The Software Defined Media Ontology for Music Events

Ray Atarashi, Takuro Sone, Yu Komohara, Manabu Tsukada, Takashi Kasuya, Hiraku Okumura, Masahiro Ikeda, Hiroshi Esaki

The Software Defined Media Ontology for Music Events

Ray Atarashi
IIJ Innovation Institute Inc.
ray@iijlab.net

Takuro Sone
YAMAHA Corporation
takurou.sone@music.yamaha.com

Yu Komohara
The University of Tokyo
como@hongo.wide.ad.jp

Manabu Tsukada
The University of Tokyo
tsukada@hongo.wide.ad.jp

Takashi Kasuya
Takenaka Corporation
kasuya@hongo.wide.ad.jp

Hiraku Okumura
YAMAHA Corporation
hiraku.okumura@music.yamaha.com

Masahiro Ikeda
YAMAHA Corporation
masahiro.ikeda@music.yamaha.com

Hiroshi Esaki
The University of Tokyo
hiroshi@wide.ad.jp

ABSTRACT

With the advent of viewing services based on the Internet, the importance of object-based viewing services for interpreting objects existing in space and utilizing them as content is increasing. Since 2014, the Software Defined Media Consortium has been researching object-based media and Internet-based viewing spaces. This paper defineds a framework in event participants and professional recorders each freely share recorded data, and a third party can creates an application based on the data. This study aims to provide an SDM ontology-based contents management mechanism with a detailed description of the object-based audio and video data and the recording environment. The data can be shared via the Internet and is highly reusable. We implemented this management mechanism and have developed and validated applications that are capable of interactively playing 3D content from any viewpoints freely.

CCS CONCEPTS

• **Applied computing** → **Document metadata**; **Sound and music computing**; • **Computer systems organization** → **Special purpose systems**;

KEYWORDS

Content management, Ontology, RDF

1 INTRODUCTION

In recent years, carriable recording equipment embedded in smartphones always has become commonplace, and a situation has been created in which recorded video and audio are instantly transmitted, shared, and processed via the Internet. Conventionally, bitmap data input from recording object and two channels of audio data input from a stereo microphone are directly transmitted and replayed on the receiving side without modification. Furthermore, an object-oriented method is attracting attention in which data is interpreted as a three-dimensional model, is transmitted by being decomposed into a multiple viewing objects, and space is reproduced using these objects on the receiving side.

Thereby, it is possible to render a flexible three-dimensional representation in accordance with the configuration of a receiving system, such as a head-mounted display (HMD), 3D television, or stereophonic audio device, and to render an unprecedented representation by separately receiving and combining viewing objects from other sources. For example, by converting audio and video into objects, it is possible to produce interactive content, such as a view from the support stand side, switching viewing away from the first person perspective of the player, and viewing a favorite person in the idle group in the vicinity in accordance with the request of the viewer.

We established the Software Defined Media (SDM) Consortium[1] in 2014, and we have continued to study object-based digital media and Internet-based viewing spaces [17]. SDM defined a distributed recording environment framework with multiple recording devices, as shown in Figure 1. For example, each of the event participants and professional recorders freely share photographed data, and a third party creates an application based on this data. This framework is intended to broaden the scope of data collection and data utilization by assuming different environments for the person recording and the author of the application.

The contributions of this paper include providing an collaborative environment, recording system, editing system, and playback system to create new content based on the SDM framework.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 describes the objectives of this paper and the requirements of the SDM Ontology design. Section 4 describes detailed data of video and audio actually recorded at music events. Section 5 describes the SDM Ontology and design policy in detail. Section 6 describes designing and implementing systems that are implemnented within the SDM Ontology. Section 7 describes the SDM 3602 and LiVRation, which we developed as an application that utilizes the SDM Ontology. The flexibility of this ontology will be verified by introducing various other use cases with these applications and the SDM Ontology. Finally, Section 8 concludes this paper and provides direction for future research.

SAAM2018, October 2018, Monterey, California, USA

[1] http://sdm.wide.ad.jp/

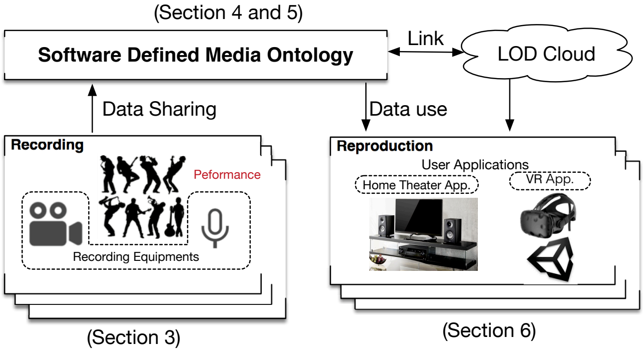

Figure 1: The Target Framework

2 RELATED WORKS

Sound recording and reproduction systems may be broadly classified into three types: channel-based, object-based, and scene-based [16]. *Channel-based* systems evolved from stereo sound (two channels) to surround sound (multi-channel). The Super Hi-Vision [11] that began test broadcasting in 2016 employs a 22.2 multi-channel stereo sound system [7]. An *object-based* system renders the sound filed from the position of the speakers in the reproduction environment by recording tone color data from a sound source, and the three-dimensional position is attached as metadata. For example, Dolby Atmos [3] and AuroMax [1] are increasingly used in movie theaters and home theaters. In addition, MPEG-H [4, 8] has been standardized by the Moving Picture Experts Group (MPEG) of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) for use in object-based systems. Futhermore, BBC established VenueExplorer, object-based interactive audio system[13]. A *scene-based* system involves combining multiple microphones with directivity toward a sound arriving at a certain point, and the all sound from all directions is recorded. This reproduces the direction of the incoming sound. The Ambisonics sound pickup [6] uses a special microphone called an Ambisonic microphone, and the picked-up data are recorded as signals called B-format. Among them, an object-based system is suitable for presenting sound that follows the motion of the viewer and rotation of the head. It can calculate the perceived sound at the listening point from the relative position between the sound source to be reproduced and the viewer.

The Large-Scale Concept Ontology for Multimedia (LSCOM) [12] provides extensive semantic space for various news-broadcast video datasets with the goal that they be accessible to end users and provided in an auto-extractable framework. The Resource Description Framework (RDF) is widely used as a metadata description for open data, and networks with cross-referenced data constructed by URI are called linked open data (LOD). Furthermore, the vocabulary representing the relationships between data and the vocabulary representing types (classes) of data is called the RDF vocabulary. The RDF vocabulary describing music data is called as music ontology [15] and is widely used by the British Broadcasting Corporation (BBC), among others. In the studio ontology [5], it is possible to describe the process of producing hardware and music production, and to share this information with other producers. In addition, the audio effects ontology [18] is designed as an extension of music and studio ontologies. It can describe sound effects parameters, and can be shared in a system. These ontologies are for better music information retrieval.

3 OBJECTIVES AND REQUIREMENTS

With the facilitation of content distribution via the Internet, there are various types of content creation activities, including editing, processing, reconfiguration, reproduction, and so on. The number of people and devices involved in increasing alongside the value of content and is becoming more complex. In order to create high-quality content within such an environment, an efficient content management mechanism for object-based video and audio is required.

The purpose of this paper is to establish a mechanism for sharing video/audio recording data over the Internet and managing content that can be used by a third party as open and highly reusable video/audio data, where the data can describe the recording environment in detail. The mechanism is required to handle various interrelated pieces of information regarding images and sounds of the target, as well as location, direction, program and venue information, and recording processes.

In order to facilitate a variety of use cases that will emerge in the future, a shared infrastructure is required that can be used to develop a wide variety of subsystems in succession. SDM defines a standardized presentation method and structures video/audio media as a shared infrastructure such that it can interoperate with video/audio media. There are four data format requirements for managing viewing space data recorded using a large number of devices.

1. **Management of 3D video and audio media.** Storage in a specified 3D video and audio media format. The recorded data includes 3D location and target information, 3D location and recording device information, and related information created by the recording devices.
2. **Obtaining data according to the level required by the user.** The data can be acquired for a variety of purposes. The recorded data should include comprehensive information ranging from 'event name' to detailed information like 'position of the fifth microphone used for recording'. The database needs to be flexibly accessible to meet the needs of different users in each use case.
3. **Data must be stored in an application-independent format.** The data must be stored in a format that can be used to develop various applications. Applications using recorded data, showing interactive behavior with the user, package media mixing, semi-automatic mixing, or the like should be considered. It is necessary to store the data in a general-purpose format without specializing in a specific application so that the data can be used by as many applications as possible.
4. **Extensibility to support to new content.** The structure has extensibility that it is not restricted to specific content. It must have a data structure that can be flexibly expanded and

stored according to content from events other than music-related events, such as concerts and live events. It must also support traditional video/audio recording, sensor data, object recording in the virtual space, and so on for extensibility and media diversity in the future.

4 DATA SET

4.1 Concert at Keio University's Collegium Musicum Early Music Academy

On January 10, 2016, we recorded a concert of the Keio University Collegium Musicum Early Music Academy held at the Fujiwara Memorial Hall, which has 509 seats on the first and second floors on the Hiyoshi campus of Keio University. The recording targets were all acoustic performances including up to 24 musicians. Instruments include Theorbo, Cembalo, Baroque Violin, Baroque Oboe, and Viola da Gamba. The combined sound of was recorded, and individual instruments were recorded separately [9].

Figure 2 shows an arrangement of a cameras and a microphone. A unidirectional Schoeps MK4+CMC6 was hung on a stage baton near the Protanium at a position of 2 meters from the center of the stage as the main microphone. Two omnidirectional DPA4006 microphones were installed in each of the left and right side balconies as an ambience microphone, and an ORTF microphone was installed in the last seat on the main floor.

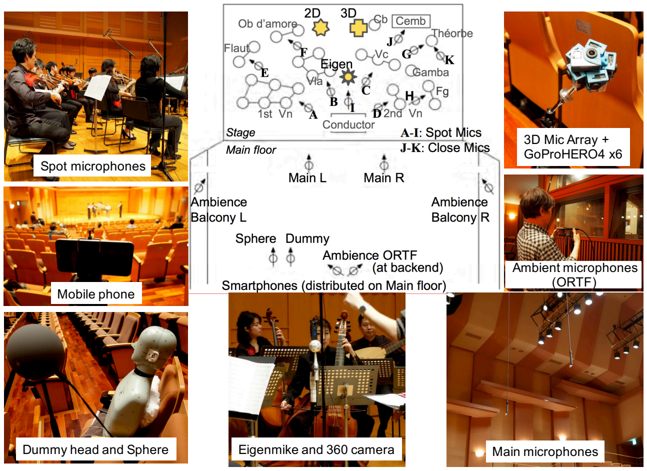

Figure 2: Arrangement of cameras and microphones at the orchestra recording

In addition, a proximity microphone for individual sound collection was installed using a unidirectional microphone for Theorbe and Cembaro, which are unique musical instruments with a weak acoustic output level. 3D XYZ and 2D 360 arrays using 6 and 8 unidirectional microphones, respectively, and Eigenmike™were installed as point array microphones. 3D and 2D audio were recorded in combination with three-dimensional video recordings with six GoPro™units, and two-dimensional video was recorded with a Ricoh Theta™ S camera, respectively. A dummy head microphone and a spherical microphone were also installed at representative points in the guest seat area. In addition, smartphones mounted on stands were installed at six points, including four main stories and two balconies. 2D and 3D microphone arrays, Eigen microphones, and smartphones recorded data to local storage. All other microphone outputs were transmitted over a head amp unit with Dante to DAWs on PCs installed in recording booths over IP-networks. In addition, 4k Video was simultaneously captured by cameras installed at the rear of the main floor.

4.2 Recording in Billboard Live Tokyo

We also recorded data at the Billboard Live Tokyo in Roppongi Midtown. The Billboard Live Tokyo has three floors, with table seats located on the third and fourth floors, and the sofa and casual seats located mainly on the fifth floor. The venue can accommodate 300 audience members.

Jazz band concerts were recorded on January 26, 2017. The band consisted of drums, electric bass, and a keyboard. On January 30, 2018, we recorded a concert given by J-pop artist Kariya Seira. The vocal is, the Singer Song Writer's 'Kariya Seira' with a drum, a keyboard, and a bus in the backband. Figure 3 shows the arrangement of the band, the installation location of the 360 degree camera, and the installation location of the microphone.

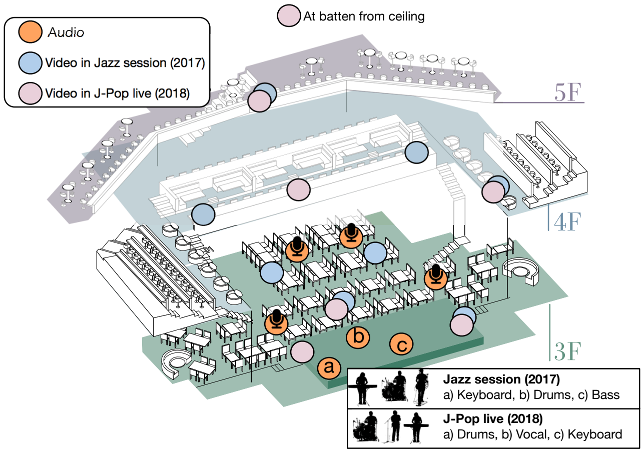
Audio
Video in Jazz session (2017)
Video in J-Pop live (2018)
At batten from ceiling
5F
4F
3F
a
b
c
Jazz session (2017)
a) Keyboard, b) Drums, c) Bass
J-Pop live (2018)
a) Drums, b) Vocal, c) Keyboard

Figure 3: Billboard Live recording (2017 and 2018)

The microphone arrangement is same for the 2017 and 2018 recordings. Four Audience microphones are installed in addition to the single musical instrument. Two microphones are installed at right and left stage sleeves, and two DPA4090 non-directional microphones were suspended from the ceilings. All the volume levels to be recorded depend on the mixing console settings. The recorded data was compiled into a single sound source for each musical instrument in an editing DAW, including the ambient mix in the left back, right back, left front, and right front of the audience seats.

The placement of the cameras was slightly changed between 2017 and 2018, and the 360-degree camera model was changed. During the jazz recording in 2017, eight Ricoh-manufactured global-sphere Ricoh Theta S cameras were placed at the locations shown in Figure 3 to record video data.

During the J-Pop concert recording in 2018, one Insta 360 Pro, four Garmin VIRB 360, and two Ricoh Theta V cameras were installed.

5 SOFTWARE DEFINED MEDIA ONTOLOGY

In this paper, we implemented an RDF description to structure 3D video/audio data and created LOD to make it accessible as open data. First, the SDM Ontology was defined in order to convert the recorded data described in Section 4 into LOD data.

5.1 Class Design

In order to make data production more convenient, improve processing and use functionality, and improve application flexibility, the SDM ontology must use data structures that are suitable for configuration of the recording object and the actual workflow state. In this paper, the recording space was constructed by following five elements and the 3D location information for audio/video data based on each element. (See Figure 4).

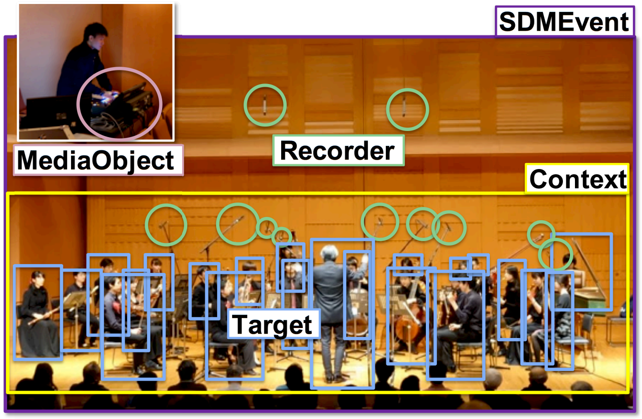

Figure 4: Five elements of the recording space

- **SDMEvent**: Information of the entire event (e.g., name and date of the event)
- **Target**: Information to be recorded (e.g., location information of the object, type of the object, etc.)
- **Context**: Information on the environment and situation (e.g., name and content of the program)
- **Recorder**: Recording device information (e.g., device location information, type)
- **MediaObject**: Media information (e.g., type of media)

SDMEvent represents the overall information of the event. **SDMEvent** has three elements: **Context**, **Target**, and **Recorder**. These are mutually referenced structures. **MediaObject** is a product produced by the recording operation.

As described above, the SDM ontology was designed so that the entire data to be recorded can be covered by decomposing the distributed recording space into five elements. In addition, in accordance with RDF, it is recommended that a pre-existing vocabulary be used as much as possible for data sharing and interoperability. Music Ontology[14], Event Ontology[19], and FRBRoo[2] were defined for music and events. However, they are not enough to describe the SDM recording system. In this paper, we adopted the schema.org[2] vocabulary for Web accessibility.

5.2 Ontology Classes

Construction of the SDM ontology is shown in Figure 5. Color coding in the figure represents group agreement for convenience in devising a data structure. In the following, we will describe classes and properties defined in the SDM ontology based on the five elements.

5.2.1 *SDMEvent*.

First, an **SDMEvent** is defined as a class representing the overall recorded target. Generally, this class is defined to record the title of the event of place, equipment, date, and time to designate the recording action and entirety of the recorded material. This class has the *has* property for indicating context representing programs, and we use the *schema:performer* vocabulary defined at shema.org. The physical place is described *schema:conentLocation* according to schema.org. In this paper, other classes also used vocabularies from schema.org.

5.2.2 *Target*.

Target class is defined to describe the performer, staff, audience, and so on. The properties are defined for a target class to describe the name and location of the recording target. The **Target** class has subclasses that depend on the type of target. **Musician**, **InstrumentPart**, **Orchestra**, **Audience**, and **Staff** classes are defined in this paper, which allow the type of data to be expanded. The **Target** class has a *name* property which describes the name of the target. The *local_x*, *local_y*, and *local_z* properties contain the location information in a coordinate system, and *local_t* contains the recorded time. Movement of the target can be determined from data corresponding to multiple locations. In addition, the *perform* property contains Context and *recorded_by* encodes the Recorder. The *attend* property in **SDMEvent** was also defined.

5.2.3 *Context*.

This class represents information in which the **Target** is situated. A typical example is a program that is structured on the event. If the event includes a plurality of programs, **SDMEvent** has multiple contexts. The **Context** will be used to represent the place and environment surrounding the recording activity. The **Context** class will have subclasses according to the type of context. Thus, the **Context** class is a super class that uses *schema:name* property for the context name, *time* encodes time information, and *geometry_definition* encodes the coordinate system and location information. *Geometry_definition*, *schema:performer*, and *recorded_by* are defined as common properties to describe the relation to the **SDMEvent/Target/Recorder**.

The **Program** class is defined as a subclass because the main target of this paper is a description of the performance data during a recording of a concert. The **Program** class is linked with the **Song** class in the *context* property. The **Song** class contains *composer*, *key*, and *tempo* properties. The ontology for describing music-related information will be integrated with related work [15]. The **Environment** class is defined as a **Context** subclass. This class

[2] http://schema.org/

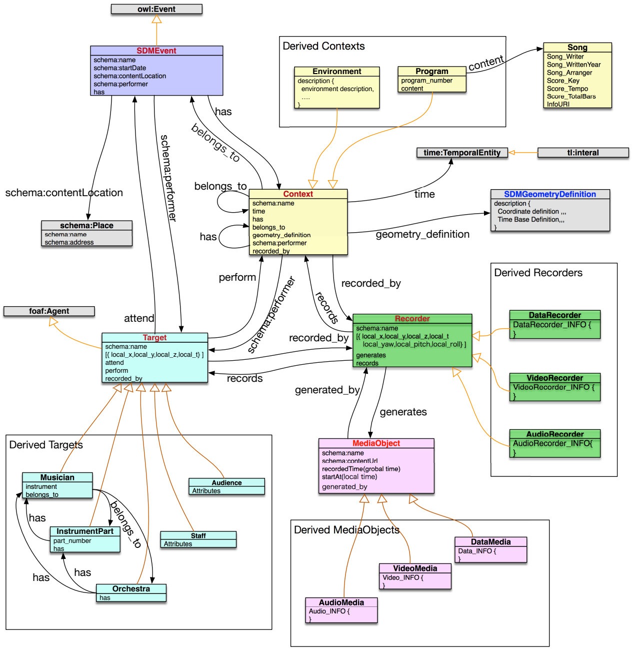

Figure 5: Design of the SDM ontology

will describe the facility, room, environment, atmosphere, and so on. **Context** is a class representing the recording event and the related environment. In other words, **Context** is defined to represent the intent of the **Recorder** combined with the **Target**.

5.2.4 ***Recorder***. The **Recorder** class contains recording equipment information, and three subclasses are defined for each set of data to be recorded. The **AudioRecorder** class is used for recording audio, the **VideoRecorder** class is used for recording images, and the **DataRecorder** class is used for recording other sensor data. Only the three subclasses are defined in this paper, but when a new data recording method appears in the future, the data recording method can be extended by defining the new data as a subclass of the **Recorder** class.

Recorder class has common properties, such as *schema:name* for name, *local_x*, *local_y*, and *local_z* for location information, *local_yow*, *local_pitch*, and *local_roll* for attitude, and *local_t* for time. The *generates* property is a property to link the **MediaObject** generated by the **Recorder** class. The **Recorder** class specifies **Target** and **Context** through the *records* property. The link shows the target and recording intent of the **Recorder** class. Typically, **Recorder** will link to **Musician** of the **Target** and to **Program**. When a recorder is linked to a recording of the overall orchestra rather than a specific musician, the **Recorder** class will link to **Orchestra**, which is a subclass of **Target** with the *records* property. When **Recorder** intends to record audience babble, the **Target** class will point to **Audience**. The **Recorder** class is able link to multiple **Targets** with the *records* properties.

The *records* property is a property defined to describe the intention to record, and can be used to provide extensive descriptions. The main feature of the SDM ontology is to describe the intention of recording by defining suitable subclasses of the **Target** and **Context** classes, and these are linked from the **Recorder** class with the *records* property.

5.2.5 ***MediaObject***. The **MediaObject** class is a class indicating a record generated by the **Recorder**. The **Recorder** class has classes for recording audio, video, and other as subclasses. Similarly, the **MediaObject** class has subclasses that respond to the **Recorder**. The audio media object is stored as a **AudioMedia** class, the video media object is stored as a **VideoMedia** class, and other media object is stored as a **DataMedia** class.

The **MediaObject** class has a link to the storage location of the recorded data as information, and the recorded data can be retrieved by tracing this link.

6 DESIGN AND IMPLEMENTATION

6.1 Design

The implementation approach is shown in Figure 6, where data recorded by recording devices in a distributed manner in the SDM Recording System is described as LOD using the SDM ontology. Next, the created LOD is stored in the SPARQL server shown in the center of Figure 6, and the server can be referred externally. This data is linked to the LOD cloud, and the SDM Edit/Processing system and SDM Playback System access the LOD cloud through SPARQL, obtain the required data, process, and reproduce.

The rough stage of implementation is as follows:

- Implementation of the SDM Ontology by ontology construction support tools, followed by publishing on the web.
- Describing the recorded data in the RDF/LOD format with the SDM ontology vocabulary.

- Publishing the LOD in SDM and access from applications.

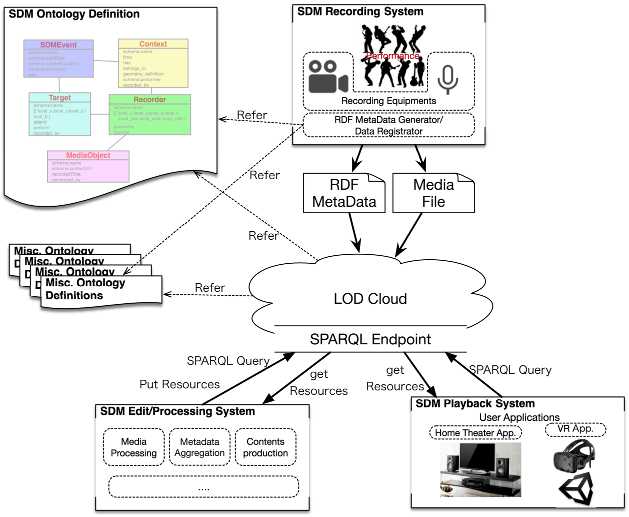

Figure 6: Approach for LOD in SDM

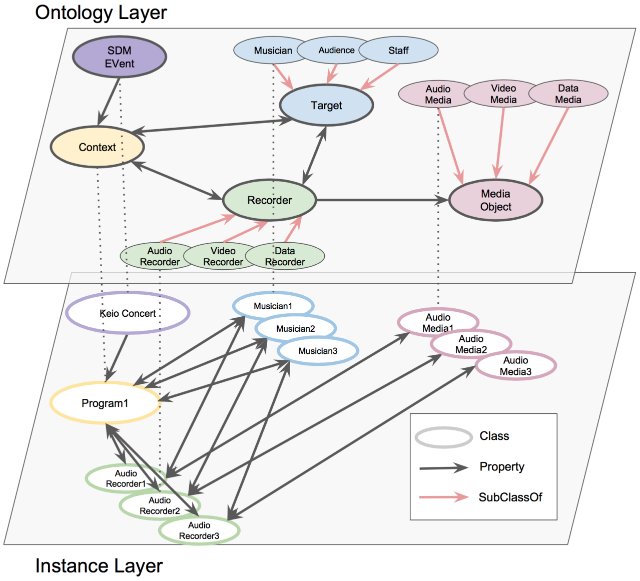

Figure 7: Ontology layer and Instance layer

6.2 Implementation of the SDM Ontology

In actually building the SDM ontology designed in Section 5, we used protégé [3], which is an ontology building support tool. It can be used to define the DataProperty and ObjectProperty classes, and the final ontology can be output in RDF or Turtle[4] format.

The ontology URI is

http://sdm.hongo.wide.ad.jp/sdmo/.

When referring to a vocabulary defined in the ontology, e.g., AudioMedia, the URL is

http://sdm.hongo.wide.ad.jp/sdmo/AudioMedia.

Web servers running sdm.hongo.wide.ad.jp are configured to automatically redirect to ontology files by referencing the vocabulary defined in the ontology.

6.3 LOD data implementation

When the video/audio data recorded in Section 4 is converted into an LOD, the recorded data can be described in an LOD format using Turtle.

Figure 7 shows a portion of the data structure from the Keio University orchestra concert described using the SDM ontology. Although some properties are omitted to avoid complicating the drawing, there are properties for referencing **Target** and **Recorder** from **SDMEvent** in the ontology layer, and **MediaObject** can also be referred to from the **Context** and **Target**. Classes related to tracks are also omitted.

The recorded Program was ’Keio University Collegium Musicum Early music Orchestra concert - The sounds of the German orchestra suite -’. **SDMEvent** links the **Program**. The **Program** has multiple **Musician** and **AudioRecorder** classes, and each class generates multiple **AudioMedia**. Note that the **Musician** and **AudioRecorder** referenced by the **Program** are not instances of their parent classes, i.e., **Target** and **Recorder** classes, but of their subclasses, i.e., concepts defined in the SDM ontology, have minimal properties, and their subclasses have unique properties, thereby enhancing scalability. Since the data handled in this paper are concerts data, **Musician** and **AudioRecorder** classes were generated. However, when another performance is described in the SDM Ontology, a suitable subclass will be generated and its instance will be created.

6.4 LOD publishing

After describing the data as an LOD using the SDM Ontology, the data is disclosed so that the data can be referred to from an external application. We built SPARQLEndpoint for SPARQL, which is an query language for LOD, and we uploaded the data. The data is then published and is accessible externally. As a result of a comparative study from the viewpoint of the number of supports in the RDF notation format and visualization of the data structure, we adopt GraphDB[5] as a SPARQL endpoint.

7 EVALUATION IN APPLICATIONS AND USECASES

7.1 Evaluation policy

The recorded data in Section 4 is managed by this LOD, and we created an application using these data. We show that the developed LOD was sufficient for use in various applications. In addition, we show some use case scenarios using these LODs. Thereafter, the availability of the data is high by indicating that the data satisfying these objectives can be acquired.

[3] https://protege.stanford.edu/

[4] https://www.w3.org/TR/turtle/

[5] https://ontotext.com/products/graphdb/

7.2 SDM applications

7.2.1 ***SDM360^2: Tablet device application.*** We developed an application called SDM360^2 that is capable of interactively playing back 3D content in any perspective by using the recorded data described in Section 4. SDM360^2 can be used to listen to music at any position, and Figure 8 show a user listening to music from the orchestra stage. Moving the tablet from side to side moves the line of sight, user can changes his/her perspective. The sound recorded for each musical instrument is displayed as Audio Objects, and the user can listen to sound from specific instruments by enabling or disabling them.

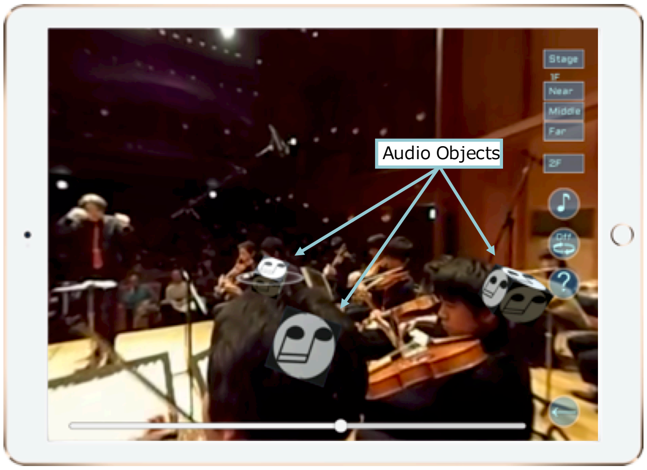

Figure 8: Screenshot of SDM360^2

SDM360^2 uses the SDM ontology to acquire the three-dimensional position of audio or video objects. During the startup process, a SPARQL query is sent as an HTTP GET query parameter to the REST API of the SPARQL endpoint provided by GraphDB. The Content-Type class in encoded in the HTTP headers, and the result is sent in the specified format. Since SDM360^2 is implemented in Unity, HTTP communication was performed using UnityWebRequest.

In the following examples, musical instruments and position data of the performers can be acquired by requesting the following SPARQL queries from the application. The part names, part numbers, and position data of the Musician who participated in the SDM recording are acquired from the queries. The application creator also synchronizes the data by requesting this query within the application, thus reducing the labor involved in manually placing the object.

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

select ?part ?part_num ?x ?y ?z
where{
    ?event a sdmo:SDMEvent ;
           s:name 慶應義塾大学コレギウム・ムジクム古楽オーケストラ
                演奏会ドイツ管弦楽組曲の響き"--"@ja ;
           s:performer ?s .
    ?s a sdmo:Musician ;
        sdmo:part_name ?part ;
        sdmo:part_number ?part_num;
        sdmo:local_x ?x ;
        sdmo:local_y ?y ;
        sdmo:local_z ?z .
}
```

7.2.2 ***LiVRation: VR application.*** LiVRation can provide a stereophonic environment in an immersive environment using an HMD. In addition, streaming reproduction of high-resolution audio is realized using MPEG4 ALS[10]. The viewer can use the HMD to experience the virtual space at the viewing position. It is also possible to extract only the sound of a particular musical instrument, display Twitter information, and feel vibration from vibration transmission devices. Figure 9 shows the application in operation. A 360° moving image or sound recorded from a spherical object was mapped and is displayed on the screen. If a Video Object is selected with the previous controller and the trigger is pulled, you can view a live stream from that location. In addition, selection of an Audio Object on the controller enables, disables, or adjusts the volume of the controller. Twitter messages are also displayed to surround the viewer.

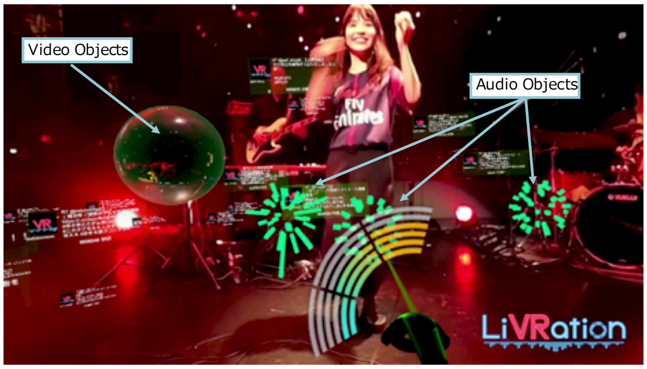

Figure 9: Screenshot of LiVRation

Metadata of SDM objects, such as sound sources, performers, location, etc., are obtained from GraphDB, which is a SPARQL endpoint built on the cloud. This data is sent to LiVRation clients via the services layer. When the service platform is accessed via HTTP from Unity during client initialization, the service platform accesses GraphDB with a SPARQL query prepared in advance, and the result is sent to the client. Therefore, the acquired metadata is passed to a rendering module. Part of the ontology defined for LiVRation is shown below.

```
@prefix sdmo: <http://sdm.hongo.wide.ad.jp/sdmo/> .
@prefix sdm: <http://sdm.hongo.wide.ad.jp/resource/> .

### Define SDMEvent
sdm:hackasong rdf:type sdmo:SDMEvent ;
  s:contentLocation sdm:billboard ;
  s:name "Billboard LIVE HACKASONG 2017"@ja ;
  s:startDate "2018/1/30 17:30" ;
  sdmo:has sdm:song1 ;
  sdmo:recording sdm:bass_mic ,
                 sdm:drum_mic ,
                 sdm:far_left_mic ,
                 sdm:far_right_mic ,
                 sdm:keyboard_mic ,
                 sdm:near_left_mic ,
                 sdm:near_right_mic ,
                 sdm:vocal_mic .
```

```
###  Define Context
sdm:song1 rdf:type sdmo:Song ;
  s:name "Colorful World" ;
  sdmo:performed_by sdm:bass ,
                    sdm:drum ,
                    sdm:keyboard ,
                    sdm:vocal ;
  sdmo:program_number "1" ;
  sdmo:recorded_by sdm:bass_mix ,
                    sdm:drum_mix ,
                    sdm:far_left_mix ,
                    sdm:far_right_mix ,
                    sdm:keyboard_mix ,
                    sdm:near_left_mix ,
                    sdm:near_right_mix ,
                    sdm:vocal_mix .

###  Define Target
sdm:vocal rdf:type sdmo:Musician ;
    s:name 仮谷せいら""@ja ;
    sdmo:attend_to sdm:hackasong ;
    sdmo:localX "0.0"^^xsd:float ;
    sdmo:localY "0.5"^^xsd:float ;
    sdmo:localZ "4.5"^^xsd:float ;
    sdmo:perform sdm:song1 .
```

7.3 Other Usecases

7.3.1 ***minus one content.*** One of our envisioned data use methods is to help orchestras rehearse and practice. Specifically, the audio is emphasized or suppressed in the content. The latter is generally called 'minus one' content and is useful for practicing with musical instruments.

A sound source excluding a particular part and other data are subtracted as patterns for using SDM data in the minus one content. A list of sound sources can be obtained with the following SPARQL queries, where the violin, viola, and **Musician** information are excluded.

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

select ?source ?p ?o
where{
    ?target a sdmo:Musician;
            sdmo:instrument ?inst ;
            ?p ?o.
    ?media a sdmo:AudioMedia ;
        sdmo:records ?target ;
        s:contentUrl ?source .
    filter(?inst not in ("Violin", "Viola")) .
}
```

7.3.2 ***Getting the event related data from outside link.*** Since all resources are represented by URI, any LOD can be cross-referenced with another LOD on the web. By tracing the links, data can be acquired from an external LOD.

SDM LOD refers to Wikidata[6] resources to obtain the composer of the song played at the event. Wikidata is a project in which data provided by Wikipedia is structured and published, and the URI can be referred to from an external LOD. Since the SPARQL endpoint is also open to the public, it can be accessed through the SPARQL endpoint of SDMs.

SPARQL queries for acquiring a list of composers of songs played at specified events and composer information are shown below. Descriptive data of the composer is not included in the LOD disclosed by the SDM, and this data can be obtained by tracing links to a Wikidata referred to inside the LOD.

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

select ?song_writer ?p ?o
where{
    ?event s:name 慶應義塾大学コレギウム・ムジクム古楽オーケストラ
        演奏会ドイツ管弦楽組曲の響き"--"@ja ;
        sdmo:has ?context.
    ?context a sdmo:Song ;
             sdmo:song_writer ?song_writer.
    service <https://query.wikidata.org/bigdata/namespace
        /wdq/sparql>
    {
        ?song_writer ?p ?o.
    }
}
```

7.3.3 ***Integration of recorded data at multiple points.*** We assume a distributed recording environment, where more than one event is recorded with a set of distributed devices. It is also assumed that one new piece of content is created by synchronizing and integrating the content recorded from multiple points.

Queries for acquiring the list of files for all **MediaObjects** recorded in the respective events and instrument data for the recorded **Target** with respect to the same musical composition played in the respective events are as follows:

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

select ?event_name ?file ?instrument
where{
    ?event a sdmo:SDMEvent ;
           s:name ?event_name ;
           sdmo:has ?song.
    ?song a sdmo:Song ;
          s:name "Le journal du printems No.4" ;
          sdmo:recorded_by ?media.
    ?media s:contentUrl ?file ;
           sdmo:records ?target.
    ?target a sdmo:Musician ;
            sdmo:instrument ?instrument.
}
```

8 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a framework in which each recorder freely shares recorded data, and a third party creates an application based on this data. A content management mechanism for sharing recorded object-based video and audio data via the Internet by SDM ontology is also presented. We have implemented this administrative mechanism, and we have developed and validated applications using a tablet and VRs that are capable of interactively playing 3D content from any perspective. LOD data using the SDM ontology could be used to provide the appropriate data for various applications.

The three-dimensional position of the microphone and camera was manually described. As a future challenge, recording environments should be described automatically by using three-dimensional positioning techniques in order to more easily utilize the SDM ontology.

[6] https://www.wikidata.org/

REFERENCES

[1] 2015. AUROMAX© Next generation Immersive Sound system. (November 2015). http://www.auro-3d.com/

[2] 2015. Definition of FRBROO, A Conceptual Model for Bibliographic Information in Object-Oriented Formalism. https://www.ifla.org/files/assets/cataloguing/FRBRoo/frbroo_v_2.4.pdf. (2015).

[3] 2015. *Dolby Atmos© Specifications*. Technical Report Issue 3. Dolby Laboratories.

[4] R. Bleidt, A. Borsum, H. Fuchs, and S. M. Weiss. 2015. Object-Based Audio: Opportunities for Improved Listening Experience and Increased Listener Involvement. *SMPTE Motion Imaging Journal* 124, 5 (July 2015), 1–13. https://doi.org/10.5594/j18579

[5] György Fazekas and Mark B Sandler. 2011. The Studio Ontology Framework.. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. 471–476.

[6] Michael A Gerzon. 1973. Periphony: With-height sound reproduction. *Journal of the Audio Engineering Society* 21, 1 (1973), 2–10.

[7] Kimio Hamasaki, Kentaro Matsui, Ikuko Sawaya, and Hiroyuki Okubo. 2011. The 22.2 multichannel sounds and its reproduction at home and personal viroment. In *Audio Engineering Society Conference: 43rd International Conference: Audio for Wirelessly Networked Personal Devices*. Audio Engineering Society.

[8] J. Herre, J. Hilpert, A. Kuntz, and J. Plogsties. 2015. MPEG-H 3D Audio —The New Standard for Coding of Immersive Spatial Audio. 9, 5 (Aug. 2015), 770–779. https://doi.org/10.1109/JSTSP.2015.2411578

[9] Masahiro Ikeda, Takuro Sone, Kenta Niwa, Shoichiro Saito, Manabu Tsukada, and Hiroshi Esaki. 2016. New Recording Application for Software Defined Media. In *Audio Engineering Society Convention Paper, 141st AES Convention*. Los Angeles, USA.

[10] T. Liebchen and Y. A. Reznik. 2004. MPEG-4 ALS: an emerging standard for lossless audio coding. In *Data Compression Conference, 2004. Proceedings. DCC 2004*. 439–448. https://doi.org/10.1109/DCC.2004.1281489

[11] E. Nakasu. 2012. Super Hi-Vision on the Horizon: A Future TV System That Conveys an Enhanced Sense of Reality and Presence. *IEEE Consumer Electronics Magazine* 1, 2 (April 2012), 36–42. https://doi.org/10.1109/MCE.2011.2179821

[12] Milind Naphade, John R Smith, Jelena Tesic, Shih-Fu Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. 2006. Large-scale concept ontology for multimedia. *IEEE multimedia* 13, 3 (2006), 86–91.

[13] Melchior F Paradis M, Gregory-Clarke R. 2015. VenueExplorer, object-based interactive audio for live events. In *Proceedings of the 1st web audio conference (WAC ’ 15) CEUR workshop proceedings*.

[14] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. 2007. The Music Ontology. *Proceedings of the International Conference on Music Information Retrieval* (2007), pp. 417–422.

[15] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. 2007. The Music Ontology. *Proceedings of the International Conference on Music Information Retrieval* (2007), pp. 417–422.

[16] ITUR Rec. 2014. ITU-R BS. 2051-0 (02/2014) Advanced Sound System for Programme Production. *Int. Telecommun. Union, Geneva, Switzerland* (2014).

[17] Manabu Tsukada, Keiko Ogawa, Masahiro Ikeda, Takuro Sone, Kenta Niwa, Shoichiro Saito, Takashi Kasuya, Hideki Sunahara, and Hiroshi Esaki. 2017. Software Defined Media: Virtualization of Audio-Visual Services. *IEEE International Conference on Communications (ICC2017)* (May 2017). Paris, France.

[18] Thomas Wilmering, György Fazekas, and Mark B Sandler. 2013. The Audio Effects Ontology.. In *International Society for Music Information Retrieval (ISMIR 2013)*. 215–220.

[19] Raimond Y and Abdallah S, A. 2006. The event ontology. http://purl.org/NET/c4dm/event.owl. (2006).