future internet

MDPI

Article

An Ontology for Spatio-Temporal Media Management and an Interactive Application

Takuro Sone [1,*], Shin Kato [2], Ray Atarashi [3], Jin Nakazato [2], Manabu Tsukada [2] and Hiroshi Esaki [2]

[1] Graduate School of Science and Technology, Shizuoka University, Hamamatsu 432-8011, Shizuoka, Japan
[2] Graduate School of Information Science and Technology, Department of Creative Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan; shin@hongo.wide.ad.jp (S.K.); jin-nakazato@g.ecc.u-tokyo.ac.jp (J.N.); mtsukada@g.ecc.u-tokyo.ac.jp (M.T.); hiroshi@wide.ad.jp (H.E.)
[3] Internet Initiative Japan Inc., Chiyoda-ku, Tokyo 102-0071, Japan; ray@iij.ad.jp (R.A.)
* Correspondence: t-sone@kitanilab.org

Abstract: In addition to traditional viewing media, metadata that record the physical space from multiple perspectives will become extremely important in realizing interactive applications such as Virtual Reality (VR) and Augmented Reality (AR). This paper proposes the Software Defined Media (SDM) Ontology designed to describe spatio-temporal media and the systems that handle them comprehensively. Spatio-temporal media refers to video, audio, and various sensor values recorded together with time and location information. The SDM Ontology can flexibly and precisely represent spatio-temporal media, equipment, and functions that record, process, edit, and play them, as well as related semantic information. In addition, we recorded classical and jazz concerts using many video cameras and audio microphones, and then processed and edited the video and audio data with related metadata. Then, we created a dataset using the SDM Ontology and published it as linked open data (LOD). Furthermore, we developed "Web360^2", an application that enables users to interactively view and experience 360° video and spatial acoustic sounds by referring to this dataset. We conducted a subjective evaluation by using a user questionnaire. Web360^2 is a data-driven web application that obtains video and audio data and related metadata by querying the dataset.

Keywords: content management; ontology; resource description framework; digital media; audio visual; proof-of-concept; virtual reality

Citation: Sone, T.; Kato, S.; Atarashi, R.; Tsukada, M.; Esaki, H. An Ontology for Spatio-Temporal Media Management and an Interactive Application. *Future Internet* **2023**, *15*, 225. https://doi.org/10.3390/fi15070225

Academic Editors: Konstantinos Kotis and Christos Goumopoulos

Received: 2 June 2023
Revised: 16 June 2023
Accepted: 17 June 2023
Published: 23 June 2023

1. Introduction

Most of the visual and audio content that we interact with daily, such as movies, TV programs, and music, is communicated using digital media. After being recorded as digital media on a recording device, it is transmitted, distributed, and played back on a playback device to provide visual and auditory experiences. Although many standards and specifications [1–5] exist for digital media, video and audio media representations are highly restricted because they are created for playback on widely used playback devices. The basic playback style of video is a single rectangular screen, and most of the audio content is stereo audio or channel-based audio with fewer than ten channels [6]. Although, there are various factors in the representation of digital media, such as video resolution, pixel representation, refresh rate, audio code length, sampling rate, and the number of channels, this representation is only a variation of the basic style of the rectangular screen plus channel-based audio. In addition, when playing back video and audio content encoded on digital media, normal playback devices may fine-tune the picture and sound quality, but they do not make any changes to the content. Users can only passively view video and listen to audio content that has been edited according to the creator's intention.

Meanwhile, the use of VR images using computer graphics and 360° images, spatial sound reproduction, and sound reproduction using binaural recordings [7], among other innovations, has been spreading, and such audio-visual content with a high sense of

realism is becoming increasingly popular [8–11]. Typical VR applications allow users to view images from any viewpoint in a virtual space while the localization of the sound changes according to the viewpoint to provide a highly realistic viewing experience. To reproduce images from arbitrary viewpoints, it is necessary to use computer graphics synthesis or a combination of many video camera images [12]. At the same time, it is also important to reproduce sound according to the user's position and posture in order to provide a high sense of realism [13]. Since video and audio content is not processed or edited in a general playback environment, the editor produces video and audio content as a finished product optimized to most effectively reproduce the editorial intent in a general playback environment. On the other hand, to reproduce an interactive viewing experience, video and audio must be processed. In this sense, the playback device can be considered responsible for the final stage of the editing process. Therefore, digital media content that is one step behind the finished product and includes the information necessary for this operation is required. In addition to video and audio, displays for various senses such as touch, force, smell, and taste are also being attempted to reproduce a higher sense of presence [14,15]. For this purpose, it would be convenient to have a system that can handle video and audio media as well as data collected by various sensors in a unified manner.

There is still no digital media framework that can comprehensively meet the above requirements. Therefore, we believe that a new digital media management framework is needed that is suitable for providing interactive and highly realistic media playback environments that are expected to become popular in the near future. Hence, we established a Software Defined Media (SDM) consortium (https://sdm.wide.ad.jp/ (accessed on 1 June 2023)) in 2014 to promote research on Internet-based video and audio media [16]. The main research objective of the SDM consortium is to develop new ways of representing digital media by managing digital media in object units and controlling them with software, and at the same time, build an architecture that can realize flexible and highly applicable systems and services. The SDM consortium has been engaged in various research activities, such as digital recording media, managing related metadata, and exploring media expression and production methods using such data in virtual spaces or by combining virtual and real spaces.

In 2016, the SDM consortium launched a new project for use on platforms for managing three-dimensional (3D) video and audio media called the SDM Ontology [17] referred to as the SDM Ontology Version 1. Digital media production can be divided into four stages: recording, editing/processing, distribution, and playback, as shown in Figure 1. In the recording stage of digital media, media data containing video and audio are generated, and at the same time, various metadata about the recording environment and content are generated. SDM Ontology Version 1 defines a vocabulary focused on metadata descriptions during the recording stage.

In this study, we propose SDM Ontology Version 2, which can comprehensively describe metadata from the recording, editing/processing and playback phases by organizing and improving the structure of SDM Ontology Version 1. Figure 1 shows the difference in scope between SDM Ontology version 1 and version 2 (Hereafter, when we refer to SDM Ontology, we refer to SDM Ontology Version 2). However, there is no established method of expression or management for the various metadata generated in the recording and editing process, and each digital media creator must manage their metadata. These metadata are generally described in a form unique to each producer or in a format specific to the equipment or software used, making it difficult for third parties to use them. Consequently, most of the digital media that can be distributed in general are limited to data in the form of audio and visual media data after editing has been completed. We developed the SDM Ontology to structurally describe the metadata necessary for a series of digital media production processes using a precisely semantically defined vocabulary. As a result, it can be expected that independently recorded and edited digital media can be referenced and reused, promoting the division of labor in digital media production. New digital content emerges through the participation of various content creators.

The remainder of this paper is organized as follows. Section 2 introduces related research, including existing ontologies that provide music-related Resource Description Framework (RDF) vocabularies, 3D audio technology, and existing interactive applications that handle 3D video and audio. Section 3 presents the design policy and structural outline classes of the SDM Ontology proposed herein. Section 4 describes the dataset used in this study and its recording environment. Section 5 describes the design and implementation of Web360^2, an interactive application based on the SDM Ontology. Sections 6 and 7 describe the evaluation of the SDM Ontology and Web360^2, respectively. Finally, in Section 8, we discuss and conclude the study.

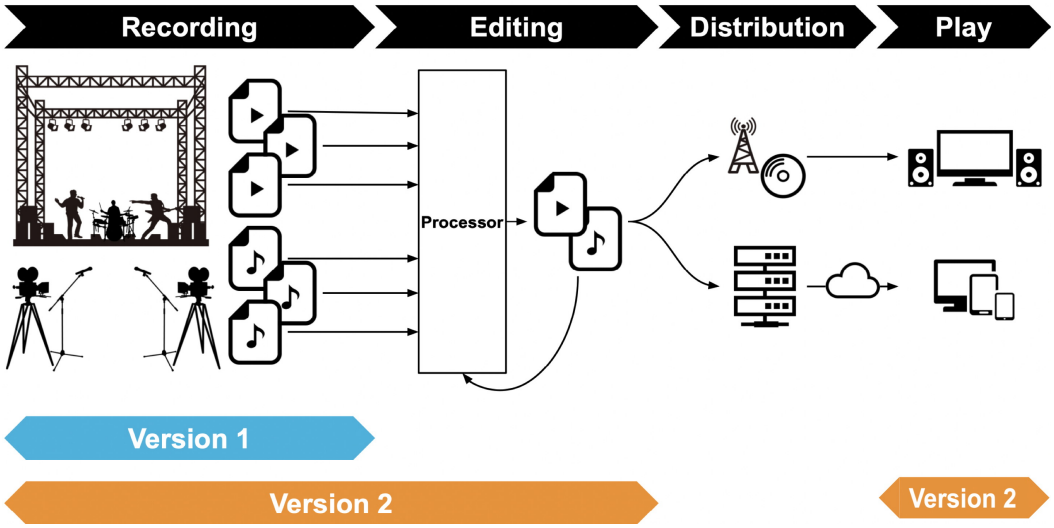

Figure 1. Overview of digital media production flow and scope of the SDM Ontology description.

2. Related Works

The SDM Ontology research began with the description of musical events. In this section, we describe the existing ontologies that provide music-related RDF vocabularies and then explain the spatial acoustic technologies that are effective for developing applications using 3D images and sounds.

2.1. Related Ontologies and Studies

The Music Ontology [18,19] is a unified framework that describes music-related information. The Music Ontology is a combination of the Timeline Ontology [20], Event Ontology [21], and International Federation of Libraries (IFLA) Functional Requirements for Bibliographic Records (FRBR) Ontology [22], Friend of a Friend (FOAF) [23], etc. It is based on existing ontologies, can describe artist names and song titles and represents each stage from composition to performance, recording, and release. Furthermore, the Music Ontology does not represent technical aspects of the music production flow, unlike the Studio Ontology [24] represents. The Studio Ontology enables the description of a series of audio recording and editing flows, including microphone placement, wiring connections between devices, mixing, and signal processing. In particular, a more detailed description of audio effects has been extended and defined as the Audio Effect Ontology [25,26].

Meanwhile, the following are studies on building an ontology for instrument classification. Implementation experiments were conducted using a prototype ontology based on two different instrument classification methods [27]. A study was performed to identify and group instruments based on their timbre characteristics and to generate a hierarchical instrument ontology automatically [28]. Several ontologies work with the Music Ontology. The Audio Features Ontology [29,30] provides a description of the computational workflow for audio feature extraction and a representation structure for various feature data formats.

Another highly abstract ontology is the Audio Commons Ontology [31,32]. The acoustics, musical compositions, and synthetic sounds used by various online services are stored in repositories on the Web. The Audio Commons Ontology was designed to facilitate the repository integration of audio content and access to the repository from client applications. The computing device network that handles data valid for music is called the Internet of the Musical Things (IoMusT), analogous to the Internet of Things (IoT). The Internet of Musical Things Ontology [33,34] describes the IoMusT ecosystem.

In addition, standards and related studies for general spatio-temporal media and data are summarised below. MPEG-7 [35] (https://mpeg.chiariglione.org/standards/mpeg-7 (accessed on 1 June 2023)) is a metadata representation standard for describing semantics information of multimedia content. MPEG-7 uses the Description Definition Language (DDL) to describe semantic information. DDL can represent a wide range of semantic information ranging from media data and the content expressed in media data to related contexts. The Multimedia Ontology (MMO) is an ontology that effectively solves the problem of the "semantic gap" between the perceptible media world and the advanced conceptual world [36]. The Multimedia Web Ontology Language (MOWL) [37] is an ontology for reasoning about media properties and potential concepts in multimedia applications. The MOWL employs the representation of spatio-temporal relations proposed in [38] to extend the MPEG-7 query language. Chan et al. [39] showed that object motion could be recognized from low-level video information by mapping vocabulary and object motion using an ontology mapping technique. Duckham et al. [40] designed and implemented a prototype of an automated spatial reasoning system for geospatial information applications called NEXUS and showed that it could perform a variety of spatio-temporal queries and also explain why it reached the conclusions it did.

2.2. Spatial Acoustic Technology

Various methods have been proposed for spatial acoustic technology. MPEG-H, which is currently being standardized by the International Organization for Standardization (ISO) and the Moving Picture Experts Group (MPEG), a working group of the International Electrotechnical Commission (IEC). MPEG-H Part 3 includes a section on 3D audio [41]. The three spatial acoustic methods allowed by MPEG-H 3D audio are channel-based, object-based, and HOA (higher-order ambisonics) [42]. Channel-based audio records and edits acoustic data tied to the playback environment based on information regarding the number of speakers and their position relative to the viewer. When moving in an acoustic space, it is necessary to prepare other acoustic data associated with the new environment or modify the original acoustic data. Therefore, expressing sound movements following the viewer's actions is challenging and lacks flexibility and interactivity.

On the other hand, in object-based audio, the 3D location information of the sound source is recorded as metadata and acoustic data. Flexible and interactive sound reproduction is possible by rendering the acoustic data according to the positional information of the sound source and the listener and the environment of the playback space. The main applications are Dolby Atmos [43,44] and DTS:X [45], which have been introduced in movie theaters and home theaters. AuroMax [46] employs object-based audio or combines it with conventional channel-based audio.

Accessing each acoustic object in the sound field remains challenging [47] because spatial information in HOA/ambisonics is not explicit geometric metadata, as it is in object-based audio. Meanwhile, the recorded data reproduces a spherical sound field centered on the microphone and thus can easily follow the rotation of the listener's head. This characteristic is similar to that of a video recorded by a 360° camera. Recently, consumer-use 360° cameras such as the Ricoh THETA Z1 and THETA V [48] have been equipped with ambisonic microphones. YouTube and Facebook's 360° video services also support HOA/ambisonics.

2.3. Interactive Application Using 3D Video and Audio

The SDM consortium's previous study developed SDM360[2], a playback application for recorded content using a tablet device, and developed LiVRation, an application for viewing live-streamed content on an HMD (Head Mounted Display) [49]. The 360° videos were projected in a virtual space, where the acoustic objects were mapped based on object-based audio techniques. The video and audio clips followed the viewer's head movement. The viewer can also move within the virtual space by manipulating the screen. In addition, by utilizing the fact that each sound source object is independent, interactive 3D content playback was realized by switching on/off and adjusting the volume for each sound source. Because we implemented both functions using Unity, a Unity runtime environment is required for development and execution. The 360° video and audio data used by this application, as well as the metadata representing the video and audio objects, were embedded in the application.

3. The Software-Defined Media Ontology

Events in real space can be represented as digital media with spatio-temporal information by recording, processing, and editing. This paper calls such digital media spatio-temporal information spatio-temporal media. Spatio-temporal media can be treated as objects in a virtual space. These objects can be reproduced freely in real space using a playback system. The SDM Ontology is an ontology designed to describe such spatio-temporal media and the process (workflow) of recording, processing, editing, and playback, and describes related semantic information comprehensively. In our previous proposal [17], SDM Ontology Version 1 as shown in Figure 1, we proposed a system design that enables descriptions of the environment, contents, and objects in the recording stage of spatio-temporal media. This paper describes SDM Ontology Version 2, which is designed to convey the recording stage and the editing, processing, and playback stages of spatio-temporal media by extending the proposal of SDM Ontology Version 1 and organizing and expanding the vocabulary structure and naming conventions.

3.1. Design Policies

To achieve these goals, we established the following design policies:

1. **Description of spatio-temporal media processing workflow**
   It is not sufficient to describe only spatio-temporal media to specify the workflow of recording, processing/editing, and playback of spatio-temporal media. It is necessary to describe the equipment and functions involved in each step of the workflow, as well as their setting information and operation details, after clarifying the input-output relationship of spatio-temporal media.
2. **Semantic description with extensibility**
   Each spatio-temporal media and each workflow stage has its semantic information with some intention or purpose. This semantic information is fundamental and indispensable to express spatio-temporal media, but at the same time, it is not easy to define the scope of expression. Since it is not realistic to comprehensively specify this information in the SDM Ontology alone, it is appropriate to utilize external ontology to describe such information. Meanwhile, if unlimited descriptions by external ontology are allowed, the rigor of the semantic description will be compromised. It is necessary to have a system that allows descriptions utilizing external ontology in a format consistent with the design intent of the SDM Ontology.
3. **Hierarchical object representation**
   Spatio-temporal media recording is not only performed with a single piece of equipment, but with a combination of equipment. As the recording scale becomes more extensive and more people are involved, the equipment becomes more complex. The types and composition of the recorded media data also become more complex, and at the same time, the semantic information associated with the data also becomes more complex. The same situation occurs in the processing, editing, and playback phases.

To cope with such a situation, it is appropriate to group devices, functions, and accompanying semantic information related to spatio-temporal media processing as necessary and represents them as hierarchical composite objects. At the same time, the spatio-temporal media itself should be described as a composite object that groups various types of spatio-temporal media and hierarchically expresses them.

4. **Spatio-temporal coordinate representation**
   Spatio-temporal coordinates are essential information in representing objects that comprise spatio-temporal media. Therefore, spatio-temporal information should be represented as a fundamental component of the SDM Ontology. Considering the diversity of objects to be handled, the spatio-temporal information they contain is also very diverse. Therefore, the SDM Ontology should be able to describe spatio-temporal information flexibly using various coordinate systems.
   For example, it would be natural to represent the positional information of microphones and cameras in a concert recording in the local coordinate system of the stage. Meanwhile, the location information of a concert venue is generally expressed in terms of latitude and longitude. Similarly, the time axis of video media recorded by a video camera at a concert is generally set to the origin of the recording start time. However, it is necessary to convert the origin of the time axis to correspond to the time in the concert program. Furthermore, when strict time management is required, the error of the built-in clock of the video camera may be a problem. In such cases, a mechanism to express the correspondence between the time axis in the generated video image data and the accurate time axis is required. Therefore, the spatio-temporal information must be a description that can accommodate coordinate transformation and accuracy compensation processing in the processing and editing phases.

3.2. Design Overview

The following is an overview of the SDM Ontology design. The SDM Ontology defines the media, recorder, processor, player, and context as the primary objects that constitute spatio-temporal media. Five classes, the **Media**, **Recorder**, **Processor**, **Player**, and **Context** are defined to express the above objects. These are called the primary classes in the SDM Ontology. The **Recorder**, **Processor**, and **Player** classes describe the devices and systems that perform the recording, editing/processing, and playback. In contrast, the **Media** class describes the spatio-temporal media data. The **Context** class is the primary class for the general metadata expression to provide semantic information for the primary class objects. The **Context** class is used not only to describe the semantic information defined in the SDM Ontology but also to describe objects that serve as relay nodes for linking various metadata related to external ontologies. Since spatio-temporal information is important metadata to be handled in the SDM Ontology, it is described using the **Geometry** class and **CoordinateSystem** class separately from the **Context** class. The **Geometry** class represents the spatio-temporal information of the primary class object. The **CoordinateSystem** class is used to specify the coordinate system of the spatio-temporal coordinates described in the **Geometry** class object.

Figure 2 shows the structural outline of the SDM Ontology. Each node in the figure represents a class, with red and blue letters representing the primary and other classes, respectively. The vocabulary written within the nodes represents the data properties, and the black arrows connecting the nodes represent the object properties. The following subsections describe each class in detail. In the following text, "**class-name** + object" means "instance data described according to the **class-name** class definition".

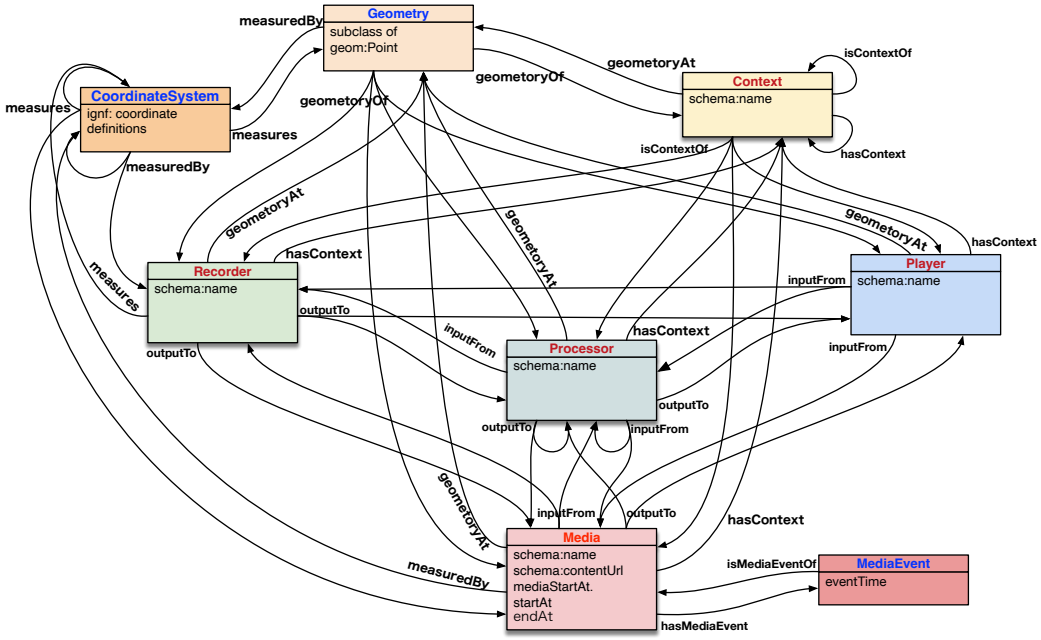

Figure 2. SDM Ontology core structure.

3.3. Description of Class Definitions

3.3.1. **Context** Class

The **Context** class is a class for comprehensively describing the various metadata of the SDM Ontology primary class.

Figure 3 shows the structure of the **Context** class and its subclasses. In the figure, the white arrows indicate a **Context** class in which the source node is a subclass and the destination node is a superclass. The vocabulary associated with the arrows between the classes is an object property. The **hasContext** and **isContextOf** properties can be used to associate the **Recorder**, **Processor**, **Player**, **Media**, and **Context** objects with **Context** objects. When associating the **Recorder** and **Player** objects with the **Target** and **Content** objects, the **records**, **recordedBy**, **plays**, and **playedBy** properties express the intention more clearly.

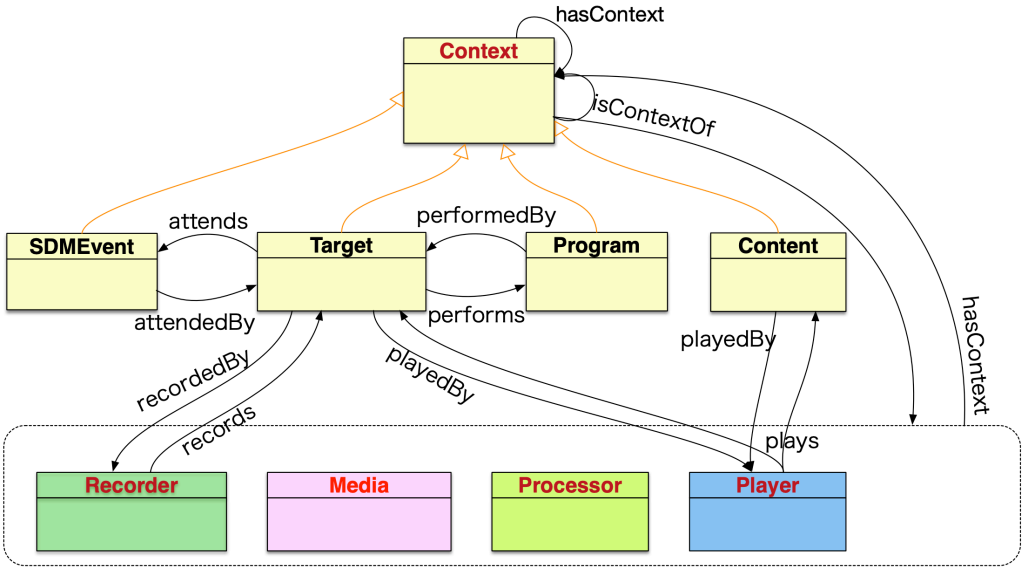

Figure 3. Context class structure and related classes.

As described in Section 3.1, it is difficult to define the scope of metadata representation. The SDM Ontology specifies only the following four classes as subclasses of the **Context** class, which are considered necessary to express the relationships among the classes defined in the SDM Ontology.

1. **SDMEvent** subclass:
   This subclass represents and expresses the entire recording act. Typically, it includes the whole recording of an event, such as a concert or sporting event. By describing **SDMEvent** in a nested structure, events can be divided and expressed hierarchically according to the content creator's intent. It is also possible to integrate and express multiple events.
2. **Target** subclass:
   This subclass represents the recording target. The recording target represents not only physical objects such as performers, instruments, instrument parts, and orchestras in a concert recording but also abstract objects such as the recording intention of microphones and cameras installed in various places in a concert hall.
3. **Program** subclass:
   This subclass represents the title of a song in a concert or the program of an event. These are the main attributes of the recording target.
4. **Content** subclass:
   This subclass describes the editorial intent and revision description of spatio-temporal media. Just as the **SDMEvent** subclass represents the act of recording, the **Content** subclass represents the act of editing. Typically, this subclass manages the editing history and completed content.

Semantic descriptions that consist of descriptions other than the above definitions are not specified in the SDM Ontology because they are considered to be information of a nature that should be expressed using the vocabulary of an external ontology. Instead, objects of the **Context** class are assumed to be used as relay nodes to link the objects defined in the external ontology. In this sense, a **Context** object is a standard interface object for linking objects represented in the SDM Ontology and objects defined in external ontologies. As a result, it is possible to describe the semantic information of the primary class objects in a unified manner.

Let us consider the Music Ontology [18,19] as an example of an external ontology that describes semantic information. The Music Ontology can describe various semantic information related to music, such as artist names, song titles, compositions, and performances. On the other hand, the **Context** class of the SDM Ontology only specifies the minimum semantic information necessary to define the SDM Ontology.

However, by describing links from **Context** class objects to the semantic information described in the Music Ontology, it should be possible to assign semantic information that incorporates the results of the Music Ontology. In the same manner, it is possible to link to various external objects described in other ontologies via **Context** class objects. It can hence be said that the SDM Ontology has flexible extensibility in semantic information description. MPEG-7 is a standard that has extensive definitions of semantic information of media data. For example, by describing semantic information using an ontology such as the MOWL [37], which describes the semantic information defined in MPEG-7, and linking it from a **Context** class object, the extensive semantic information description of MPEG-7 can be incorporated into the SDM Ontology.

3.3.2. **Recorder**, **Processor**, **Player** Class

The **Recorder**, **Processor**, and **Player** classes represent the equipment and functions used for recording, processing/editing, and playback. Each class has a **Composite** subclass and subclasses corresponding to the type of digital media handled. Figure 4 shows the structure of each class. The **CompositeRecorder**, **CompositeProcessor**, and **CompositePlayer** subclasses describe the composite objects described in Section 3.1. The **Composite** subclasses are defined as subclasses of each class, and the **hasComponent** and **isCompo-**

nentOf properties express inclusion relationships. As a result, the objects of each class can be grouped freely and described hierarchically. The **CompositeRecorder** class is used as an example. The **CompositeRecorder** class is a subclass of the **Recorder** class that represents a composite of **Recorder** objects as a single **Recorder** object. **CompositeRecorder** represents the **Recorder** objects in the inclusion relationship using the **hasComponent** property. By linking multiple **Recorder** objects with the **hasComponent** property, **Recorder** objects can be grouped and represented as a single **Recorder** object. The **isComponentOf** property expresses an opposite relationship. For example, when one wants to represent four sets of monaural **AudioRecorder** 's together as a four-channel **AudioRecorder**, one can create a **CompositeRecorder** object using the **isComponentOf** property to link the four monaural **AudioRecorder** objects. From the linked monaural **AudioRecorder** object, the **CompositeRecorder** object can be connected with the **isComponentOf** property to describe the object inclusion relationship in both directions. By defining **CompositeRecorder** recursively, it is possible, for example, to represent multiple audio and video recording devices that were used to record a concert as a single recording device. Similar expressions are possible using the **CompositeProcessor** and **CompositePlayer** subclasses. The **CompositeMedia** subclass, a subclass of the **Media** class, can also be used for similar representations but has a more extended structure, which is explained in detail in the Section 3.4 clause.

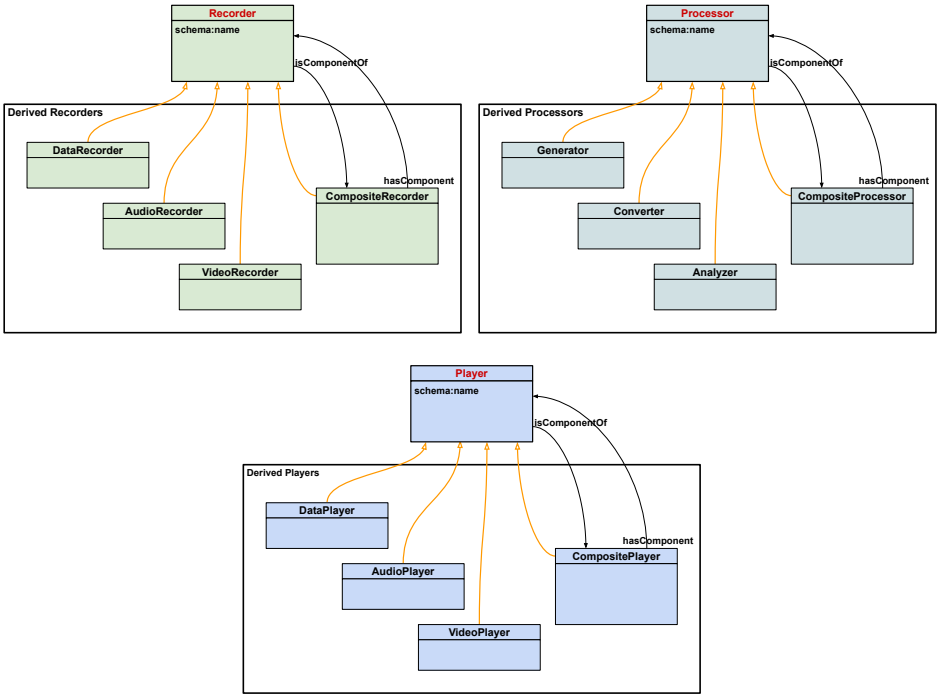

Figure 4. Recorder, **Processor**, **Player** class structure.

3.3.3. **Recorder** Class

The **Recorder** class expresses the devices and functions that perform media recordings. The **Recorder** object performs a recording of the **Target** object referenced by the **records** property and generates a **Media** object as a aresult of the recording. The **Recorder** object links the **Media** object to the **outputTo** property. As mentioned above, a composite of **Recorder** objects can be defined using the **CompositeRecorder** subclass. The **Recorder** class defines three subclasses: **AudioRecorder**, **VideoRecorder**, and **DataRecorder**, depending on the type of media that will be recorded. These subclasses were defined for convenience. There is a degree of freedom in how the **Recorder** object is expressed, and the data creator can decide which subclass to use and whether or not to describe it as a composite device using **CompositeRecorder**. For example, a **VideoRecorder** generally combines stereo audio recording and video recording functions. In the SDM Ontology, this

can be represented as a single **VideoRecorder** or as a **CompositeRecorder** that combines a stereo audio **AudioRecorder** and a video-only **VideoRecorder**. Furthermore, a stereo **AudioRecorder** can represent two monaural **AudioRecorder**. It is up to the data creator to decide which form of representation should be adopted.

3.3.4. **Processor** Class

The **Processor** class represents a device or system that analyzes, processes, edits, and generates spatio-temporal media objects. The **Analyzer** subclass is a **Processor** subclass with only **Media** object input. The **Converter** subclass has both **Media** object input and output, and the **Generator** subclass has only **Media** object output. The **inputFrom** property represents an input **Media** object. The **outputTo** property represents an output **Media** object. Similar to the **Recorder** class, the **Processor** class also has the **CompositeProcessor** subclass, which can recursively describe a composite of **Processor** objects.

3.3.5. **Player** Class

The **Player** class represents a device or system that plays **Media** objects. The **inputFrom** property expresses the **Media** object to playback, and the **plays** property specifies the **Target** object to be used to represent information about the target of the **Media** recording. Similar to the **Recorder** and **Processor** classes, the **Player** class has the **CompositePlayer** class as a subclass, and a composite of **Player** objects can be described recursively.

*3.4. **Media** Class*

The **Media** class represents spatio-temporal media data. One can trace all the objects specified in SDM Ontology, such as media data, related devices and functions, and semantic information that constitutes the spatio-temporal media, starting from the **Media** object. Consequently, we believe that media objects represent digital copies of recorded objects and have the potential to be used as comprehensive descriptions of the history and semantic information of digital media. Figure 5 shows the structure of the **Media** class. The inclusion of the **MediaEvent** class in the definition of the **Media** class differs from the definitions of the **Recorder**, **Processor**, and **Player** classes. The **Media** object can represent any playback section in a media file by specifying the media file and the start and end times in the media file. The default origin of the time axis is the beginning of the media file; however, the time origin can be changed using the **mediaStartAt** property. The start time of the playback section is represented by the **startAt** property, and the end time by the **endAt** property and expresses an offset value from the time origin. A media file is a file that records video, audio, and various observation data along with time information. Audio media files are time-series recordings of audio data and video media files are time-series recordings of image data. A data media file is a time-series record of various types of observational data. Typical media files are mp3, AAC, and WAV format files for audio, mp4, H.264/265, and other compressed video formats. JSON files that record sensor data with timestamps are also media files. Data that does not have a time representation, such as image data, can also be treated as a media file by interpreting it as having a playback time of 0.

The point cloud data acquired by LiDAR is a spatio-temporal media that is a time-series aggregate of many ranging results and can be expressed as **DataMedia**. By processing this point cloud data together with the movement trajectory of LiDAR device, which is expressed as the location information of **Recorder**, it can be converted into a 3D object [50], and the resulting media file (e.g., mesh data) can also be expressed as **DataMedia**. In addition, technologies have been developed to synthesize realistic 3D images by combining point clouds with video images or photographs [51]. In the near future, it is expected that CG images will increasingly be used instead of video images, and the media files used for this purpose can be described as **DataMedia**.

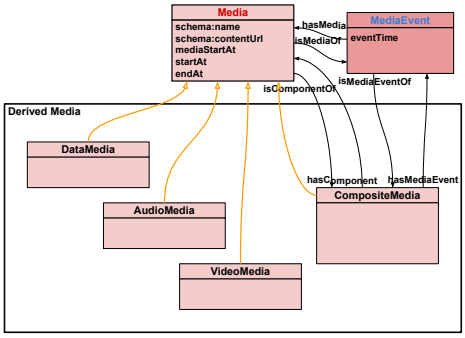

Figure 5. Media class structure.

In the SDM Ontology, there is no definitional distinction between the **AudioMedia** subclass, **VideoMedia** subclass, and **DataMedia** subclass, and the definition of each subclass is for convenience. It is up to the data creator to decide which subclass to use or to describe the object simply as a **Media** class. An SMF (Standard MIDI File) describes MIDI protocol data with time information (https://www.midi.org/specifications-old/item/standard-midi-files-smf (accessed on 1 June 2023)); for example, if the SMF file is a file of performance data that assumes the playback of musical sounds by a MIDI tone generator, it is appropriate to treat it as **AudioMedia**. However, if the SMF file describes a control sequence of stage equipment using the MIDI Show Control protocol (http://www.richmondsounddesign.com/docs/midi-show-control-specification.pdf (accessed on 1 June 2023)), it may be more appropriate to treat it as a **DataMedia** object or simply a Media object.

The **CompositeMedia** subclass represents a composite object of the **Media** class. However, this definition is extended because it is sometimes necessary to specify the start time of the Media object to be included in the description of a composite object. In addition, a **CompositeMedia** object may be created by simply grouping multiple **Media** objects or by grouping the encompassing **Media** objects with their playback start times specified as a pair. An example of the former is a case in which video and audio media are recorded simultaneously to represent a single medium. In this usage, the **hasComponent** property specifies the **Media** object that the **CompositeMedia** object contains. The **hasComponent** property indicates that the linked **Media** object is a component of its own **CompositeMedia** object. The time axes represented by the **Media** objects linked by the **hasComponent** property are all parallel, and each is a **Media** object with the same start time. The **isComponentOf** property describes the inverse relationship to the **hasComponent** property. The latter method, which represents a **CompositeMedia** object by specifying the start time of playback of the encompassing **Media** objects, allows **Media** objects with more complex structures to be expressed. For example, consider multiple audio and video **Media** objects used in producing CDs, DVDs, and Blu-ray discs. The **Media** objects are divided into multiple tracks or chapters and arranged along the time axis. In a typical production flow, media files are created as units of tracks and chapters. These can be represented by **AudioMedia** and **VideoMedia** objects. By arranging each Media object on the time axis of the **CompositeMedia** object, it is possible to describe the **CompositeMedia** object for an entire CD, DVD, or Blu-ray disc. The **MediaEvent** class represents a pair of individual Media objects in a **CompositeMedia** object and its start time. The **CompositeMedia** object is linked to the **MediaEvent** object using the **hasMediaEvent** property. The **isMediaEventOf** property is a property that expresses the reverse relationship. The **eventTime** property of the **MediaEvent** class specifies the start time of the playback section (from **startAt** to **endAt**) of the **Media** object linked by the **MediaEvent** object. The time specified by **eventTime** is expressed in terms of the time axis specified by the **measuredBy** property of the **CompositeMedia** object. If the value of the **eventTime** property of the **MediaEvent** object is zero, it is equivalent to specifying a **Media** object with **hasComponent** as described above. By

combining **CompositeMedia** objects hierarchically, **Media** objects can be represented with arbitrary complexity.

*3.5. **Geometry** and **CoordinateSystem** Classes*

Since spatio-temporal coordinate information is essential among the metadata handled by the SDM Ontology, we defined the **Geometry** class and the **CoordinateSystem** class separately from the **Context** class, as shown in Figure 2. Primary class objects in the SDM Ontology can express position and time information by linking **Geometry** object through the **geometryAt** property. The **geometryOf** property indicates a link from the **Geometry** object to the referenced object. The **CoordinateSystem** class defines the coordinate axes of the position and time information described in the **Geometry** class. Based on this definition, users can interpret the meaning of values expressed in **Geometry** objects. The **measures** property expresses the link in the reverse direction. A derived coordinate system can be defined by linking **CoordinateSystem** objects with the **measuredBy** property.

When a **CoordinateSystem** object is linked to a **Recorder** object with the **measuredBy** property, the linked Recorder object defines the coordinate system of the **CoordinateSystem** object. For example, consider the case in which the internal clock of the video camera used for recording is not correctly set. As a result, the timestamps of the media files recorded by the video camera will be recorded incorrectly. If this media file is edited together with the media files recorded by other equipment, it will not be correctly processed. It is necessary to correct the time information before editing. The design policy of the SDM Ontology is to describe events in the real space as they are in the real world as much as possible. If the recorder's clock was wrong, we believe that we should describe the state of the recorder as it is to demonstrate that the internal clock of the **Recorder** object with the wrong clock defines the time axis of the **CoordinateSystem** object and links the **Recorder** object from the **CoordinateSystem** object with the **measuredBy** property. It is also possible to link a **CoordinateSystem** object to a **Media** object using the **measuredBy** property. This case also indicates that the time information in the **Media** object is defined in the linked **CoordinateSystem** object. Suppose that the linked **CoordinateSystem** object is defined by the internal clock of the **Recorder** object with the wrong clock, as described above. In this case, the time information in the **Media** object is also recorded using the wrong clock.

It is necessary to measure the error of the recorder's clock to correct the timestamps in the subsequent process. If the clock's error is known, we can define a **CoordinateSystem** object with the correct clock. Then, the time information of **Geometry** objects, **Media** objects, and other objects can be linked by the **measures** property from the **CoordinateSystem** object defined by the wrong clock for correction. Finally, the time information correction process is completed by replacing the **measuredBy** property of these objects with the **CoordinateSystem** defined by the correct clock. The time correction process can also be described by writing the time correction program in a **Processor** object and linking the objects before and after correction with the **inputFrom** and **outputTo** properties.

4. Dataset

We created a dataset based on video and audio media data as well as related metadata for the following two music events recorded in the past by the SDM Ontology Version 2.

4.1. Music Events

See [17,52] for a more detailed description.

1. Keio University Collegium Musicum Academy Concert Recording. This is a recording of the Keio University Collegium Musicum Academy concert held at Fujiwara Memorial Hall, Keio University Hiyoshi Campus, on 10 January 2016. The concert was performed by 24 musicians, all of them acoustic. The microphones located near the instruments recorded the performance sound of each instrument part, while the microphones located on the audience side recorded the combined sound of each instrument [52]. Figure 6 shows the arrangement of the cameras and microphones.

.

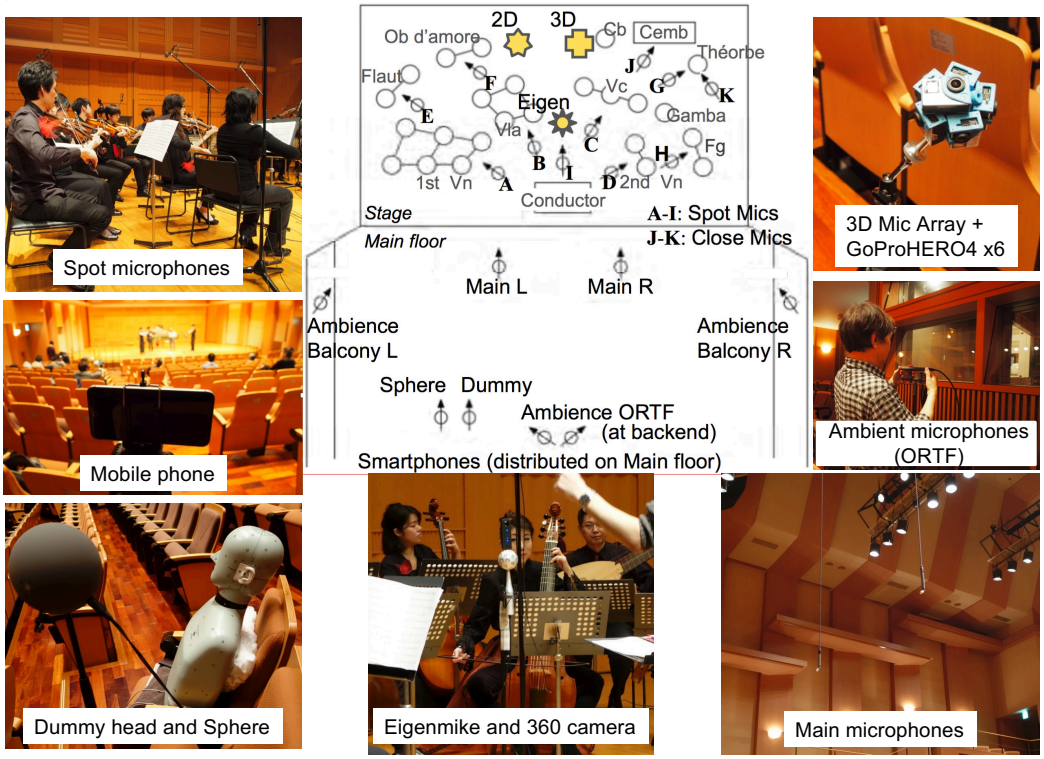

Figure 6. Keio University Fujiwara Memorial Hall recording.

2. Recording at Billboard Live Tokyo On 26 January 2017, a jazz band concert was recorded at Billboard Live Tokyo, a 300-seat venue in Roppongi Midtown, Tokyo. The band consists of three parts: drums, electric bass, and keyboard. Figure 7 shows the band composition, the location of the 360-degree camera, and the location of the microphones.

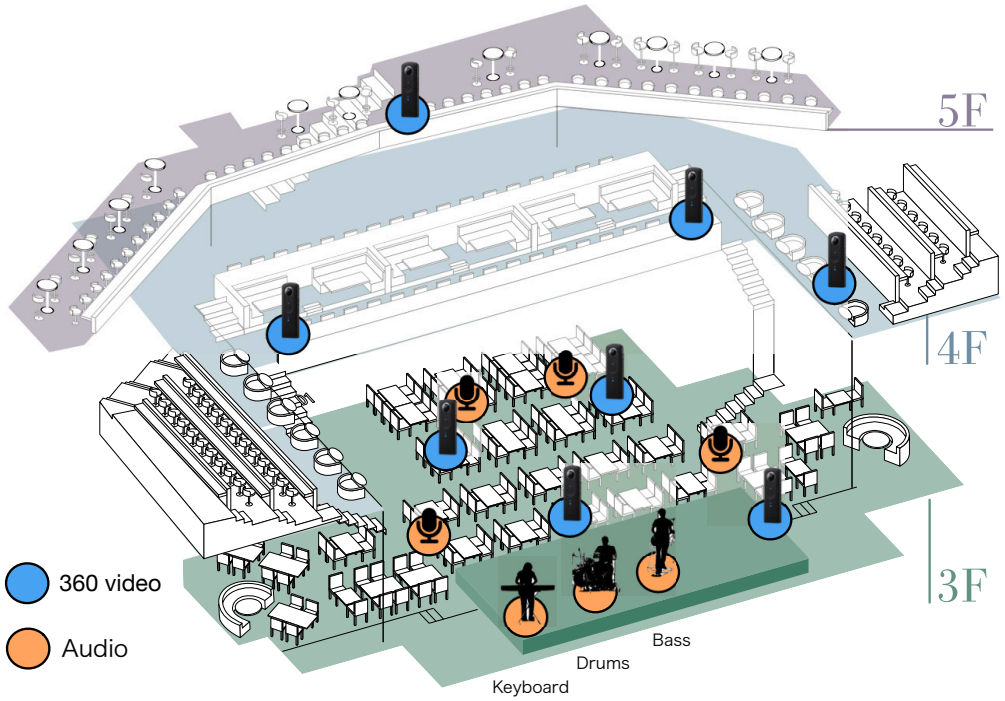

Figure 7. Billboard Live Tokyo recording.

The recording was made with four audience microphones in addition to the three individual instruments. The video was shot by eight cameras.

Information related to the recording, such as the microphone and camera equipment information, installation locations, and recording targets was manually tabulated and compiled into a list. In addition, the time synchronization of the recorded video and audio media data was performed manually.

4.2. Creating Dataset

The above-recorded data and related information can be reorganized to be consistent with the SDM Ontology data structure and described as a group of object data that conform to each class definition and are linked to each other to create an SDM Ontology-compliant dataset. For example, as shown in Figure 8, object data are defined after organizing and classifying various types of information, such as the recording target, recording equipment, generated media files, and their editing/processing processes, to correspond to the various classes of the SDM Ontology. These object data are in the form of RDF datasets described in turtle notation, as shown in Figure 9. By storing the RDF dataset created in this manner in GraphDB (http://graphdb.ontotext.com/ (accessed on 1 June 2023)), an endpoint that accepts SPARQL queries can be implemented and used as LOD (http://sdm.hongo.wide.ad.jp:7200/ (accessed on 1 June 2023)).

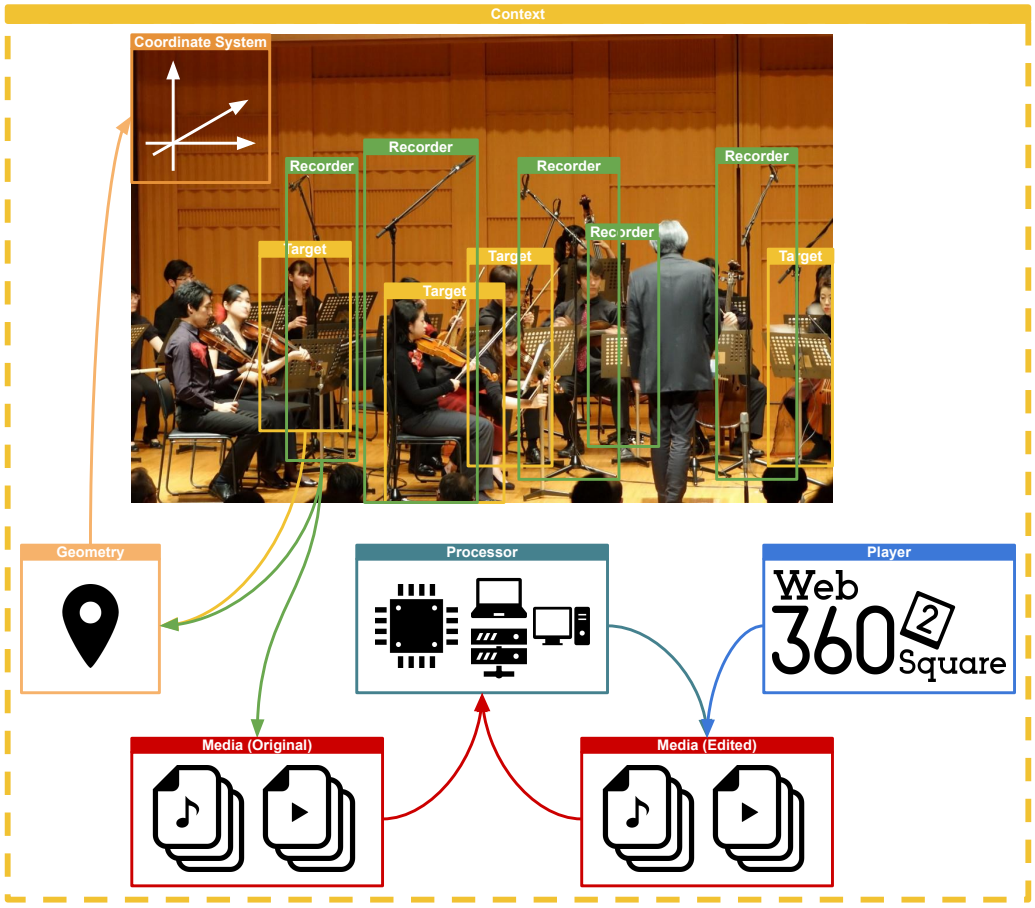

Figure 8. Schematic diagram of object definition.

```
# 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 .

# ...
```

Figure 9. RDF data (excerpts).

5. Web360^2—Interactive Application Applying SDM Ontology

In this study, we developed an application called Web360^2 [53], which operates based on an SDM Ontology-compliant dataset and was evaluated through experiments. This development is intended to verify the structure of the ontology from the viewpoint of the content creator and to help the adoption of the SDM Ontology.

Web360^2 is a web application that allows users to experience 360° videos and spatial audio. Free viewing using pre-recorded 360° videos on a web browser makes it possible to control the playback of audio sources based on the viewer's interactive operations. Moreover, Web360^2 can be run without installation by simply accessing this URL (https://sdm-wg.github.io/web360square-vue/#/ (accessed on 1 June 2023)) from a PC, smartphone, or tablet browser. The source code of Web360^2 is available on GitHub pages (https://github.com/sdm-wg/web360square-vue (accessed on 1 June 2023)).

5.1. Design Policy

We established the following design principles for the Web360^2 development.

1. Data-driven application
   The application retrieves media files and the metadata it uses from the RDF store and operates on them.
2. Web application
   We aim for ease of use directly from a web browser without the need to install a dedicated application.
3. Video and audio playback from arbitrary viewpoints
   By automatically rendering and presenting appropriate video and audio based on the 3D positional information of the viewing object, the viewer can experience video and audio playback from any viewing position or angle.
4. Interactive audio object manipulation
   Viewer operations allow the individual access to each audio object and switch it on or off, enabling a highly flexible viewing experience based on the viewer's interests.
5. Streaming distribution
   Stream 360° video via HTTP Live Streaming (HLS). By supporting streaming distribution, it is possible to shorten the waiting time before playback starts and eliminate unnecessary communication. Although MPEG DASH and MediaSouce APIs have the potential to achieve higher streaming performance than HLS, this study will use HLS in favor of its native support in Android and iOS.

5.2. System Design

Web360^2 retrieves the list of available content by querying the RDF store using a SPARQL query(Figure 10). When the user selects content to watch, Web360^2 again performs a SPARQL query on the RDF store to retrieve the video and audio media files and metadata contained in the selected content to start playing the video and audio. In addition, Web360^2 accepts viewer and touch operations on audio objects as input. The system renders and outputs video and audio in real time in response to the user operations, making the process intuitive and interactive. The video and audio media files are independent of each other and playback in synchronization within the system. Furthermore, 360° videos are streamed in HLS and projected onto a virtual sphere centered on the viewer. Based on the 3D positional information of the audio objects, audio visualization objects are drawn at each point inside the virtual sphere where the video is projected. At the same time, spatial acoustic playback is performed based on the positional information.

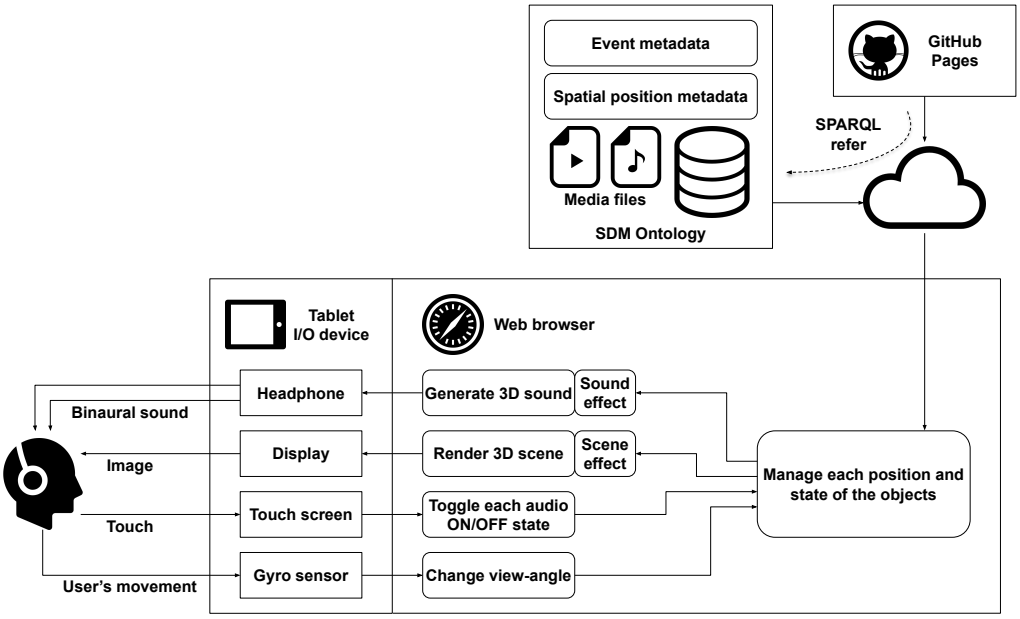

Figure 10. Web360^2 system design.

5.3. Implementation

Web360^2 was implemented using Vue.js, which is a JavaScript framework. The framework for the web-based VR was implemented using A-Frame (https://aframe.io/ (accessed on 1 June 2023)) and the Web Audio API (https://www.w3.org/TR/webaudio/ (accessed on 1 June 2023)), which enables advanced audio processing on the Web. Figure 11 presents an overview of the implementation of the content viewing part. Figure 12 shows screenshots of the viewing experience. The application's operating environment is a browser application on PCs and smart devices (Android and iOS) and VR headsets are not included.

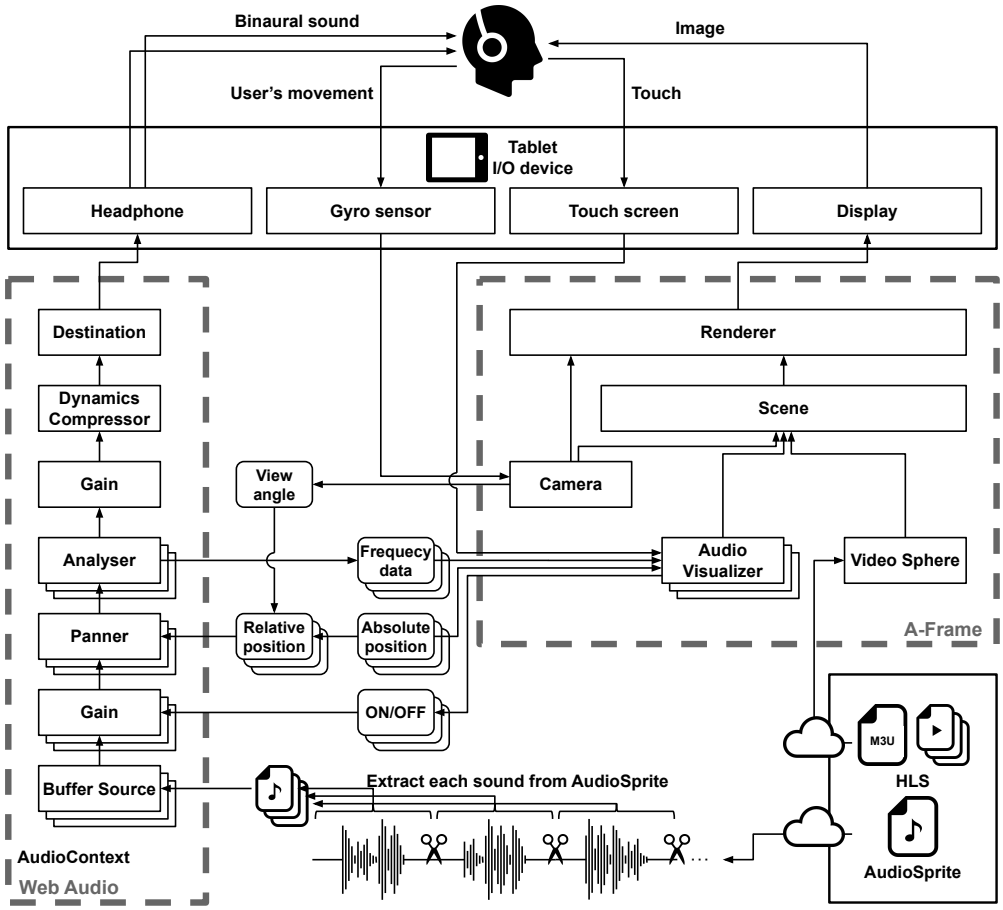

Figure 11. Web360^2 implementation overview. The left dashed box indicates the processing using Web Audio API, and the right dashed box indicates the processing using A-Frame.

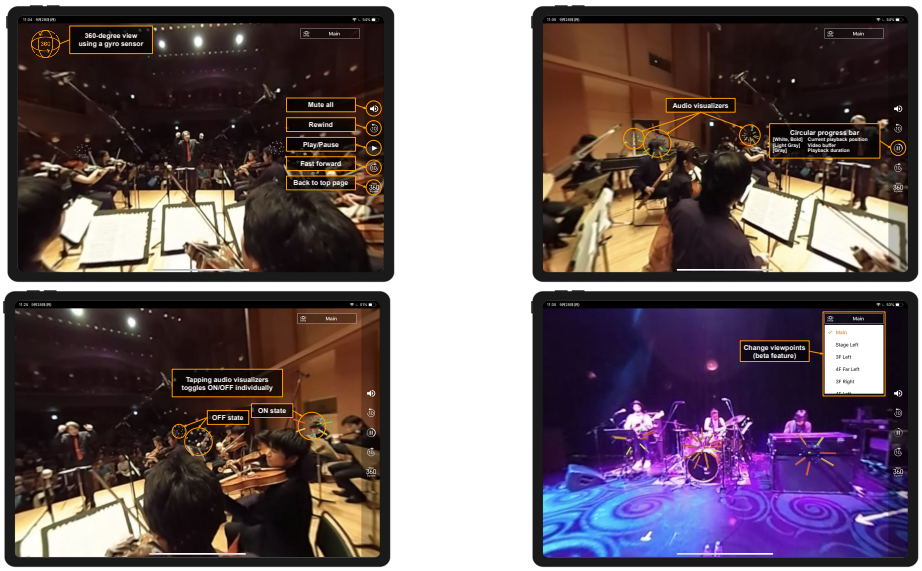

Figure 12. Screenshots of Web360^2 during the viewing experience. The control panel on the right side of the screen allows basic operations such as playback, pause, fast forward, and rewind. On devices equipped with gyro sensors, the viewing angle fluctuates by movement. Audio is managed individually and can be turned on and off independently by tapping (clicking).

5.3.1. Dynamic Data Acquisition by SPARQL Query

As mentioned earlier, Web360^2 retrieves the data necessary for the operation of an application by issuing SPARQL queries. The following two types of queries were issued.

1. Event list
   Web360^2 issues a SPARQL query of Figure 13 to the SPARQL endpoint and displays the event selection screen based on the information obtained through the search, as shown in Figure 14. Figure 15 shows the SPARQL search path.

```
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 .
}
```

Figure 13. SPARQL query for obtaining event list.

Figure 14. Screenshot while selecting 'Keio University Collegium Musicum Academy Concert Recording'.

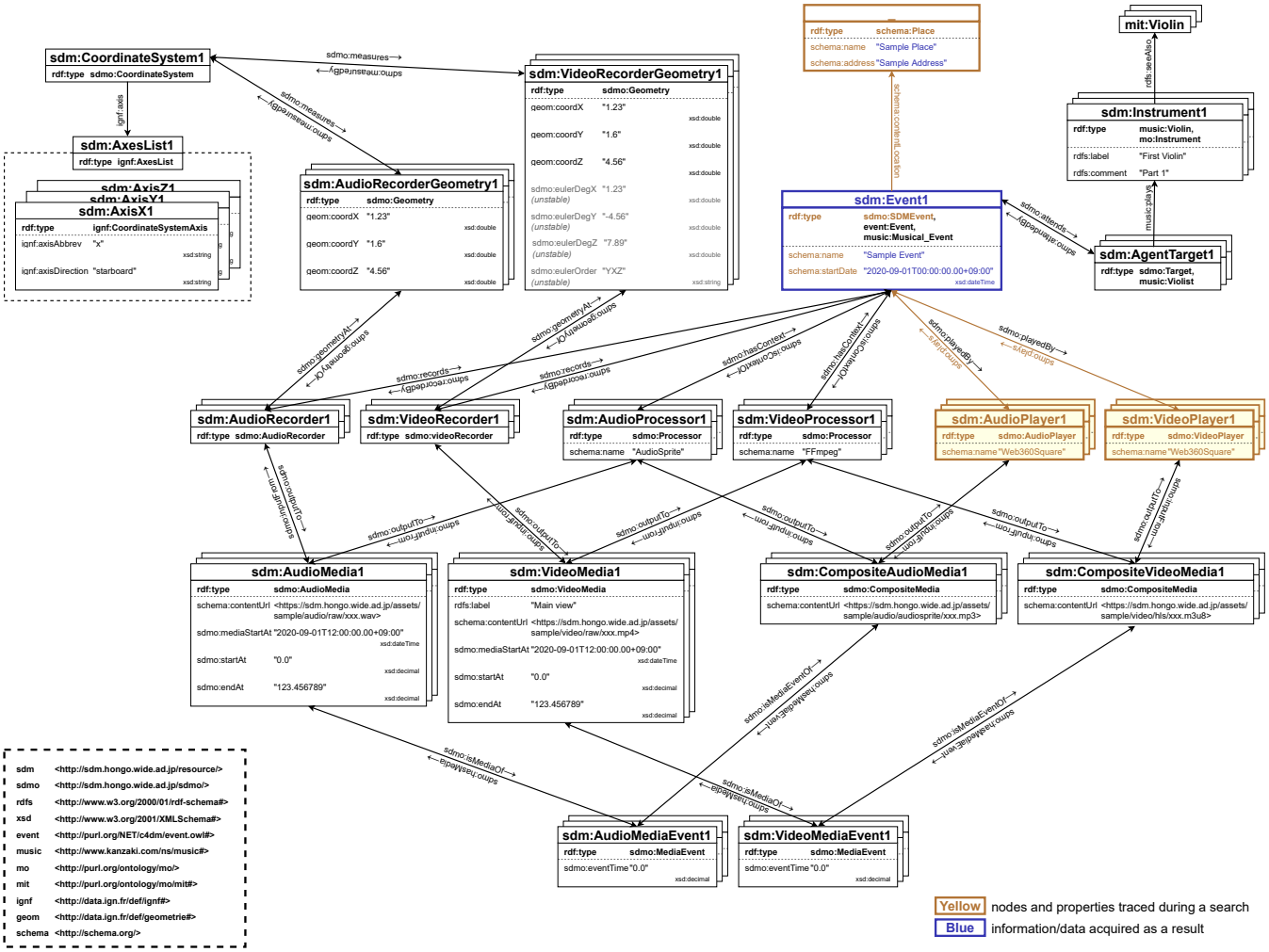

Figure 15. Overview of obtaining event resources. When the query shown in Figure 16 is issued, a search is performed based on VideoPlayer1 and AudioPlayer1, and the sdmEvent (blue letters) that matches the search condition is returned.

2. Event resources

Web360[2] issues the SPARQL query in Figure 16 to obtain viewing information when an event is selected from the event selection screen. Figure 17 shows the SPARQL search path. Then it transitions to the viewing screen in Figure 12. The 360° video and audio media files and their positional information are needed because each audio visualization object must be rendered in the correct alignment concerning the viewpoint on the viewing screen. In addition, since video and audio are independent media files, synchronization of video and audio is required. Details of the synchronization processes are described in Section 5.3.5.

```
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 .
  }
}
```

Figure 16. SPARQL query for obtaining viewing information.

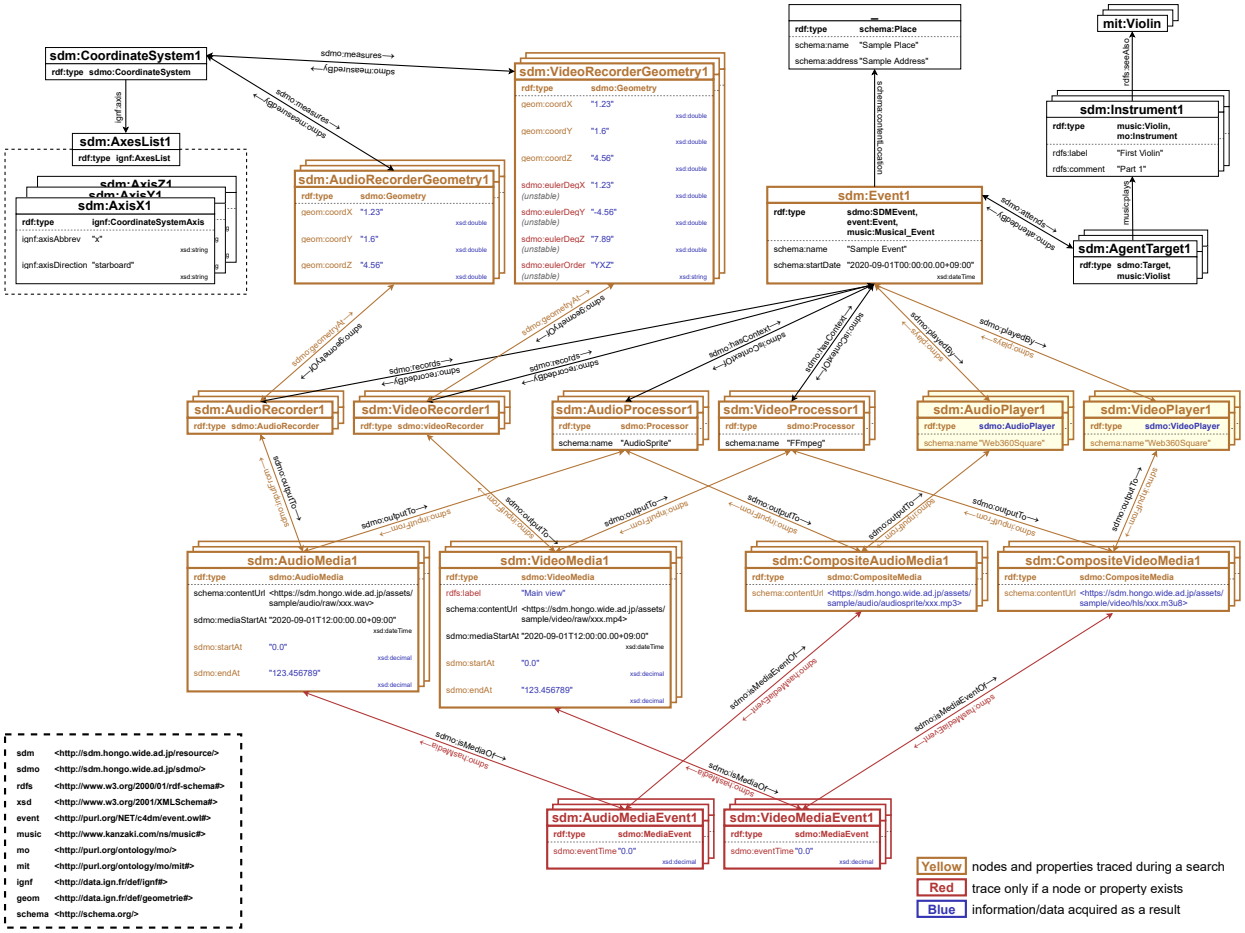

Figure 17. Overview of obtaining viewing information. When the query shown in Figure 16 is issued, a search is performed based on VideoPlayer1, AudioPlayer1 and the properties (blue text) that match the search criteria are returned.

5.3.2. Video Processing Using A-Frame

Web360^2 uses A-Frame to project 360° videos and draw audio visualization objects in the procedure shown in the right dashed line box of Figure 11. The 360° videos are streamed in HLS and projected inside a sphere centered at the camera coordinates using an A-Frame `a-videosphere` object. HLS is natively supported by Safari, Edge, and some mobile device browsers; however, many browsers such as Chrome, Firefox, and IE do not support HLS. Web360^2 supports HLS using hls.js. Audio visualization is represented using the frequency domain data of each voice provided by the Analyzer node of Web Audio. The viewer's interactive operations are input mainly through the processing handled by A-Frame. The viewer's head movement is detected by the gyro sensor mounted on the terminal, and the viewing angle is changed by inputting the detection results in the camera object in A-Frame. The touch operation is recognized as input by touching an object on the screen rendered by A-Frame, and the ON/OFF state of each audio can be changed.

5.3.3. Audio Processing Using Web Audio API

Web360^2 performs audio processing, as shown in the left dashed box of Figure 11 and renders the audio from each audio object in the ON state in real time according to the viewing position and angle. In addition, Web360^2 extracts each audio source data from the AudioSprite (https://github.com/tonistiigi/audiosprite (accessed on 1 June 2023)) file. It then inputs it into the Buffer Source node of the Web Audio API. The next Gain node expresses the ON/OFF state by setting the audio gain in the OFF state to 0. A-Frame can automatically render appropriate images by specifying each display object's viewing

position, viewing angle, and position. However, the position of the audio object played by the Web Audio API does not automatically follow the viewing position and angle changes. Web360^2 computes the relative coordinates of audio objects sequentially based on the viewing position and viewing angle and passes them to the Panner node to track the position of the audio object. The frequency spectrum of each audio source data is calculated using the Analyzer node and used for the A-Frame audio visualization process. Web360^2 mixes these audio data via Gain and Dynamics Compressor nodes, which serve as master gains, and outputs them as binaural audio. The user can experience spatial acoustic reproduction synchronized with the application video by listening to this binaural audio through headphones or earphones.

5.3.4. Visualization of Audio objects

Web360^2 displays audio objects as audio visualization objects. An audio visualization object is a display object consisting of spine-shaped objects on a spherical object, each of which changes in length and color according to the frequency spectrum of the corresponding audio. The number of spine-shaped objects in the audio visualization object is 32, and the frequency domain data of the audio is allocated to 32 objects as shown in Figure 18.

The audio spectrum is obtained approximately 20 times per second by using the Web Audio API method (AnalyserNode.getByteFrequencyData()). Because the audio spectrum is biased and the absolute value differs significantly from one audio data point to another, the appearance of the audio visualization object will be unbalanced if the weights are used as they are. To balance the appearance of the audio visualization object, the upper and lower limits of the range in which the frequency spectrum exceeds certain levels are detected and then divided into 32 groups. The values of each group are normalized, and the order of the groups is randomized and then converted to the length and color of the spine-like object for drawing. Only the visible spine-like objects are drawn, and the drawing load is reduced by omitting the drawing of objects in the blind spots. The color of the spines of the audio visualization object in the OFF state is gray, and in the ON state, the color changes from blue to light blue, green, yellow, orange, and red as the average value increases.

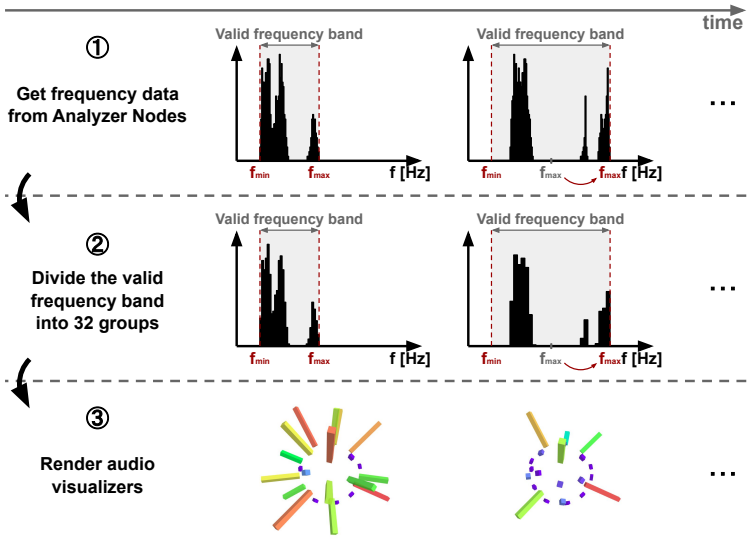

Figure 18. Visualization procedure of audio objects.

5.3.5. Video and Audio Synchronization

Video and audio media files are independent and must be played back during synchronization. The playback position of the video can be obtained and set using the JavaScript's `HTMLMediaElement.currentTime` property. Meanwhile, the Web Audio API does not provide a way to directly obtain the playback position of audio. Web360^2 indirectly detects the

playback position of audio by recording the value of the `AudioContext.currentTime` property when the `AudioBufferSourceNode.start()` and `AudioBufferSourceNode.stop()` methods are used. At the start of playback and when the playback position returns to the beginning owing to loop playback, the playback position of the video is forcibly synchronized based on the calculated playback position of the audio. During playback, the difference between the playback position of the audio and video is detected and synchronized in the A-Frame camera object's `tick` handler, which is called 60-120 times per second. Instantaneous changes in the playback position of the video or audio degrade the playback quality, which is undesirable. If the playback speed of at least one of the video and audio is changed to match the playback position of both, the playback position will synchronize without degrading the playback quality. Changing the playback speed of the video can be easily set using the JavaScript property (HTMLMediaElement.playbackRate). However, changing the playback speed of the audio requires complex processing owing to the pitch conversion required. Web360^2 synchronizes video and audio by doubling or halving the playback speed of the video when the time difference between the audio and video playback positions exceeds a threshold value. The threshold value for this time difference is 0.1 s. However, if the time difference between the video and audio is more than 1 s, the playback position of the video is forcibly synchronized with that of the audio.

5.3.6. Moving the Viewing Position

The function of moving the viewing position is in the experimental stage and is a tentative implementation. Because Web360^2 plays back 360° videos, the viewing position is limited to the position of the video camera that captured every 360° video. This limitation was caused by the fact that the filmed objects were not objectified. It is not easy to efficiently convert a moving object into a 3D object and play it back at a quality suitable for viewing. Because audio is converted into object data using location information, it is possible to playback audio from arbitrary locations. As the video data volume is larger than a realistically available communication bandwidth, there are restrictions on switching the viewing position. To quickly change the viewing position, loading at least the first part of all video data in advance is necessary. However, this will increase the communication load before playback starts, and the user will have to wait until playback begins. With the current Web360^2 implementation, the application loads only the video data of the initial viewing position, and then playback starts; therefore, it takes time to change the viewing position. There is a trade-off between the amount of communication before playback starts and the smoothness of changing the viewing position.

6. Evaluation of SDM Ontology

In the following sections, we summarize the results of our evaluation of the SDM Ontology described in Section 3 using existing index values and online services.

6.1. Evaluation by Ontology Evaluation Index

Fernández et al. [54] defined 12 indicators for evaluating and measuring ontologies. In their Internet of Musical Things Ontology [33], Turchet et al. used the following indicators, which were modified from some of the above indicators, for ontology evaluation.

(a) Indicators of knowledge coverage and popularity

- Number of classes
  The number of classes in the target ontology.
- Number of Properties
  The number of properties in the target ontology.
- Number of individuals
  The number of individuals in the target ontology.
- Number used in the external ontology
  The number of times the target ontology is used by external ontologies. In general, the more mature the ontology, the larger the value of this indicator.

- Number of external ontologies used
  The number of times the target ontology uses external ontologies.

(b) Indicators for data structure

- Minimum number of triples
  The minimum number of triples required to describe the data for which the target ontology exists.
- Maximum number of triples
  The maximum number of triples that can be used to describe the data for which the ontology of interest exists.

Table 1 presents the performance of the SDM Ontology based on the indicators of knowledge coverage and popularity among these indicators. Because the SDM Ontology is a new ontology, it has not yet been used by external ontologies. External ontologies referenced by the SDM Ontology are ignf, Dublin Core, and schema-org. Only a basic vocabulary has been defined for using an external ontology. We expect this number to increase as more detailed definitions are added in the future.

Table 1. Indicators on knowledge coverage and popularity.

Index	Value
Number of classes	30
Number of properties	31
- Object properties	23
- Data properties	8
Number of individuals	0
Number of external ontologies using it	0
Number of external ontologies used	3
- Percentage of external classes	10.0%
- Percentage of external properties	16.1%

6.2. Evaluation by OOPS!

The SDM Ontology was also evaluated using OntOlogy Pitfall Scanner! (OOPS!) [55]. OOPS! is a RESTful web service that detects common hazards that can lead to ontology flaws according to the existing literature. The SDM Ontology detected several important or minor hazards but no critical hazards. The following are the important hazards detected.

Important pitfalls

1. The prime relationships among elements are not defined
   The SDM Ontology contains elemental relationships that are prime. However, the definition does not reflect them because they are not discussed in detail.
2. The subject or object of the property is not defined
   These warnings occur for properties referenced using external ontologies and do not consist of a nature that should be defined within the SDM Ontology. We believe that the handling of subjects and objects that are not included in the definition of the SDM Ontology should be handled by the operational rules at the time of data production.
3. Use of recursive definitions
   Recursiveness is an intentionally designed structure. We believe that operational rules should handle problems such as the circular references that may occur with recursive descriptions during data production.

Minor pitfalls

1. Existence of independent ontology elements
   This warning occurs for class definitions with an external ontology as a superclass. This structure was intentionally designed.

2. No inverse relationship defined
   This warning also occurs for properties referenced from external ontologies and is an intentionally designed structure. The properties inside the SDM Ontology have inverse relationships defined.
3. No label/annotation present
   This warning also occurs for properties referenced from external ontologies and is an intentionally designed structure.
4. Misuse of labels and annotations
   This warning occurs when there is a misuse of a label/annotation property for human reading. We believe that a warning is being generated for a label definition defined for symmetry with the definitions of other elements. However, it has not yet been determined whether it should be corrected.

As described in Section 3.3.1, the **Context** class is used as a standard interface for semantic information described in external ontologies. It is up to the user to determine what external ontology to use, and the SDM Ontology definition does not include them. Moreover, recursion is noted as an important pitfall, but this is an intentionally designed structure. Both cases suggest limitations in evaluation methods that refer only to the ontology definition. In order to use the SDM Ontology to create a valid dataset according to the design intent, it is necessary to embed a judgment function in the application that creates the RDF data.

6.3. SDM Ontology Promotion Activities

The purpose of this study is to develop a new way of representing digital media by managing digital media in object units and controlling them with software, and at the same time, to realize flexible and highly applicable systems and services. The SDM Ontology is an ontology designed to serve as a data representation infrastructure for this purpose, and Web360^2 is an application prototype that demonstrates the effectiveness of digital media representation utilizing the SDM Ontology. We believe that the existence of ontology-related content and applications will effectively promote the validity of ontologies to the public and increase the interest in ontologies. The SDM Ontology vocabulary system is available as a web document (https://sdm.wide.ad.jp/sdmo/ (accessed on 1 June 2023)) using WIzard for DOCumenting Ontologies (WIDOCO) [56], which was developed by Garijo et al. In addition, we submitted a Web360^2 application and an SDM Ontology-compliant dataset to the LOD Challenge 2020 (https://2020.lodc.jp/ (accessed on 1 June 2023)). We received the Grand Prize and Sponsor's award, which provides us with an excellent opportunity to showcase the effectiveness of our research.

7. Evaluation of Web360^2

We conducted a subjective evaluation of Web360^2 using an online questionnaire. The results of the assessment are presented in the next subsection. Next, based on the knowledge obtained through the implementation of the application, the performance requirements for the network performance and rendering performance necessary for the application to provide a good viewing experience are described.

7.1. Online Survey

The survey was publicized through online meetings of research organizations, research presentations, and social networking services from mid-December 2020 to mid-January 2021 and was conducted online using a web-based survey response form. Subjects accessed Web360^2 on their device and in their network environment, following written instructions and notes, and completed a web survey after the viewing experience. Given that the viewing environment differs from subject to subject, participants were asked to enter information such as the device name, OS type, web browser type, and network environment, as well as whether they had a gyro sensor and headphones. In addition, information on the

content viewed was also obtained, considering that the results of the questionnaire may be affected by the content owing to differences in the recording method.

7.1.1. Participants

There were 31 subjects: 28 males and three females. The age composition was as follows:

Age Group	Number of Subjects
Teenagers	1
20s	10
30s	2
40s	8
50s	7
60s or older	3

Since publicity was focused on the study group, the subjects were primarily people familiar with information technology.

7.1.2. Questionnaire

The questionnaire contained six questions, Q1 through Q6, as shown below:

Q1 Did you hear the sound from the right direction?
Q2 Did you hear the sound from the proper depth?
Q3 Do you feel that the sound follows the image?
Q4 Can you interact with the 3D content easily?
Q5 Is it intuitive to turn ON/OFF the audio objects using the audio visualizer?
Q6 Can you hear a specific instrument sound by turning ON/OFF audio object?

The subjects answered the questions using a 7-point Likert scale from 1 to 7, with 1 being the lowest rating, 4 being the middle rating, and 7 being the highest rating. Respondents are asked to answer "7" if they feel the question is completely correct, and "1" if they feel the question is completely incorrect, after explaining in advance that they will be asked to complete the survey. Q3 asks whether the user perceives the 3D effect of the sound synthesized in the system. Q4 and Q5 pertain to the interactive viewing experience through head movements and touch movements while holding the tablet. Q4 asks about the overall operability, and Q5 asks about the acoustic objects' visualization and ON/OFF operation. Q6 asks whether the user can decompose the overall sound and hear the tones of the individual instruments through the ON/OFF operation of each sound.

The following information was also collected to investigate the different viewing environments for each subject.

- Device name
- OS type/version
- Web browser type/version
- Network environment
- Gyro sensor use
- Headphone use
- Content viewed and experienced

This information was used to investigate the influence of differences in the viewing experience environment on the evaluation results.

In addition, a free description column for "impressions, requests, and concerns" and "bug reports" was provided at the end of the questionnaire.

7.1.3. Results

Figure 19 presents the online survey results.

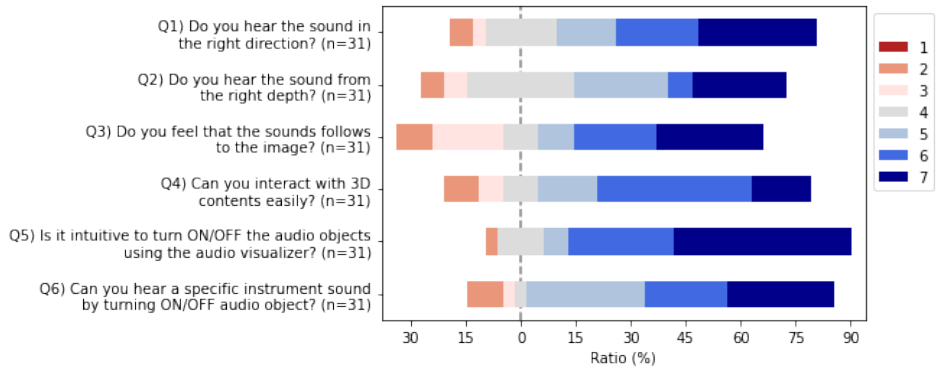

Figure 19. Percentage of survey responses in the online survey.

In this figure, the center score of the 7-point Likert scale (i.e., 4), representing the middle rating, is placed at the origin 0 of the horizontal axis. The response scores 5, 6, and 7, which express high ratings, extend out in the positive direction. The response scores 3, 2, and 1, which represent low ratings, extend out in the negative direction. The results of all questions showed that the number of high evaluation score exceeded those of the low evaluation scores, indicating that the quality of the interactive viewing experience of 3D video and audio provided by the Web360^2 was generally good. Figures 20–23 summarize the evaluations for the different operating environments. Table 2 lists the mean and *p*-values of the evaluation values for each environment, organized by question item. The *p*-values were calculated by performing a t-test on the evaluation results for each environment. The *p*-value is 1.00 when the mean scores of the two environments are identical. Generally, a *p*-value of 0.05 or less is considered a significant difference. In the present study, there were differences in the mean values in some cases, but none were considered significant because all *p*-values were larger than 0.05.

Table 2. List of mean evaluation scores and *p*-values for different operating environments.

	N	Q1	Q2	Q3	Q4	Q5	Q6
PC	17	5.77	5.12	5.53	5.18	6.18	5.65
Smartphone	14	5.00	4.79	4.43	5.29	5.86	5.14
	p-value	0.19	0.57	0.10	0.85	0.49	0.38
iOS	9	5.00	4.56	4.67	5.44	5.67	5.56
Android	5	5.00	5.20	4.00	5.00	6.20	4.40
	p-value	1.00	0.61	0.54	0.69	0.38	0.33
Gyro	16	5.31	5.06	4.63	5.19	6.06	5.19
NoGyro	16	5.53	4.87	5.47	5.27	6.00	5.67
	p-value	0.69	0.72	0.19	0.89	0.90	0.39
Headphone	25	5.40	5.08	5.08	5.28	6.20	5.64
No Headphones	6	5.50	4.50	4.83	5.00	5.33	4.50
	p-value	0.87	0.24	0.70	0.70	0.29	0.16

In Figure 20, we compared the types of devices used by the subjects, roughly classified into PC and tablet/smartphone. The mean scores for the tablet/smartphone group were lower than those for the PC group for all questions. Nevertheless, even the lowest Q3 has a *p*-value of 0.10, which is not a significant difference. Although the results indicate said that the PC group tended to be rated higher overall, the *p*-value was 0.10, even for the lowest Q3, and hence this is not a significant difference. Some open-ended responses from the tablet/smartphone group indicated that the application rendering process was insufficient, suggesting that the performance may have been inadequate compared with PCs. Another

reason for the low evaluation may be the poor operability of the tap operation on the smartphone device with a small screen size compared with the mouse operation on the PC when switching the audio visualization object on and off.

In Figure 21, we compared the differences in the evaluation according to the OS for the tablet/smartphone group and found no apparent difference between iOS and Android. Meanwhile, Figure 22 shows a comparison of the spatio—temporal-related ratings with and without the gyro sensor. We expected that using the gyro sensor would improve the evaluation of interactive operation in question Q4, however, no clear difference was observed, and a trend toward lower evaluation was noticeable. On terminals with gyro sensors, Web360^2 automatically reflects the orientation of the device in the direction of the line of sight. Because most devices equipped with gyro sensors are tablets and smartphones, this may reflect the effect of the difference in evaluation by the type of device shown in Figure 20.

Figure 23 shows a comparison of the evaluation results regarding the use of headphones. Because spatial sound reproduction by the Web Audio API is more effective when headphones are used than loudspeakers, the answers to questions Q1, Q2, Q3, and Q6 should be affected. A comparison of the average of the evaluation values for each question reveals that the subjects using headphones gave higher evaluation scores than those using headphones for all questions except for Q1, which was scored almost the same. The difference in the evaluation score for Q6 was the largest, suggesting that headphones may effectively distinguish individual instrumental sounds. However, the p-value for Q6 was 0.16; hence, the questionnaire results alone do not indicate a significant difference. Some of the responses from headphone users indicated in the free description column that they only listened to one ear. It should be noted that using headphones does not necessarily mean that the user is listening to the sound in stereo.

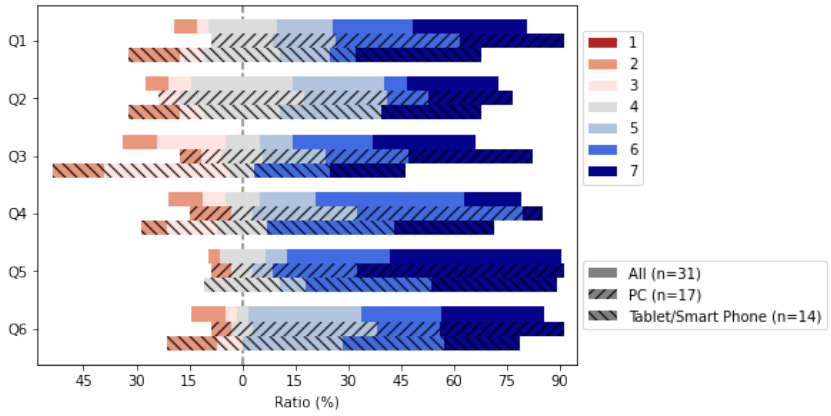

Figure 20. Comparison of ratings by device type.

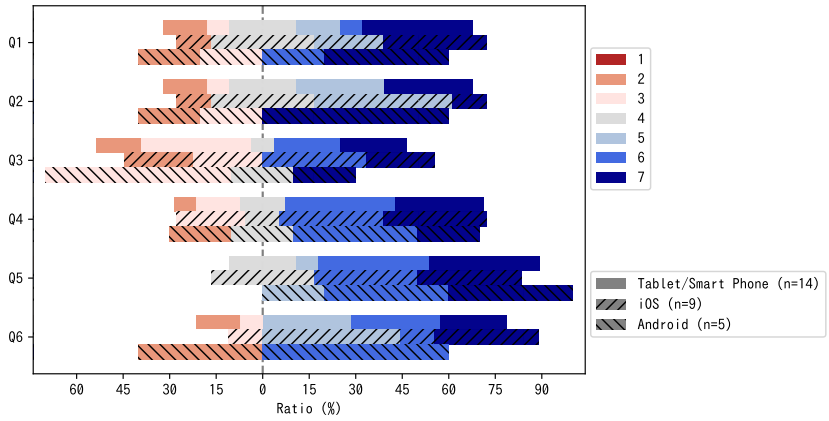

Figure 21. Comparison of ratings by OS.

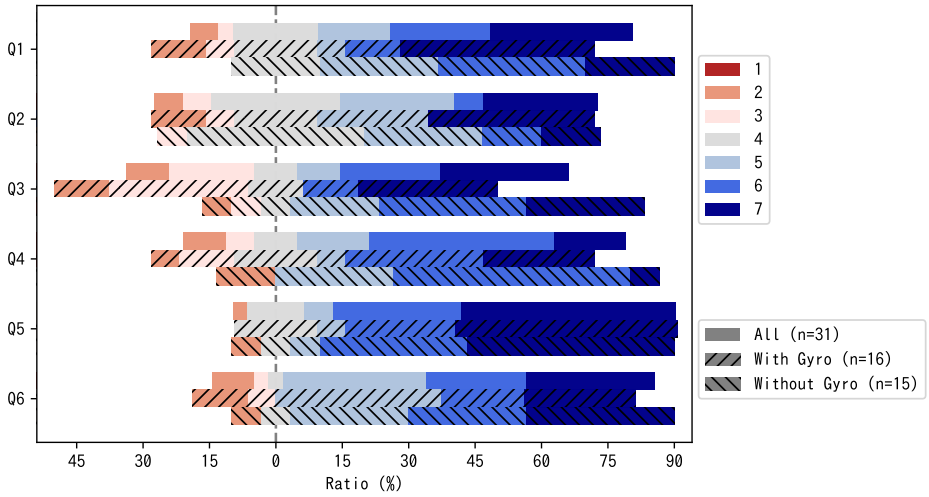

Figure 22. Comparison of evaluation results with respect to gyro sensor use.

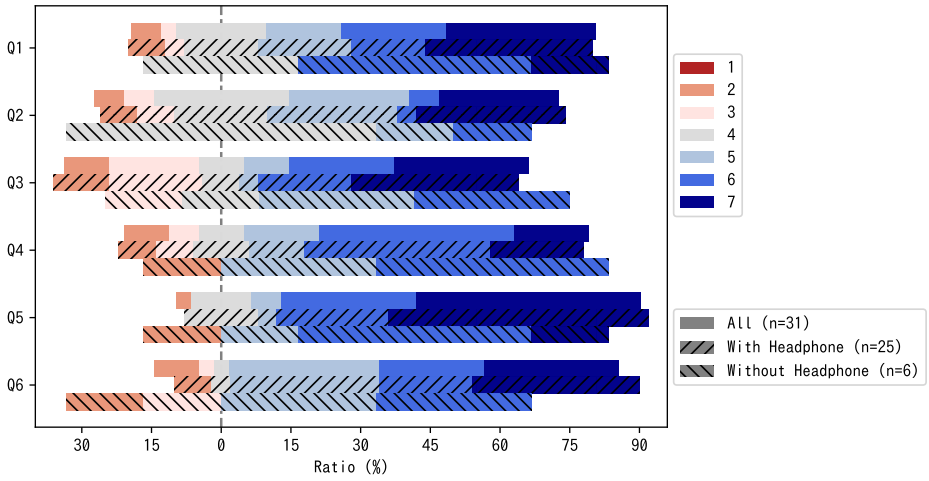

Figure 23. Comparison of evaluation results with respect to headphone use.

7.2. Performance Requirements Assessment

7.2.1. Required Network Performance

Web360^2 implements a specification to load all audio data and up to 40 s of video data to view the content before playback begins. For both the concert and jazz session contents, the audio data size was approximately 10 MB and the 40 s of video data were approximately 30 MB. Hence, approximately 40 MB of media data had to be loaded before playback started. Therefore, if the allowable latency to start playback is less than 10 s, a communication speed of 32 Mbps is required. Because the number of audio objects and playback time were relatively small for the content used on Web360^2, we implemented a specification to load all audio data before playback began. If audio data are streamed in the same manner as video data, the application can reduce the amount of data to be loaded before playback starts to some extent. After playback starts, the application acquires the video data at any time in units of approximately 10 s. Consequently, it is necessary to retrieve 5–9 MB of video data approximately every 10 s, which requires a minimum communication speed of approximately 8Mbps. However, a higher communication speed is required if audio data are streamed and played back. Although there is still room to optimize the buffering size and loading intervals of the audio and video data described above, which is conducted by Web360^2, we do not believe that they will change significantly. The communication speed requirement (32 Mbps) specified in the above evaluation is the allowable latency to start playback. An acceptable latency of 10 s is a rather loose setting

and should be set to 1–2 s so that users are unaware of the latency. The resolution of the 360° videos is 1920 × 960, therefore, the image quality after rendering is quite rough. The audio data are also compressed at a low bit rate (64 kbps) from the recorded sound source, and the audio quality is not very good. Considering the resolution of a typical display, the resolution of the 360° videos should be at least 4K, and the audio quality should be at least 256 kbps. Furthermore, as the number of audio objects increased, the audio data will also increase proportionally.

Considering the above, we can roughly estimate that a communication speed of approximately 0.5 Gbps or higher is required to provide a satisfactory viewing experience. This estimated value is feasible even under current conditions with a wired LAN or a high-speed WiFi network. However, it is still challenging to achieve this communication speed with tablets and smartphones in a mobile environment unless special conditions are met. There are high expectations regarding the future evolution of the communication environment.

7.2.2. Required Rendering Performance

Web360^2 renders 360° videos in the viewing direction while drawing an audio visualization object. The number of triangular meshes for displaying the rendered video was 3968, and each audio visualization object consisted of approximately 550 triangular meshes. For the content played by Web360^2, it was necessary to render up to approximately 7500 triangular meshes. By default, A-Frame warns when the number of triangular meshes exceeds 1000. We believe that A-Frame set this criterion so that it would run comfortably on various devices, but Web360^2 exceeded this criterion by a large margin only for rendering video. In tests during the development phase, the slow-rendering behavior on a smartphone (iPhone SE 2020) was improved by omitting object rendering in the blind spot of the audio visualization object (Section 5.3.4). Nevertheless, it is a fact that even in the results of the online survey responses in Section 7.1, there was a tendency for tablet/smartphone evaluations to be worse than PC evaluations. When considered in conjunction with the above evaluation of the required performance of communication speed (Section 7.2.1), it can be said that the communication and rendering performance in the PC environment is satisfactory to some extent. However, the performance of tablets and smartphones is still insufficient.

8. Discussion and Conclusions

In this paper, we proposed the SDM Ontology Version 2, an ontology of spatio-temporal media that can comprehensively and precisely describe events and their semantic information in real space. The SDM Ontology can structurally describe spatio-temporal media, which describes positional and temporal information in 3D space as a whole, and their related equipment, processing, and semantic information. By taking advantage of this feature, the SDM Ontology is expected to become a digital media management framework that comprehensively describes the workflow of digital media from recording, processing, and editing to playback and the related semantic information. The SDM Ontology is an ontology for realizing digital media suitable for providing interactive and highly realistic viewing experiences in AR/VR applications, which are expected to become popular. We have created a dataset based on the recorded data of jazz band and classical orchestra concerts in accordance with SDM Ontology Version 2 and confirmed that the dataset could comprehensively describe a large number of recorded media and a variety of metadata. Furthermore, we developed a web-based VR application (Web360^2) to reproduce concerts by referring to this dataset and showed that it is possible to realize an interactive application that can reproduce a high sense of presence based on the dataset described by the SDM Ontology. In addition, from the results of the subjective evaluation of the Web360^2, it was estimated that the network and rendering speed were generally satisfactory in a PC environment. At the same time, the performance was insufficient for tablets and smartphones, and hence we discussed the performance requirements of these devices.

These results demonstrate that the SDM Ontology Version 2 has the potential to serve as a framework for digital media management, but the scope identified in this work is limited, and more use cases need to be tested by creating data sets to confirm its effectiveness. A website (https://tlab.hongo.wide.ad.jp/sdmo/ (accessed on 1 June 2023)) for the SDM Ontology in Japanese has already been created, but the English version is still in progress and we would like to develop it. In addition, many research issues remain to be addressed. Expanding the definition of the SDM Ontology and reducing the burden of dataset creation are critical issues.

The definition of SDM Ontology version 2 still needs to be revised to realize the specified design policy (Section 3.1). The **Recorder**, **Processor** and **Player** classes still define only essential functions and lack information to describe the workflow from recording to playback. It will be necessary to extend the definitions so that the **Recorder**, **Processor** and **Player** classes can describe the equipment and programs that performed the respective processes and their configuration information. The **CoordinateSystem** class definition is a provisional version, and the detailed definition needs to be completed. Descriptions of the coordinate system origin, the direction of the coordinate axes, and the definition of the derived coordinate system are the essential requirements items described in Section 3.1. This functionality is essential for describing media data and metadata handled in the coordinate transformation processes. By enhancing these definitions, it will be possible to describe a series of workflows from media recording, processing, and editing to playback in a dataset. As a result, it is expected that media management will become more sophisticated and include automation of the data processing and tracking of the processing history.

Another issue is to reduce the workload of dataset creation. As mentioned in Section 4.2, creating a dataset requires recording many video and audio media, processing and editing them, and collecting the metadata to connect them or add semantic information. There is much room for automation in collecting objectively definable metadata such as information about the recording equipment and its settings, location information, media file format, and recording time. It would be possible to automatically monitor the recording status by connecting the recording equipment to a network. It would also be possible to automate the collection of information (e.g., location information) that is difficult to obtain through the equipment alone by adding a device equipped with a sensor. In addition, techniques that recognize objects from video and point cloud data and estimate 3D models of objects have been rapidly evolving in recent years [57,58]. They could also provide an effective solution to this problem. If various metadata such as location information and the type of object can be collected as a byproduct of the object recognition process, this would reduce the burden of metadata collection significantly.

Furthermore, it would be possible to synthesize 360° video from arbitrary viewpoints using 3D models of the objects recognized in this way, greatly reducing the burden of recording and editing video media as a result.

Author Contributions: Conceptualization, H.E. and M.T.; methodology, T.S.; software, T.S. and S.K.; validation, T.S., S.K. and M.T.; formal analysis, T.S.; investigation, T.S.; resources, H.E.; data curation, M.T.; writing—original draft preparation, T.S. and S.K.; writing—review and editing, T.S., S.K. and J.N.; visualization, T.S. and S.K.; supervision, R.A., J.N., M.T. and H.K.; project administration, M.T. and H.E.; funding acquisition, M.T and H.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by JST, CREST Grant Number #JPMJCR22M4; and in part by JSPS KAKENHI under Grant #22H03574.

Data Availability Statement: The SDM Ontology specification can be found at https://sdm.wide.ad.jp/sdmo/ (accessed on 1 June 2023). In addition, a Japanese website with explanations and related information can be found at https://tlab.hongo.wide.ad.jp/sdmo/ (accessed on 1 June 2023). The dataset used in this paper can be accessed at http://sdm.hongo.wide.ad.jp:7200/ (accessed on 1 June 2023). The database application used is GraphDB. The Web360^2 application can be run by opening https://sdm-wg.github.io/web360square-vue/#/ (accessed on 1 June 2023) from a browser on a PC or smart device (iOS, Android). The source code of the application is available on GitHub at https://github.com/sdm-wg/web360square-vue (accessed on 1 June 2023). An introductory video clip can be found at https://www.youtube.com/watch?v=HC3uwE4laS4 (accessed on 1 June 2023).

Conflicts of Interest: The authors declare no conflict of interest regarding the publication of this research article.

References

1. Musmann, H. Genesis of the MP3 audio coding standard. *IEEE Trans. Consum. Electron.* **2006**, *52*, 1043–1049. [CrossRef]
2. Noll, P. MPEG digital audio coding. *IEEE Signal Process. Mag.* **1997**, *14*, 59–81. [CrossRef]
3. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
4. Alvestrand, H.T. *Overview: Real-Time Protocols for Browser-Based Applications*; RFC 8825; Internet Engineering Task Force: Fremont, CA, USA , 2021. [CrossRef]
5. Jesup, R.; Loreto, S.; Tuexen, M. *WebRTC Data Channels*; RFC 8831; Internet Engineering Task Force: Fremont, CA, USA , 2021. [CrossRef]
6. Panjkov, Z.; Draskovic, S.; Pele, Z.; Katona, M. Porting, validation and verification of Dolby Pro Logic II decoder. In Proceedings of the 2011 19th Telecommunications Forum (TELFOR) Proceedings of Papers, Belgrade, Serbia, 22–24 November 2011; pp. 723–726. [CrossRef]
7. Laitinen, M.V.; Pulkki, V. Binaural reproduction for Directional Audio Coding. In Proceedings of the 2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 18–21 October 2009; pp. 337–340. [CrossRef]
8. Brooks, F. What's real about virtual reality? *IEEE Comput. Graph. Appl.* **1999**, *19*, 16–27. [CrossRef]
9. Zyda, M. From visual simulation to virtual reality to games. *Computer* **2005**, *38*, 25–32. [CrossRef]
10. Landone, C.; Sandler, M. 3-D sound systems: A computationally efficient binaural processor. In Proceedings of the IEE Colloquium on Audio and Music Technology: The Challenge of Creative DSP (Ref. No. 1998/470), London, UK, 18 November 1998; pp. 6/1–6/8. [CrossRef]
11. Serafin, S.; Geronazzo, M.; Erkut, C.; Nilsson, N.C.; Nordahl, R. Sonic Interactions in Virtual Reality: State of the Art, Current Challenges, and Future Directions. *IEEE Comput. Graph. Appl.* **2018**, *38*, 31–43. [CrossRef] [PubMed]
12. Kubota, A.; Smolic, A.; Magnor, M.; Tanimoto, M.; Chen, T.; Zhang, C. Multiview Imaging and 3DTV. *IEEE Signal Process. Mag.* **2007**, *24*, 10–21. [CrossRef]
13. Fitzpatrick, W.; Wickert, M.; Semwal, S. 3D sound imaging with head tracking. In Proceedings of the 2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), Napa, CA, USA, 11–14 August 2013; pp. 216–221. [CrossRef]
14. Salisbury, K.; Conti, F.; Barbagli, F. Haptic rendering: Introductory concepts. *IEEE Comput. Graph. Appl.* **2004**, *24*, 24–32. [CrossRef] [PubMed]
15. Narumi, T.; Kajinami, T.; Nishizaka, S.; Tanikawa, T.; Hirose, M. Pseudo-gustatory display system based on cross-modal integration of vision, olfaction and gustation. In Proceedings of the 2011 IEEE Virtual Reality Conference, Singapore, 19–23 March 2011; pp. 127–130. [CrossRef]
16. Tsukada, M.; Ogawa, K.; Ikeda, M.; Sone, T.; Niwa, K.; Saito, S.; Kasuya, T.; Sunahara, H.; Esaki, H. Software defined media: Virtualization of audio-visual services. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7.
17. Atarashi, R.; Sone, T.; Komohara, Y.; Tsukada, M.; Kasuya, T.; Okumura, H.; Ikeda, M.; Esaki, H. The software defined media ontology for music events. In Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music, Monterey, CA, USA, 9 October 2018; pp. 15–23.
18. Raimond, Y.; Abdallah, S.A.; Sandler, M.B.; Giasson, F. The Music Ontology. *ISMIR* **2007**, *2007*, 8.
19. Raimond, Y.; Gängler, T.; Giasson, F.; Jacobson, K.; Fazekas, G.; Reinhardt, S.; Passant, A. The Music Ontology. Available online: http://musicontology.com/ (accessed on 26 April 2020).
20. Raimond, Y.; Abdallah, S. The Timeline Ontology. Available online: http://purl.org/NET/c4dm/timeline.owl (accessed on 26 April 2020).
21. Raimond, Y.; Abdallah, S. The Event Ontology. Available online: http://purl.org/NET/c4dm/event.owl (accessed on 26 April 2020).
22. Davis, I.; Newman, R. Expression of Core FRBR Concepts in RDF. Available online: https://vocab.org/frbr/core (accessed on 26 April 2020).

23. Brickley, D.; Miller, L. FOAF Vocabulary Specification 0.99. Available online: http://xmlns.com/foaf/spec/ (accessed on 26 April 2020).
24. Fazekas, G.; Sandler, M.B. The Studio Ontology Framework. In Proceedings of the ISMIR, Miami, FL, USA, 24–28 October 2011; pp. 471–476.
25. Wilmering, T.; Fazekas, G.; Sandler, M.B. The Audio Effects Ontology. In Proceedings of the ISMIR, Curitiba, Brazil, 4–8 November 2013; pp. 215–220.
26. Wilmering, T.; Fazekas, G. The Audio Effect Ontology. Available online: https://w3id.org/aufx/ontology/1.0# (accessed on 26 April 2020).
27. Kolozali, S.; Barthet, M.; Fazekas, G.; Sandler, M.B. Knowledge Representation Issues in Musical Instrument Ontology Design. In Proceedings of the ISMIR, Miami, FL, USA, 24–28 October 2011; pp. 465–470.
28. Kolozali, S.; Fazekas, G.; Barthet, M.; Sandler, M. A framework for automatic ontology generation based on semantic audio analysis. In Proceedings of the Audio Engineering Society Conference: 53rd International Conference: Semantic Audio, London, UK, 26–29 January 2014.
29. Allik, A.; Fazekas, G.; Sandler, M.B. An Ontology for Audio Features. In Proceedings of the ISMIR, New York City, NY, USA, 7–11 August 2016; pp. 73–79.
30. Fazekas, G.; Allik, A. Audio Features Ontology. Available online: https://semantic-audio.github.io/afo/ (accessed on 26 April 2020).
31. Ceriani, M.; Fazekas, G. Audio Commons ontology: A data model for an audio content ecosystem. In *The Semantic Web—ISWC 2018, Proceedings of the 17th International Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 20–35.
32. Fazekas, G.; Ceriani, M. The Audio Commons Ontology. Available online: https://w3id.org/ac-ontology/aco# (accessed on 26 April 2020).
33. Turchet, L.; Antoniazzi, F.; Viola, F.; Giunchiglia, F.; Fazekas, G. The Internet of Musical Things Ontology. *J. Web Semant.* **2020**, *60*, 100548. [CrossRef]
34. Antoniazzi, F. Internet of Musical Things Ontology (IoMusT). Available online: https://fr4ncidir.github.io/IoMusT/ (accessed on 26 April 2020).
35. Chang, S.F.; Sikora, T.; Purl, A. Overview of the MPEG-7 standard. *IEEE Trans. Circuits Syst. Video Technol.* **2001**, *11*, 688–695. [CrossRef]
36. Man, J.F.; Yang, L.M.; Wu, Z.H.; Xu, G.W. Research on multimedia ontology bridging "semantic gap" between perceivable world and conceptual world. In Proceedings of the 2008 First IEEE International Conference on Ubi-Media Computing, Lanzhou, China, 31 July–1 August 2008; pp. 100–105. [CrossRef]
37. et al., A.M. MOWL: An ontology representation language for web-based multimedia applications. *ACM Trans. Multimed. Comput. Commun. Appl.* **2013**, *10*, 1–21. [CrossRef]
38. Sujal Subhash Wattamwar, H.G. Spatio-temporal query for multimedia databases. In Proceedings of the 2nd ACM workshop on Multimedia Semantics, International Multimedia Conference, Vancouver, Canada, 31 October 2008; pp. 48–55.
39. Choi, C.; Wang, T.; Esposito, C.; Gupta, B.B.; Lee, K. Sensored Semantic Annotation for Traffic Control Based on Knowledge Inference in Video. *IEEE Sens. J.* **2021**, *21*, 11758–11768. [CrossRef]
40. Duckham, M.; Gabela, J.; Kealy, A.; Khan, M.; Legg, J.; Moran, B.; Rumi, S.K.; Salim, F.D.; Sharmeen, S.; Tao, Y.; et al. Explainable spatiotemporal reasoning for geospatial intelligence applications. *Trans. GIS* **2022**, *26*, 2455–2479. [CrossRef]
41. *23008-3:2019*; Information Iechnology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 3: 3D Audio. ISO/IEC: Washington, DC, USA, 2019.
42. Beack, S.; Sung, J.; Seo, J.; Lee, T. MPEG Surround Extension Technique for MPEG-H 3D Audio. *ETRI J.* **2016**, *38*, 829–837. [CrossRef]
43. Dolby Laboratories. *Dolby Atmos® Specifications*; Dolby Laboratories: San Francisco, CA, USA, 2015; Isuue 3.
44. Dolby Laboratories. *Dolby Atmos® Home Theater Installation Guidelines*; Technical report; Dolby Laboratories: San Francisco, CA, USA, 2018.
45. DTS, Inc. Home Theater Sound Gets Real. Available online: https://dts.com/dtsx (accessed on 5 September 2020).
46. Auro Technologies. *AUROMAX® Next Generation Immersive Sound System*; Technical Report; Auro Technologies: Ayer Keroh, Malaysia, 2015.
47. Herre, J.; Hilpert, J.; Kuntz, A.; Plogsties, J. MPEG-H 3D audio—The new standard for coding of immersive spatial audio. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 770–779. [CrossRef]
48. Ricoh Company, Ltd. 360-Degree Camera RICOH THETA. Available online: https://theta360.com/ (accessed on 5 September 2020).
49. Kasuya, T.; Tsukada, M.; Komohara, Y.; Takasaka, S.; Mizuno, T.; Nomura, Y.; Ueda, Y.; Esaki, H. LiVRation: Remote VR live platform with interactive 3D audio-visual service. In Proceedings of the 2019 IEEE Games, Entertainment, Media Conference (GEM), New Haven, CT, USA, 18–21 June 2019; pp. 1–7.
50. Wang, R.; Peethambaran, J.; Chen, D. LiDAR Point Clouds to 3-D Urban Models: A Review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 606–627. [CrossRef]

51. Turner, E.; Cheng, P.; Zakhor, A. Fast, Automated, Scalable Generation of Textured 3D Models of Indoor Environments. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 409–421. [CrossRef]
52. Ikeda, M.; Sone, T.; Niwa, K.; Saito, S.; Tsukada, M.; Esaki, H. New recording application for software defined media. In Proceedings of the Audio Engineering Society Convention 141, Los Angeles, CA, USA, 29 September–2 October 2016.
53. Kato, S.; Ikeda, T.; Kawamorita, M.; Tsukada, M.; Esaki, H. Web360^2: An Interactive Web Application for viewing 3D Audio-visual Contents. In Proceedings of the 17th Sound and Music Computing Conference (SMC), Torino, Italy, 24–26 June 2020.
54. Fernández, M.; Overbeeke, C.; Sabou, M.; Motta, E. What makes a good ontology? A case-study in fine-grained knowledge reuse. In *The Semantic Web, Proceedings of the Fourth Asian Conference, ASWC 2009, Shanghai, China, 6–9 December 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 61–75.
55. Poveda-Villalón, M.; Gómez-Pérez, A.; Suárez-Figueroa, M.C. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *Int. J. Semant. Web Inf. Syst.* **2014**, *10*, 7–34. [CrossRef]
56. Garijo, D. WIDOCO: A wizard for documenting ontologies. In *The Semantic Web—ISWC 2017, Proceedings of the 16th International Semantic Web Conference, Vienna, Austria, 21–25 October 2017*; Springer: Cham, Switzerland, 2017; pp. 94–102.
57. Mian, A.; Bennamoun, M.; Owens, R. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1584–1601. [CrossRef] [PubMed]
58. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928. [CrossRef]