDM Ontology Version 2 準拠のデータ例

4 章で述べたように Web360^2 では SDM Ontology Version 2 に準拠するデータを用いてアプリケーションを動作させている．ここでは，簡単なメディア制作の例を挙げて，SDM Ontology Version 2 に準拠するデータの作成例を示す．**図 A.1** は最小限の収録形態の例を表す．図中では具体的に音に関する収録であることを示しているが，収録機器をカメラに置き換えれば静止画や映像収録の様子を表すことができる．さらに，収録対象も人物に限らずモノや空間を対象としてもよく，環境音や自然風景画の収録，収録機器を気温や湿度などのセンサ類に置き換えれば環境データの収集を表すようになる．

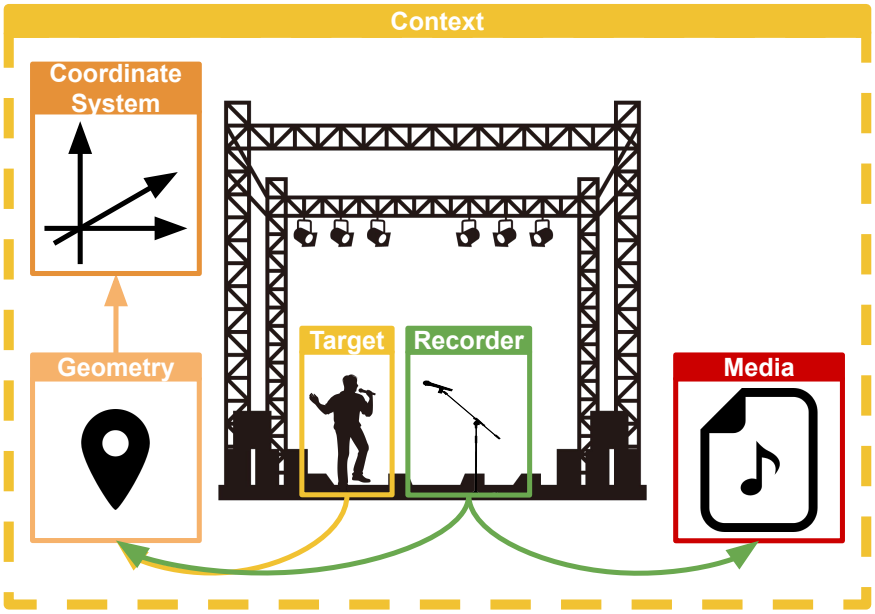


図 A.1. 最小限の収録形態の例．図では例として音に関する収録を示しているが，収録内容は音に限らず様々であってよい．

図 A.2 では，この最小収録形態を SDM Ontology Version 2 準拠の RDF データに記述した例を示す．RDF データ内には，SDM Ontology Version 2 が記述するメディア収録の中核となる部分のみが含まれており，収録対象や内容に関する詳細な情報は書かれていない．実運用する際には，音楽収録であれば音楽関連のオントロジーと連携するなど，収録内容に応じて専門の外部オントロジーを用いて，収録対象となる人物やモノや収録内容に関する専門的な詳細情報を記述する必要がある．また，例の簡略化のためにメディアの収録段階までしか考慮していないが，3 章で紹介した Processor クラスや Player クラスを用いることで，メディア編集段階と簡単な再生系に関する記述も可能になる．

```
@prefix sdm: <http://sdm.hongo.wide.ad.jp/resource/> .
@prefix sdmo: <http://sdm.hongo.wide.ad.jp/sdmo/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix ignf: <http://data.ign.fr/def/ignf#> .
@prefix geom: <http://data.ign.fr/def/geometrie#> .
@prefix schema: <http://schema.org/> .

# Context > SDMEvent
# イベント情報

sdm:SimpleRecordingEvent
    a sdmo:SDMEvent ;
    schema:name "音源収録"@ja ;
    schema:contentLocation [
            a schema:Place ;
            schema:name "東京大学 工学部号館 2 階 10 102号室 B1"@ja ;
            schema:address "東京都文京区本郷 7-3-1"@ja ;
        ] ;
    schema:startDate "2021-01-15T10:00:00.00+09:00"^^xsd:dateTime ;
    schema:duration "PT1H"^^xsd:duration ;
    sdmo:attendedBy sdm:SimpleRecordingTarget ;
    sdmo:isContextOf sdm:SimpleRecordingAudioRecorder .


# Context > Target
# 収録対象

sdm:SimpleRecordingTarget
    a sdmo:Target ;
    sdmo:attends sdm:SimpleRecordingEvent ;
    sdmo:recordedBy sdm:SimpleRecordingAudioRecorder ;
    sdmo:geometryAt sdm:SimpleRecordingTargetGeometry .


# Recorder > AudioRecorder
# 収録機器

sdm:SimpleRecordingAudioRecorder
    a sdmo:AudioRecorder ;
    sdmo:hasContext sdm:SimpleRecordingEvent ;
    sdmo:records sdm:SimpleRecordingTarget ;
    sdmo:outputTo sdm:SimpleRecordingAudioMedia ;
    sdmo:geometryAt sdm:SimpleRecordingAudioRecorderGeometry .


# Media > AudioMedia
# 収録されたメディアデータ

sdm:SimpleRecordingAudioMedia
    a sdmo:AudioMedia ;
    schema:contentUrl <https://sdm.example.net/simple/recording/20210115.wav> ;
    sdmo:hasContext sdm:SimpleRecordingEvent ;
    sdmo:mediaStartAt "2021-01-15T10:00:00.00+09:00"^^xsd:dateTime ;
    sdmo:startAt "0.0"^^xsd:decimal ;
```

```
55      sdmo:endAt "3600.0"^^xsd:decimal .
56
57
58  # CoordinateSystem
59  # 座標系の定義
60
61  sdm:SimpleRecordingCoordinateSystem
62      a sdmo:CoordinateSystem ;
63      ignf:axis sdm:SimpleRecordingAxesList ;
64      sdmo:measures sdm:SimpleRecordingTargetGeometry ,
65                    sdm:SimpleRecordingAudioRecorderGeometry .
66
67  sdm:SimpleRecordingAxesList
68      a ignf:AxesList ;
69      rdf:first sdm:SimpleRecordingAxisX ;
70      rdf:rest [
71              a ignf:AxesList ;
72              rdf:first sdm:SimpleRecordingAxisY ;
73              rdf:rest [
74                      a ignf:AxesList ;
75                      rdf:first sdm:SimpleRecordingAxisZ ;
76                      rdf:rest rdf:nil ;
77                  ] ;
78          ] .
79
80  sdm:SimpleRecordingAxisX
81      a ignf:CoordinateSystemAxis ;
82      ignf:axisAbbrev "x"^^xsd:string ;
83      ignf:axisDirection "starboard"^^xsd:string .
84
85  sdm:SimpleRecordingAxisY
86      a ignf:CoordinateSystemAxis ;
87      ignf:axisAbbrev "y"^^xsd:string ;
88      ignf:axisDirection "up"^^xsd:string .
89
90  sdm:SimpleRecordingAxisZ
91      a ignf:CoordinateSystemAxis ;
92      ignf:axisAbbrev "z"^^xsd:string ;
93      ignf:axisDirection "aft"^^xsd:string .
94
95
96  # Geometry
97  # 座標の定義
98
99  sdm:SimpleRecordingTargetGeometry
100     a sdmo:Geometry ;
101     geom:coordX "0.1"^^xsd:double ;
102     geom:coordY "2.3"^^xsd:double ;
103     geom:coordZ "4.5"^^xsd:double ;
104     sdmo:geometryOf sdm:SimpleRecordingTarget ;
105     sdmo:measuredBy sdm:SimpleRecordingCoordinateSystem .
106
107 sdm:SimpleRecordingAudioRecorderGeometry
108     a sdmo:Geometry ;
109     geom:coordX "6.7"^^xsd:double ;
110     geom:coordY "8.9"^^xsd:double ;
111     geom:coordZ "1.0"^^xsd:double ;
112     sdmo:geometryOf sdm:SimpleRecordingAudioRecorder ;
113     sdmo:measuredBy sdm:SimpleRecordingCoordinateSystem .
```

図 A.2. SDM Ontology Version 2 準拠の単純な RDF データの例．最小限の収録形態を Turtle 記法で記述している

付録 B

SDM Ontology Version 2 の語彙一覧

5.1.2 項で述べた通り，SDM Ontology Version 2 の語彙体系は，Web ドキュメント[*1]として公開されている．本章では，執筆時点における最新版[*2]の語彙を一覧としてまとめる．**表 B.1** は以下の語彙一覧における RDF の名前空間を表す接頭辞である．語彙一覧においては，簡単のために SDM Ontology Version 2 の語彙は接頭辞を用いずに記述しており，外部オントロジーなどの語彙にのみ接頭辞を用いていることに注意されたい．

表 B.1. RDF の名前空間.

Prefix	**URI**
	<http://sdm.hongo.wide.ad.jp/sdmo/>
ignf	<http://data.ign.fr/def/ignf#>
xsd	<http://www.w3.org/2001/XMLSchema#>
schema	<http://schema.org/>
geom	<http://data.ign.fr/def/geometrie#>

[*1] https://sdm.wide.ad.jp/sdmo/

[*2] https://sdm.wide.ad.jp/sdmo/2.0.0-beta2/

B.1 クラス一覧

Analyzer	Class
URI	http://sdm.hongo.wide.ad.jp/sdmo/Analyzer
Comment	メディアを読み込んで処理を行う Processor のサブクラス．メディア生成を行わない．
Super Classes	Processor
Sub Classes	
Domain	
Range	

Audio Media	Class
URI	http://sdm.hongo.wide.ad.jp/sdmo/AudioMedia
Comment	音声メディアを表現するための Media のサブクラス．
Super Classes	Media
Sub Classes	
Domain	
Range	

Audio Player	Class
URI	http://sdm.hongo.wide.ad.jp/sdmo/AudioPlayer
Comment	音声メディアの再生システムを表現するための Player のサブクラス．
Super Classes	Player
Sub Classes	
Domain	
Range	

Audio Recorder　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/AudioRecorder
Comment	音声メディアを生成する音声収録システムを表現するための Recorder のサブクラス.
Super Classes	Recorder
Sub Classes	
Domain	
Range	

Composite Media　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/CompositeMedia
Comment	複数のメディアを統合したメディアを表現するための Media のサブクラス.
Super Classes	Media
Sub Classes	
Domain	has component, has media event
Range	is component of, is media event of

Composite Player　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/CompositePlayer
Comment	複数のメディア再生システムを統合するための Player のサブクラス.
Super Classes	Player
Sub Classes	
Domain	has component
Range	is component of

Composite Processor　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/CompositeProcessor
Comment	複数のメディア処理システムを統合するための Processor のサブクラス.
Super Classes	Processor
Sub Classes	
Domain	has component
Range	is component of

Composite Recorder

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/CompositeRecorder
Comment	複数の音声収録システムを統合するための Recorder のサブクラス.
Super Classes	Recorder
Sub Classes	
Domain	has component
Range	is component of

Context

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Context
Comment	Recorder, Processor, Player, Media, Context に付加する MetaData を統一的に扱うためのクラス．SDMO の主要クラスの一つ.
Super Classes	
Sub Classes	Program, SDM event, Target
Domain	geometry at, is context of, has context, played by, recorded by, schema:name
Range	geometry of, is context of, has context, plays, records

Converter

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Converter
Comment	メディアを読み込んで加工してメディアを出力する Processor のサブクラス.
Super Classes	Processor
Sub Classes	
Domain	
Range	

Coordinate System

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/CoordinateSystem
Comment	座標系を定義するためのクラス．Geometry で表現した値の意味を規定する．緯度経度の地理座標系やイベント会場内に定義する空間座標系，収録したメディアファイルの時間軸などを記述する.
Super Classes	ignf:CoordinateSystem
Sub Classes	
Domain	measured by, measures
Range	measured by, measures

Data Media　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/DataMedia
Comment	データメディアの再生システムを表現するための Media のサブクラス.
Super Classes	Media
Sub Classes	
Domain	
Range	

Data Player　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/DataPlayer
Comment	データメディアの再生システムを表現するための Player のサブクラス.
Super Classes	Player
Sub Classes	
Domain	
Range	

Data Recorder　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/DataRecorder
Comment	データメディアの収録システムを表現するための Recorder のサブクラス.
Super Classes	Recorder
Sub Classes	
Domain	
Range	

Generator　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Generator
Comment	メディアの生成を行う Processor のサブクラス．メディア入力を持たない.
Super Classes	Processor
Sub Classes	
Domain	
Range	

Geometry

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Geometry
Comment	CoordinateSystem で定義された座標系を用いた座標を記述するためのクラス.
Super Classes	geom:Point
Sub Classes	
Domain	geometry of
Range	geometry at

Media

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Media
Comment	メディアファイルを表現するためのクラス．SDMO の主要クラスの一つ.
Super Classes	
Sub Classes	Audio Media, Composite Media, Data Media, Video Media
Domain	end at, is component of, is media of, media start at, start at, schema:contentUrl, schema:name
Range	has component, has media

Media Event

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/MediaEvent
Comment	メディアと MediaEvent が属する CompositeMedia の相対時間軸上の時刻情報の対応関係を記述するクラス．MediaEvent の相対時刻情報をもとに CompositeMedia の時間軸上にメディアを配置する.
Super Classes	
Sub Classes	
Domain	event time, has media, is media event of
Range	has media event, is media of

Player

Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Player
Comment	メディアの再生システムを表現するためのクラス．SDMO の主要クラスの一つ.
Super Classes	
Sub Classes	Audio Player, Composite Player, Data Player, Video Player
Domain	geometry at, has context, input from, is component of, plays, schema:name
Range	geometry of, has component, is context of, output to, played by

Processor　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Processor
Comment	メディアの分析・編集・加工システムを表現するためのクラス．SDMO の主要クラスの一つ.
Super Classes	
Sub Classes	Analyzer, Composite Processor, Converter, Generator
Domain	geometry at, has context, input from, is component of, output to, schema:name
Range	geometry of, has component, input from, is context of, output to

Program　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Program
Comment	収録対象のパフォーマンス (演奏や演技など) を表現するための Context のサブクラス.
Super Classes	Context
Sub Classes	
Domain	performed by, schema:duration, schema:startDate
Range	performs

Recorder　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/Recorder
Comment	音声メディアを出力する音声収録システムを表現するためのクラス．SDMO の主要クラスの一つ.
Super Classes	
Sub Classes	Audio Recorder, Composite Recorder, Data Recorder, Video Recorder
Domain	geometry at, has context, is component of, output to, records, schema:name
Range	geometry of, has component, input from, is context of, recorded by

SDM Event　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/SDMEvent
Comment	収録対象となったイベント全体を表現するための Context のサブクラス.
Super Classes	Context
Sub Classes	
Domain	attended by, schema:contentLocation, schema:duration, schema:startDate
Range	attends

Target

Class

URI	`http://sdm.hongo.wide.ad.jp/sdmo/Target`
Comment	Recorder の収録対象を表現するための Context のサブクラス．特定の出演者，オーケストラのパート，楽器などを記述する．
Super Classes	Context
Sub Classes	
Domain	attends, geometry at, performs, played by, recorded by
Range	attended by, geometry of, performed by, plays, records

Video Media

Class

URI	`http://sdm.hongo.wide.ad.jp/sdmo/VideoMedia`
Comment	映像メディアを表現する Media のサブクラス．
Super Classes	Media
Sub Classes	
Domain	
Range	

Video Player

Class

URI	`http://sdm.hongo.wide.ad.jp/sdmo/VideoPlayer`
Comment	映像メディアの再生システムを表現するための Player のサブクラス．
Super Classes	Player
Sub Classes	
Domain	
Range	

Video Recorder　Class

URI	http://sdm.hongo.wide.ad.jp/sdmo/VideoRecorder
Comment	映像メディアを出力する映像収録システムを表現するための Recorder のサブクラス.
Super Classes	Recorder
Sub Classes	
Domain	
Range	

B.2　オブジェクトプロパティー覧

attended by　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/attendedBy
Comment	自身が参加している SDMEvent を参照する．attends と逆の意味.
Domain	SDM Event
Range	Target
Inverse	attends

attends　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/attends
Comment	自身が SDMEvent に参加している状態を表す．attendedBy で逆方向リンクを表現できる.
Domain	Target
Range	SDM Event
Inverse	attended by

geometry at

Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/geometryAt
Comment	geometryAt でリンクした Geometry でオブジェクトの位置・時間情報を表現する．リンク先の Geometry からの逆方向リンクは geometryOf で表す.
Domain	Context or Player or Processor or Recorder
Range	Geometry
Inverse	geometry of

geometry of

Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/geometryOf
Comment	geometryAt で自身をリンクしているオブジェクトへの逆方向リンク.
Domain	Geometry
Range	Context or Player or Processor or Recorder
Inverse	geometry at

has component

Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/hasComponent
Comment	CompositeRecorder, CompositeProcessor, CompositePlayer, CompositeMedia に包含されている Recorder, Processor, Player, Media をそれぞれ表現する．包含されたオブジェクトからは isComponentOf で被包含関係を表現する.
Domain	Composite Media or Composite Player or Composite Processor or Composite Recorder
Range	Media or Player or Processor or Recorder
Inverse	is component of

has context

Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/hasContext
Comment	オブジェクトに付随する Context を表現する．リンク先の Context からは isContextOf で逆方向リンクを表現できる.
Domain	Context or Media or Player or Processor or Recorder
Range	Context
Inverse	is context of

has media　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/hasMedia
Comment	MediaEvent が参照する Media を表現する．参照先の Media からは isMediaOf で逆方向リンクを表現できる．
Domain	Media Event
Range	Media
Inverse	is media of

has media event　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/hasMediaEvent
Comment	CompositeMedia が包含している MediaEvent を表す．包含された MediaEvent からは isMediaEventOf で包含元の CompositeMedia をリンクする．
Domain	Composite Media
Range	Media Event
Inverse	is media event of

input from　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/inputFrom
Comment	入力となるメディアを指定するためのリンク．リンク先は Recorder, Processor, Media のいずれかになる．リンク先からは outputTo により逆方向リンクを表現できる．
Domain	Media or Player or Processor
Range	Media or Processor or Recorder
Inverse	output to

is component of　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/isComponentOf
Comment	自身を包含している CompositeRecorder, CompositeProcessor, CompositePlayer, CompositeMedia を表現する．hasComponent と逆の意味.
Domain	Media or Player or Processor or Recorder
Range	Composite Media or Composite Player or Composite Processor or Composite Recorder
Inverse	has component

is context of — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/isContextOf
Comment	自身を参照しているオブジェクトを参照する．hasContext と逆の意味.
Domain	Context
Range	Context or Media or Player or Processor or Recorder
Inverse	has context

is media event of — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/isMediaEventOf
Comment	MediaEvent がリンク先の CompositeMedia に包含されていることを表す．hasMediaEvent と逆の意味.
Domain	Media Event
Range	Composite Media
Inverse	has media event

is media of — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/isMediaOf
Comment	自身を参照している MediaEvent を表現する．hasMedia と逆の意味.
Domain	Media
Range	Media Event
Inverse	has media

measured by — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/measuredBy
Comment	CoordinateSystem を参照することで Geometory で表現している値の意味を規定する．measure と逆の意味.
Domain	Coordinate System or Geometry or Media or Recorder
Range	Coordinate System
Inverse	measures

measures — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/measures
Comment	自身を参照している Geometry を表現する．measuredBy と逆の意味.
Domain	Coordinate System
Range	Coordinate System or Geometry or Media or Recorder
Inverse	measured by

output to　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/outputTo
Comment	メディアの出力先を指定するためのリンク．リンク先からは inputFrom により逆方向リンクを表現できる．
Domain	Media or Processor or Recorder
Range	Media or Player or Processor
Inverse	input from

performed by　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/performedBy
Comment	自身を演奏している Target を表す．performs と逆の意味．
Domain	Program
Range	Target
Inverse	performs

performs　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/performs
Comment	自身がリンク先の Program を演じるまたは演奏することを表現する述語．performedBy で逆方向リンクを表現できる．
Domain	Target
Range	Program
Inverse	performed by

played by　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/playedBy
Comment	自身が演奏者によって演奏されることを表す．plays で逆方向リンクを表現できる．
Domain	Context
Range	Player
Inverse	plays

plays　Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/plays
Comment	演奏者が楽器を再生することを表す．playedBy で逆方向リンクを表現できる．
Domain	Player
Range	Context
Inverse	played by

recorded by — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/recordedBy
Comment	自身が Recorder の収録対象となっていることを表現する．records で逆方向リンクを表現できる．
Domain	Context
Range	Recorder
Inverse	records

records — Object Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/records
Comment	Recorder の収録対象を表現する．recordedBy で逆方向リンクを表現できる．
Domain	Recorder
Range	Context
Inverse	recorded by

B.3 データ型プロパティ一覧

end at — Data Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/endAt
Comment	メディアの再生区間の終了時間．mediaStartAt で指定した時刻を 0 としたオフセット時間を指定する．
Domain	Media
Range	xsd:dateTimeStamp

event time — Data Property

URI	http://sdm.hongo.wide.ad.jp/sdmo/eventTime
Comment	MediaEvent で Media の再生開始時刻を CompositeMedia の時間軸で指定する．指定した時刻において Media の再生区間が始まることを表現する．
Domain	Media Event
Range	xsd:dateTimeStamp

media start at　Data Property

URI	`http://sdm.hongo.wide.ad.jp/sdmo/mediaStartAt`
Comment	メディアファイルの有効区間開始時刻をメディアファイル先頭からのオフセットで表記する．startAt 及び endAt で再生区間を指定する際の時刻 0 を規定する．
Domain	Media
Range	xsd:dateTimeStamp

start at　Data Property

URI	`http://sdm.hongo.wide.ad.jp/sdmo/startAt`
Comment	メディアの再生区間の開始時間．mediaStartAt で指定した時刻を 0 としたオフセット時間を指定する．
Domain	Media
Range	xsd:dateTimeStamp