MPEG-7-Aligned Spatiotemporal Video Annotation and Scene Interpretation via an Ontological Framework for Intelligent Applications in Medical Image Sequence and Video Analysis

by

Leslie Frank Sikos

Thesis
Submitted to Flinders University
for the degree of

Doctor of Philosophy

College of Science and Engineering
5 March 2018

Contents

Preface

In the past decades, the rapid growth of medical technologies has dramatically increased the amount of multimedia data generated in medical imaging, which urges the development of efficient automated mechanisms for processing video contents. This is a big challenge due to the huge gap between what software agents can obtain from signal processing and what practitioners can comprehend based on cognition, expert knowledge, and experience. Automatically extracted low-level video features typically do not correspond to concepts, such as human organs and medical devices, and procedural events depicted in medical videos. To narrow the Semantic Gap, the depicted concepts and their spatial relations can be described in a machine-interpretable form using formal definitions from structured data resources. Rule-based mechanisms are efficient in describing the temporal information of actions and video events. The fusion of these structured descriptions with textual and, where available, audio descriptors is suitable for the machine-interpretable spatiotemporal annotation of complex medical video scenes. The resulting structured video annotations can be efficiently queried manually or programmatically, and can be used in scene interpretation, video understanding, and content-based video retrieval.

Despite the advantages and potential of Semantic Web technologies in this field, previous approaches failed to exploit rich semantics due to their reliance on proprietary solutions, incorrect and incomplete knowledge-based abstraction of medical domains, and limited temporal and spatial segmentation capacity. Because of the successful semantic implementations seen in the literature for a variety of knowledge domains and applications, there is no doubt that it is worthwhile to apply formal knowledge representation and automated reasoning to medical multimedia resources. Interestingly enough, while medical image semantics have been extensively researched, medical video semantics have long been neglected, and research efforts on video semantics have mainly been focusing on news videos, sports videos, and surveillance videos only.

In this work, novel formalisms and ontologies are proposed to set new directions in standards-aligned semantic video annotation and spatiotemporal reasoning. These can be applied to knowledge domains of practical importance, among which the medical domain has been selected as the primary domain for this thesis, but other domains are also mentioned and demonstrated.

List of Figures

List of Tables

List of Listings

Declaration

I certify that without acknowledgment this thesis does not incorporate any material previously submitted for a degree or diploma in any university, and that to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Leslie Frank Sikos

Adelaide, 5 March 2018

Acknowledgements

The research work presented in this thesis has been carried out between 2015 and 2017 at the Centre for Knowledge and Interaction Technologies, Flinders University, Australia, under the supervision of Professor David M. W. Powers, Dr. Trent W. Lewis, and Dr. Richard Leibbrandt.

I wish to express my sincere gratitude to my supervisors, particularly my principal supervisor, Professor Powers, for the thoughtful supervision and steady support.

I would also like to thank the reviewers, Professor Antoine Doucet and the anonymous reviewer, for their helpful comments and constructive criticism that ultimately contributed to the overall quality of the thesis.

Chapter 1

Introduction and Motivation

> “I never came upon any of my discoveries
> through the process of rational thinking.”
>
> — *Albert Einstein*

Video contents are inherently complex and ambiguous, and comprehensible due to experience and knowledge typical of humans. Without formal descriptions to process, software agents often struggle to interpret the meaning of video scenes. Approaches that work well for a large share of images, such as textual descriptors and technical metadata, are very limited when it comes to video semantics, particularly in terms of context, temporal information, and multimodality. Signal processing captures low-level features that are useful for classification or finding a particular concept for which machine learning is pretrained; however, they do not directly correspond to the meaning of the video content.

1.1 The Limitations of Video Metadata

Common *technical metadata* implemented in video files, such as duration, frame width and height, and frame rate, and *rights metadata*, such as licensing, do not convey information about the visual content and the meaning of video scenes. In fact, even *descriptive metadata*, which is the closest metadata type to content description, provide information such as title, keywords, and genre only [1], which is rather limited in describing audiovisual contents.

Keywords, tags, or labels, categories, genre, rating, and age rating, are frequently used on video sharing portals, which can be useful for video classification, but not necessarily for video understanding. A textual description, such as the plot, helps humans understand the story the video tells, but it is less useful for software agents, although they might retrieve some keywords from it via natural language processing. Recent efforts from the signal processing and natural language processing communities attempted to employ deep learning to automatically generate a complete natural sentence for describing video contents, called *video captioning* [2], although the output is often limited, prone to errors, and overly generic.

1.2 The Limitations of Feature Descriptors: the Semantic Gap

Low-level features, such as loudness and motion trajectory, which are automatically extracted from audio and video signals, provide information that might be useful for video classification, object matching with a reference object (even if the object has been rotated and/or scaled) and object tracking. However, they are not suitable for efficient scene interpretation and video understanding, because they, similar to video metadata, do not correspond directly to the depicted concepts and events. For example, detecting red as a dominant color in a frame of a clinical video does not provide information about the meaning of the visual content [3], which might depict anything with red in it, and a software agent cannot make an assumption about the depicted body part, organ, or medical device (see Fig. 1.1), even though skin hue may be useful in tasks such as face detection.

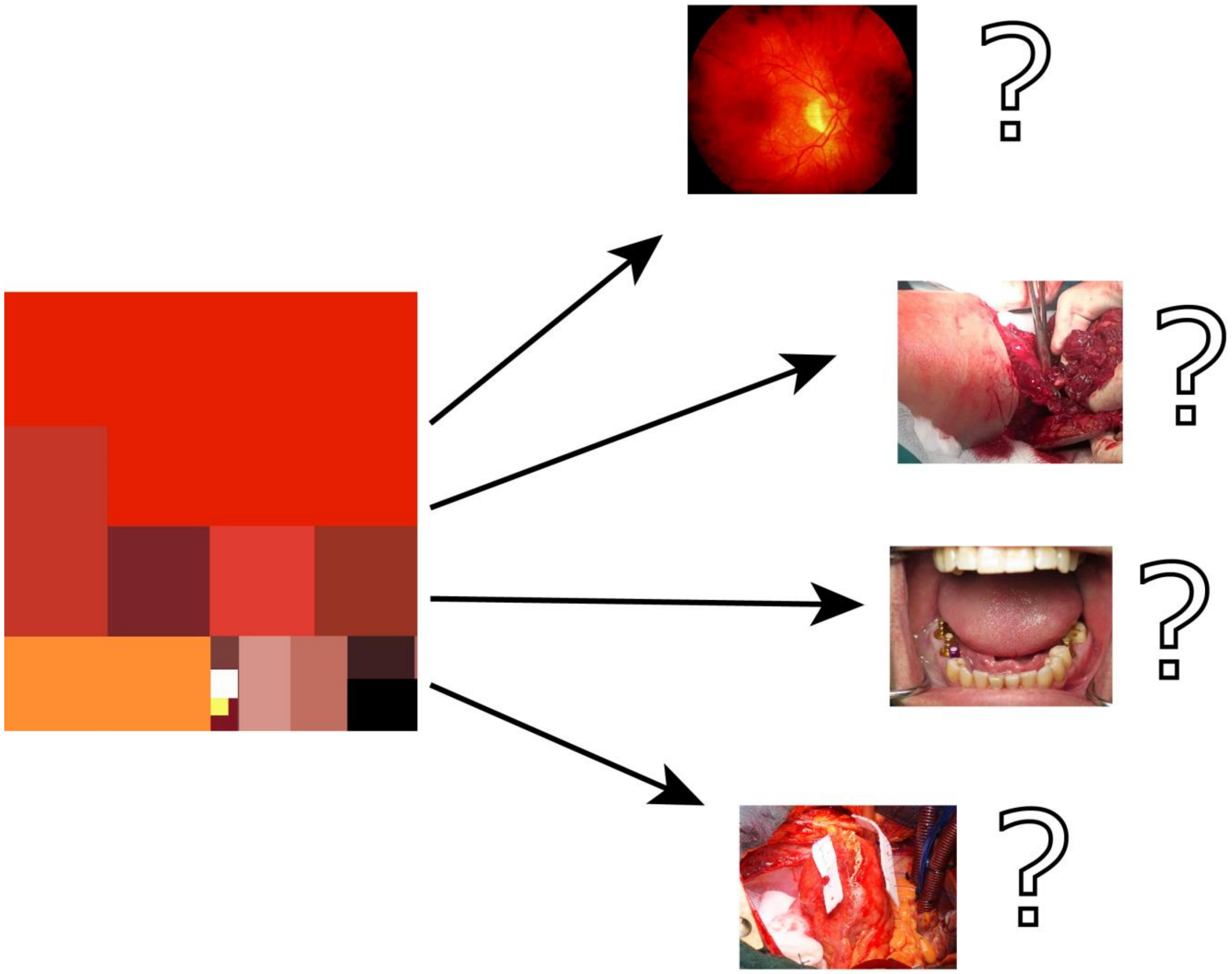


Figure 1.1 Many automatically extracted low-level features, such as dominant colors (left), do not hold enough information to interpret the visual content (right)

This *semantic gap* between what computers can interpret using automatically extracted low-level features and what humans understand based on cognition, knowledge, and experience makes video understanding particularly challenging.[1] Training from a few hundred or few thousand clips provides a very limited recognition capability, which cannot compete with years or decades of life experience and learning typical to humans. Training provides information for particular viewing angles only for the represented 3D space, although scale-/rotation-invariant features in 2D space can be used for object tracking in videos. For video processing algorithms, occlusion poses a real challenge, while recognizing partially covered objects is often very easy for humans. There are very few meth-

---

[1] Sikos, L. F. (2017) The Semantic Gap. In: Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5

ods for complex video event detection, while humans understand even nonlinear narratives, such as extensive flashbacks and flash-forwards. On top of this, if the noise-signal ratio falls below a threshold, algorithms perform poorly. For these reasons, video understanding is often infeasible even without time constraints, let alone in near-real time or real time.

1.2.1 Common Visual Descriptors

Common *visual descriptors* capture perceptual features, such as color, texture, and shape, which can be useful for frame-level video annotation, and motion, which corresponds to camera movements and moving objects. The four main categories of visual descriptors are the following:

- *Color descriptors*. The *dominant color descriptor* specifies a set of dominant colors for an image (typically 4–6 colors), and considers the percentage of image pixels each color is used in, as well as the variance and spatial coherence of the colors. The *color structure descriptor* encodes local color structure by utilizing a structuring element, visiting all locations in an image, and summarizing the frequency of color occurrences in each structuring element. The *color layout descriptor* represents the spatial distribution of colors. The *scalable color descriptor* is a compact color histogram descriptor represented in the HSV color space[2] and encoded using Haar transform.
- *Texture descriptors*. The *homogeneous texture descriptor* characterizes the regional texture using local spatial frequency statistics extracted by Gabor filter banks. The *texture browsing descriptor* represents a perceptual characterization of texture in terms of regularity, coarseness, and directionality as a vector. The *edge histogram descriptor* represents the local edge distribution of an image as a histogram that corresponds to the frequency and directionality of brightness changes in the image.
- *Shape descriptors*. The *region-based shape descriptor* represents the distribution of all interior and boundary pixels that constitute a shape by decomposing the shape into a set of basic

---

[2] Low-level color descriptors often use non-linear color spaces, because those are closely related to the human perception of color. One such color space is HSV, which represents colors with hue–saturation–value triplets. Another one is HMMD, which holds the hue, the maximum and the minimum of RGB values, and the difference between the maximum and minimum values. MPEG-7 supports both of these color spaces, in addition to the widely used RGB, YCbCr, and monochrome color spaces.

functions with various angular and radial frequencies using angular-radial transformation, a two-dimensional complex transform defined on a unit disk in polar coordinates. The *contour-based shape descriptor* represents a closed two-dimensional object or region contour in an image or video. The *3D shape descriptor* is a representation-invariant description of three-dimensional mesh models, expressing local geometric attributes of 3D surfaces defined in the form of shape indices calculated over a mesh using a function of two principal curvatures.

- *Motion descriptors*. The *camera motion descriptor* represents global motion parameters, which characterize a video scene in a particular time by providing professional video camera movements, including moving along the optical axis (dolly forward/backward), horizontal and vertical rotation (panning, tilting), horizontal and vertical transverse movement (tracking, booming), change of the focal length (zooming), and rotation around the optical axis (rolling). The *motion activity descriptor* indicates the intensity and direction of motion, and the spatial and temporal distribution of activities. The *motion trajectory descriptor* represents the displacement of objects over time in the form of spatiotemporal localization with positions relative to a reference point and described as a list of vectors. The *parametric motion descriptor* describes the global motion of video objects using a classic parametric model (translational, scaling, affine, perspective, quadratic).

1.2.2 Common Audio Descriptors

The audio channel of video files can be described with temporal, spectral, cepstral, and perceptual audio descriptors.

- *Temporal audio descriptors*. The *energy envelope descriptor* represents the root mean square of the mean energy of the audio signal, which is suitable for silence detection. The *zero crossing rate descriptor* represents the number of times the signal amplitude undergoes a change of sign, which is used for differentiating periodic signals and noisy signals, such as to determine whether the audio content is speech or music. The *temporal waveform moments descriptor* represents characteristics of waveform shape, including temporal centroid, width, asymmetry, and flatness. The *amplitude modulation descriptor* describes the tremolo of a sustained sound (in the frequency range 4–8 Hz) or the graininess or roughness of a sound (between 10–40Hz). The *autocorrelation coefficient descriptor* represents the spectral distribution of the audio signal over time, which is suitable for musical instrument recognition.

- *Spectral audio descriptors*. The *spectral moments descriptor* corresponds to core spectral shape characteristics, such as spectral centroid, spectral width, spectral asymmetry, and spectral flatness, which are useful for determining sound brightness, music genre, and categorizing music by mood. The *spectral decrease descriptor* describes the average rate of spectral decrease with frequency. The *spectral roll-off descriptor* represents the frequency under which a predefined percentage (usually 85–99%) of the total spectral energy is present, which is suitable for music genre classification. The *spectral flux descriptor* represents the dynamic variation of spectral information computed either as the normalized correlation between consecutive amplitude spectra or the derivative of the amplitude spectrum. The *spectral irregularity descriptor* describes the amplitude difference between adjacent harmonics, which is suitable for the precise characterization of the spectrum, such as for describing individual frequency components of a sound. The descriptors of *formants parameters* represent the spectral peaks of the sound spectrum of voice, and are suitable for phoneme and vowel identification.
- *Cepstral audio descriptors*. Cepstral features are used for speech and speaker recognition and music modeling. The most common cepstral descriptors are the *mel-frequency cepstral coefficient descriptors*, which approximate the psychological sensation of the height of pure sounds, and are calculated using the inverse discrete cosine transform of the energy in predefined frequency bands.
- *Perceptual audio descriptors*. The *loudness descriptor* represents the impression of sound intensity. The *sharpness descriptor*, which corresponds to a spectral centroid, is typically estimated using a weighted centroid of specific loudness. The *perceptual spread descriptor* characterizes the timbral width of sounds, and is calculated as the relative difference between the specific loudness and the total loudness.
- *Specific audio descriptors*. The *odd-even harmonic energy ratio descriptor* represents the energy proportion carried by odd and even harmonics. The descriptors of *octave band signal intensities* represent the power distribution of the different harmonics of music. The *attack du-*

ration descriptor represents how quickly a sound reaches full volume after it is activated, and is used for sound identification. The *harmonic-noise ratio descriptor* represents the ratio between the energy of the harmonic component and the noise component, and enables the estimation of the amount of noise in the sound. The *fundamental frequency descriptor*, also known as the *pitch descriptor*, represents the inverse of the period of a periodic sound.

1.2.3 Common Spatiotemporal Feature Descriptors, Feature Aggregates, and Feature Statistics

Local spatiotemporal feature descriptors, aggregates, and statistics capture aspects of both appearance and motion. In medical video analysis, they are suitable for recognizing medical procedures and tracking body parts, organs, and medical devices. The most common ones are the following:

- *SIFT (Scale-invariant feature transform)*, as its name suggests, is a scale-invariant feature descriptor [4], which is suitable for, among other things, object recognition and video tracking.
- The *cuboid descriptor* is a spatiotemporal interest point detector, which finds local regions of interest in space and time (cuboids) to be used for behavior recognition [5].
- The *histogram of oriented gradients (HOG)* describes the number of occurrences of gradient orientation in localized portions of images and video frames [6], and is one of the most powerful feature statistics.
- *Motion boundary histograms (MBH)* represent local orientations of motion edges by emulating static image HOG descriptors.
- *The speeded up robust features (SURF)* feature descriptor is based on the sum of the Haar wavelet response around the point of interest [7]. SURF is suitable for locating and recognizing objects and people, reconstructing 3D scenes, extracting points of interest, and tracking objects.

1.3 Machine Learning Approaches for Multimedia Understanding

Low-level multimedia descriptors are typically fed into a recognition system powered by supervised learning, such as SVM classifiers (support vector machines), which look for an optimal hyperplane to find the most probable interpretation, such as via a Lagrangian optimization problem [8] (see Equation 1.1).

$$\min_{\beta,\beta_0} L(\beta) = \frac{1}{2}\|\beta\|^2 \text{ s. t. } y_i(\beta^T x_i + \beta_0) \geqslant 1 \text{ for } \forall i \qquad \text{(Eq. 1.1)}$$

where $\beta^T x_i + \beta_0$ represents a hyperplane, β the weight vector of the optimal hyperplane, β_0 the bias of the optimal hyperplane, y_i the labels of the training examples, and x_i the training examples. Since there is an infinite number of potential representations of the optimal hyperplane, there is a convention to choose the one where $\beta^T x_i + \beta_0 = 1$. Once a classifier is trained on images depicting the object of interest (positive examples) and on images that do not (negative examples), it can make decisions regarding the probability of object matching in other images (i.e., object recognition). Complex events of unconstrained real-world videos can also be efficiently detected and modeled using an intermediate level of semantic representation using support vector machines [9].

Bayesian networks are used in content-based image understanding while integrating low-level features and high-level semantics [10]. By using a set of images for training to derive simple statistics for the conditional probabilities, Bayesian networks usually provide more relevant concepts than discriminant-based systems, such as neural networks. Machine learning in image classification may fuse global image classification and local, region-based image classification data [11].

Other machine learning models used for multimedia understanding include the following:

- Hidden Markov models (HMM): video shot classification labels can be performed using decision trees, and a Hidden Markov Model trained by using these labels as observations for video segmentation [12].
- K-nearest neighbor (k-NN): the semantic analysis of videos can be performed using kernel discrimination for sparse representation features and weighted k-NN, which is suitable for the classification of video features with nonlinear relations while tolerating noise and interference [13].
- Gaussian mixture models (GMM): video scene classification with Gaussian mixture models can automatically assign a scene to a semantic category, and can handle outliers and noise [14]. Extending Bag-of-Visual-Words (BoW) to a probabilistic framework using Gaussian mixture models has proven to be effective for semantic indexing of videos [15].

- Logistic regression: Kernel Logistic Regression (KLR), the kernel version of logistic regression, is a proven classifier, and can be implemented for video genre classification [16].
- Adaboost: Adaboost-based multi-classifiers can be used in a fusion algorithm to extract semantic concepts from videos [17].

1.4 Motivation: Multimedia Semantics in the Medical Domain

Digital health, the convergence of digital and healthcare-related technologies to enhance the prevention, prediction, diagnosis, and management of medical conditions, is among the biggest and fastest growing industries of the 21st century. Information and communication technology (ICT) is now utilized in health services (eHealth), even on mobile devices, such as in the form of medical smartphone apps (mobile health or mHealth), or remotely (telehealth, telemedicine, and connected health). Wearable medical devices are widely used with the Internet of Things (IoT) to significantly improve patient outcomes and reduce healthcare costs.

Many health information systems employ machine-readable data to store and process the corresponding medical knowledge and bioinformatics data in general, and personalized health information in particular. Several of these are generated from, or complemented by, medical imaging. The visualization of body parts and internal organs of the human body, as well as medical devices and procedures, can be used for clinical analysis, medical intervention, and medical decision support.

Some of the most common medical scans include magnetic resonance imaging (MRI), X-ray, dual-energy X-ray absorptiometry (DXA), ultrasound scans, and computed tomography (CT), including X-ray CT, positron emission tomography (PET CT), and single-photon emission computed tomography (SPECT CT), all of which produce medical multimedia materials, such as medical images, medical image sequences, and medical videos (see Fig. 1.2).

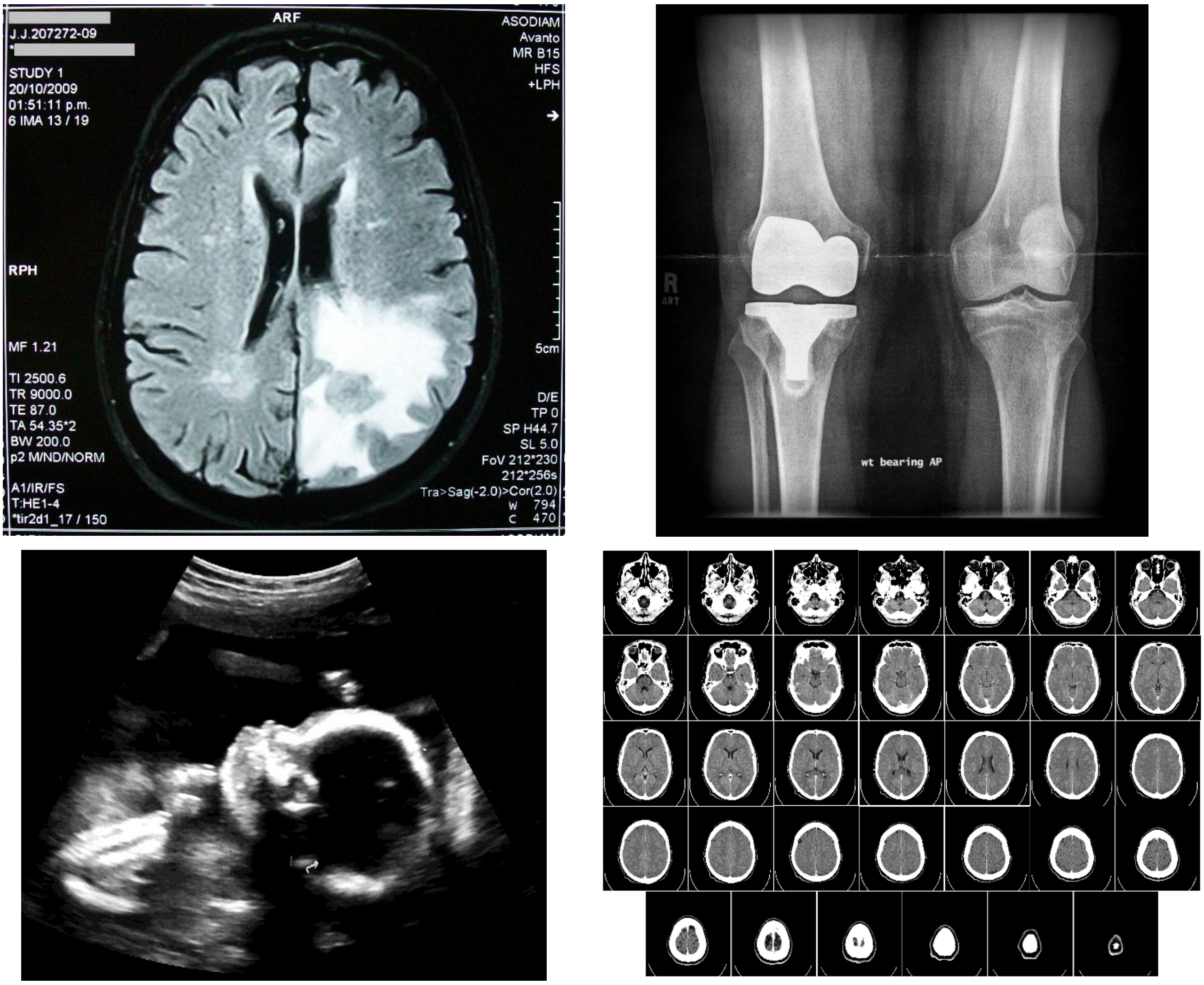


Figure 1.2 Common medical scans, from top left: MRI,[3] X-ray,[4] ultrasound,[5] and CT[6]

Automatically extractable low-level image features, such as dominant color and color layout, are inadequate for representing medical image semantics [18]. Bridging the semantic gap between automatically extracted low-level visual features and high-level semantics, which correspond to the meaning of the image or video content, has been a well-known challenge in medical imaging for decades [19]. *Semantic Web* technologies have been used in healthcare applications, including medical imaging [20], since their introduction in the 2000s [21]. The number of structured biomedical datasets based on Semantic Web standards, which formally describe protein sequences, drugs,

---

[3] https://upload.wikimedia.org/wikipedia/commons/8/8a/MRI_brain_tumor.jpg

[4] http://rmehs.fullerton.edu/_documents/graphics/knee/xray.jpg

[5] http://ultrasoundtechnicianexpert.com/wp-content/uploads/2013/07/ultrasound-image.jpg

[6] https://upload.wikimedia.org/wikipedia/commons/5/50/Computed_tomography_of_human_brain_-_large.png

genes, diseases, and chemical compounds, is constantly increasing [22]. *Big Data* tools and technologies are utilized in healthcare applications to store, manage, and process medical data of high volume, velocity, and variety. Big Semantic Web analysis has the potential to provide actionable information that supports better decisions for personalized medicine, advances clinical decision-making, improves health outcomes, and minimizes healthcare costs [23].

Intelligent medical imaging applications can utilize the annotation of medical images and videos with formally defined terms using Semantic Web standards for the efficient indexing and content-based retrieval of medical multimedia resources. The expert knowledge for a wide range of medical domains is already represented formally. This can be combined with the low-level descriptors automatically extracted from medical images, image sequences, and videos. This makes it possible to utilize the expertise of hundreds of medical professionals for a particular medical treatment, automatically identify image features that indicate a disease, and provide medical decision support for practitioners. The corresponding data can be efficiently reused, shared, and linked to similar cases or medical conditions, and queried manually and programmatically with complex queries by medical researchers, practitioners, and e-patients alike.

Semantic relatedness (or semantic similarity) often provides context, which can be retrieved from the relationships between depicted concepts, and is very useful for the semantic enrichment of multimedia resources. For example, corpora of concepts frequently depicted together can be used to set likelihood values for the correctness of concept mapping in images and videos, while scene detection can benefit from combining semantics and visual feature similarity [24]. The most common semantic relatedness measures are based on one of the following:

- *Taxonomical structure.* These measures exploit the geometric model provided by concept hierarchies.
- *Information content.* These measures represent the amount of information provided by a particular term on its probability of appearance in a corpus.
- *Context vector relatedness.* These measures exploit the hypothesis that words are similar if their contexts are similar.

Because path-based similarity measures cannot exploit the detailed taxonomical knowledge available in large biomedical ontologies, a semantic relatedness measure has also been introduced specially for large biomedical ontologies, which compiles all the available taxonomic knowledge [25].

So far, the majority of multimedia semantics research has focused on image understanding, and to a far lesser extent, on audio and video semantics. Since the term multimedia refers to various combinations of two or more content forms, including text, audio, image, animation, video, and interactive content, images alone are not multimedia contents, but can be components of multimedia contents. Yet the literature often uses the term "multimedia" to refer to techniques and tools that are limited to capturing image semantics.

During a thorough literature review, serious limitations have been found in ontologies used for video annotation and spatiotemporal reasoning; the most important of these include design issues, poor conceptualization, implementation issues, low expressivity, and undecidability. Many of these, such as the underutilization of mathematical constructors available in the representation and implementation languages, have not caught the attention of any ontology engineer before. While some might argue the importance of certain aspects and features of ontologies over others, without a holistic approach it is impossible to create multimedia ontologies with favorable characteristics, standards alignment, and high reasoning potential.

The goal of this PhD is to overcome these limitations by introducing novel formalisms, ontologies, and methodologies, and applying them to videos, in particular videos in the medical domain, and to confirm the research results through evaluation and case studies.

1.5 Summary

This chapter introduced the reader to the importance of video semantics, the potential of medical video semantics, and the limitations of technical video metadata and feature descriptors. It described the most common low-level audio and visual descriptors, and common machine learning approaches used in multimedia understanding. It also revealed the motivation and main goal of this work, along with its practical importance.

Chapter 2

Medical Video Semantics

> “The purpose of abstraction is not to be vague,
> but to create a new semantic level in which
> one can be absolutely precise.”
>
> — *Edsger W. Dijkstra*

While technical video metadata, automatically extracted low-level video features, and category labels can be used for video classification and keyphrase-based video indexing, they cannot be used for scene interpretation and content-based video retrieval, because they do not correspond directly to video events and concepts depicted in videos.

2.1 Challenges in Medical Video Interpretation

Inherently, video interpretation is far more complex than image understanding. Knowledge acquisition in video content analysis involves the extraction of spatial and temporal information, which can be used for a wide range of applications including, but not limited to, face recognition, object tracking, dynamic masking, tamper detection, abandoned luggage detection, automated number plate recognition, and lane departure warning. Without context and semantics (meaning), however, even basic tasks are limited or infeasible. For example, a transplant cannot be differentiated from the original organ without seeing the preceding events in the surgery video. As a consequence, video event identification alone is inadequate for classification. Automated scene interpretation and video understanding rely on the formal representation of human knowledge, which is suitable for automated subtitle generation, intelligent medical decision support, and so on. While computers can be trained to recognize features based on signal processing, they cannot interpret sophisticated visual contents without additional semantics (see Table 1.1).

Table 1.1 Major differences between human understanding and computer interpretation of medical videos

Practitioners	Software Agents
Intelligence	
Video understanding is straightforward in many cases, often even in real time, although shots of complex conditions or procedures are often watched multiple times to be fully understood	Overwhelming amount of information to process, algorithms and methods are often inadequate, making video understanding infeasible even without time constraints, let alone in near-real time or real time
Context is largely understood from metadata and textual descriptors	Potential interpretations are challenging to handle; metadata, if available, can be combined with concepts mapped to common sense knowledge bases or ontologies
Visual content is understood, even if the footage is grayscale	Automatically extractable features and their statistics convey no information about the actual visual content; nothing is self-explanatory
Years or decades of professional experience and expert knowledge obtained through studying make it possible to recognize any human organ, and hundreds of medical conditions and procedures	Machine learning relying on training from a few hundred video clips has a very limited recognition capacity
General understanding of how the universe works (e.g., common sense, naïve physics)	Only tiny, isolated representations of the world are formalized, therefore unconstrained video scenes cannot be processed efficiently

Table 1.1 (Cont'd) Major differences between human understanding and computer interpretation of medical videos

Practitioners	Software Agents
Intelligence	
The human mind and eyes are adaptive and recognize organs and medical devices moving to or being in darkness, or in noisy or less detailed recordings (e.g., sonographic video)	If the noise-signal ratio falls below a threshold, algorithms perform poorly
Spatial Information	
3D projected to 2D can be interpreted by stereo vision: planes of graphical projection with multiple vanishing points are understood, which enables perspective viewing	Most videos have no depth information, although proprietary and standardized 3D recording and playback mechanisms are available; RGB-D and Kinect depth sensors can provide depth information
3D objects are recognized from most angles	Training provides information for particular viewing angles only—recognition from different viewpoints is problematic; scale-/rotation-invariant features are used for object tracking in videos
Many partially covered organs are relatively easily recognized by feature, such as shape, or anatomic location and neighboring organs (e.g., heart and lungs)	Occlusion is problematic
Temporal Information	
Continuity and video events are understood even in the case of nonlinear narratives with extensive flashbacks and flashforwards	Very few mechanisms exist for complex event detection; videos are usually compressed using lossy compression, therefore only certain frames can be used; no information can be obtained on complex events from signal processing

Table 1.1 (Cont'd) Major differences between human understanding and computer interpretation of medical videos

Practitioners	Software Agents
Information Fusion	
Seamless/straightforward understanding of simultaneous multimodal information playback (e.g., video with audio and subtitle(s), hypervideo)	Information fusion is desired, which needs more research
Audio channel is understood relatively easily and conveys additional information	Algorithms for detecting distinct noises (e.g., screaming) are available; complex audio analysis is a challenge
Subtitles and closed captions can be read by most humans and convey additional information	Text-based, timestamped subtitle files can be processed very efficiently, however, incorporating the obtained information into higher-level video understanding is still challenging

2.2 From Video Metadata to Medical Video Semantics

A wide variety of video types are used in medicine, including, but not limited to, medical training videos, 3D animations of medical devices and equipment, videos about medical procedures and treatments, medical educational videos for patients, patient testimonial videos, and virtual tours about onsite facilities of hospitals.

Medical videos hold much more information than medical images and even medical image sequences. Surgery videos provide rich semantics for medical training, medical examination videos help better understand the underlying health issues, while the CGI-based simulation of medical procedures may help surgeons in decision-making. For example, endoscopic videos enable a higher level of understanding compared to endoscopic images by providing temporal information and continuity (see Fig. 2.1).

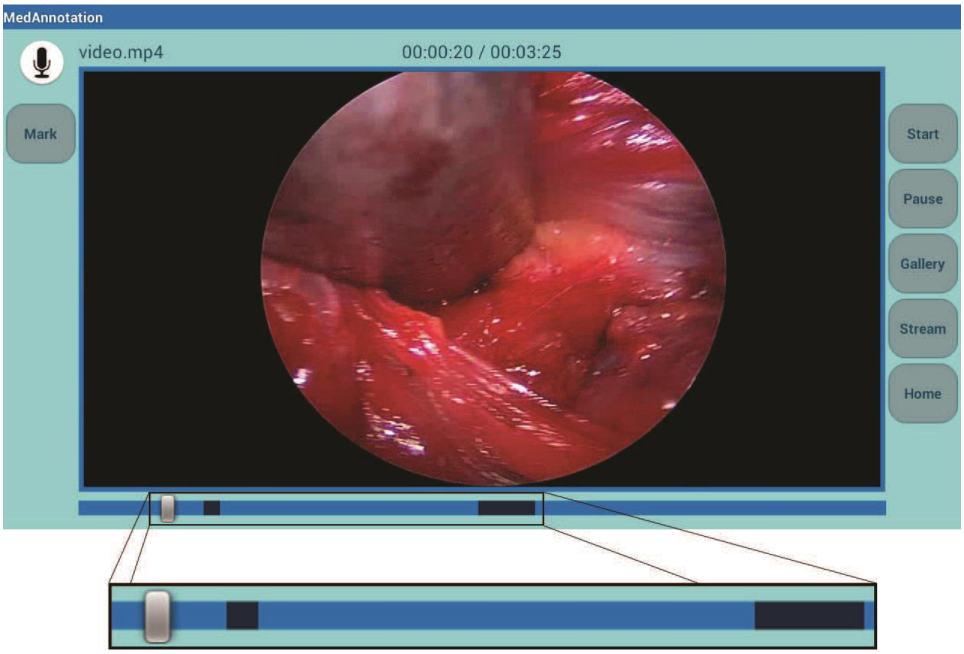


Figure 2.1 Timeline represented as a slider under an endoscopic video [26]

Medical videos can also be used in telemedicine for the remote diagnosis and treatment of patients [27, 28] (see Fig. 2.2).

Figure 2.2 Medical multimedia is well utilized in telemedicine. Image by athenahealth[1]

Despite the importance of video analysis in healthcare and its benefits over medical images, medical video semantics are researched to a far less extent than medical image semantics. Keywords, tags, and category labels are frequently used to classify medical videos, however, for the most part,

---

[1] https://insight.athenahealth.com/cost-benefit-telemedicine-icu

they are inadequate for scene interpretation and video understanding. A textual description, such as a video summary, helps medical professionals understand the scenario depicted in the video, but it is less useful for software agents.

The meaning of medical video contents can be described only with rich video semantics, for example context, possible interpretations, depicted and related medical concepts, such as medical condition or human organs, the depicted medical procedure, etc.

Frame-level video semantics include depicted concepts, whose formal definitions and properties can be retrieved from biomedical and life science ontologies. However, frame-based representations of video scenes are quite limited, because they do not exploit potential information from the audio channel (if available) and happenings over time.

Motion semantics of video scenes include moving objects, their speed (and whether it is constant, accelerating, or decelerating), direction, and motion trajectory of objects in a scene, the interaction of objects when a moving object hits another one (whether the moving object goes through or collides with the stationary one, or stops). These semantics, together with rules associated with the depicted scene, can form video events.

Background knowledge can be utilized from biomedical ontologies, including concept hierarchy, relationships between concepts, and rules that define the specific knowledge domain of the video. The hierarchy of depicted concepts defined with logical formalisms enables specialization or generalization via subclass-superclass relationships, such as skin cancer is a subclass of the cancer class. Then, if skin cancer is detected in a video, not only can it be stated that the video scene features skin cancer, but also that cancer is depicted, which was not stated explicitly. Objects depicted in videos usually do not appear in isolation as they are often correlated to each other. A corpora of concepts frequently depicted together can be used to set likelihood values for the correctness of concept mapping, hence the co-occurrence of objects provides an additional layer of semantics to videos [29]. For example, the human heart is likely to be depicted with the lungs, but more than unlikely with the toes. Links to related resources enables knowledge discovery, advanced information retrieval, displaying useful information in hypervideo applications during video playback, and providing relevant videos that are, based on practitioners' preferences, potentially interesting for a particular medical case. Rules are suitable for annotating complex video events and provide very rich semantics about the specific knowledge domain related to the content of medical videos. In medical video scene interpretation, explicit expert knowledge provided by biomedical ontologies can be complemented by a priori knowledge obtained via rule-based reasoning [30].

2.3 Summary

This chapter listed the main challenges of automated medical video interpretation, and the potential of formalized expert knowledge in complementing automatically extracted low-level features. The importance of rich semantics has been highlighted, and the limitations of metadata and low-level features demonstrated.

Chapter 3
Formal Knowledge Representation

> “The Semantic Web is not a separate Web
> but an extension of the current one, in which
> information is given well-defined meaning,
> better enabling computers and people
> to work in cooperation.”
>
> — *Tim Berners-Lee*

Which automated tasks software agents can perform on video scene descriptions depends on the formalism used and the level of abstraction. Some data formats allow datatype association, others provide rich semantics. The choice of the representation language depends on the semantics you intend to capture, and affects implementations at the level of both knowledge organization and formal description.

3.1 Structured Data

Textual descriptions of multimedia resources constitute *unstructured data*, which is human-readable only [31]. For example, if a sentence in a natural language makes a statement about the duration of a medical video clip as plain text, software agents can only process the string as meaningless consecutive characters. If the same information is written as *semistructured data*, such as in XML, it becomes machine-readable, so that computers can extract different entities and properties from the text (e.g., the duration can be declared and retrieved as a positive integer). However, the meaning of the number is still not defined. By leveraging organized, *structured data*, the same statement can be made machine-interpretable.

Structured knowledge representations are usually expressed in, or based on, the *Resource Description Framework (RDF)*,[1] which can describe machine-readable statements in the form of subject-predicate-object (resource-property-value) triples, called *RDF triples*, e.g., scene-depicts-hand (see Definition 3.1).

Definition 3.1 (RDF Triple). Assume there are pairwise disjoint infinite sets of

1) International Resource Identifiers ($\mathbb{I}$), i.e., sets of strings of Unicode characters of the form `scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]` used to identify a resource;
2) RDF literals ($\mathbb{L}$), which are either a) self-denoting plain literals $\mathbb{L}_P$ of the form `"<string>"(@<lang>)?`, where `<string>` is a string and `<lang>` is an optional language tag, or b) typed literals $\mathbb{L}_T$ of the form `"<string>"^^<datatype>`, where `<datatype>` is an IRI denoting a datatype according to a schema, such as the XML Schema, and `<string>` is an element of the lexical space corresponding to the datatype;
3) Blank nodes ($\mathbb{B}$), i.e., unique but anonymous resources that are neither IRIs nor RDF literals.

A triple $(s, o, p) \in (\mathbb{I} \cup \mathbb{B}) \times \mathbb{I} \times (\mathbb{I} \cup \mathbb{L} \cup \mathbb{B})$ is called an *RDF triple* (or RDF statement), where s is the subject, p is the predicate, and o is the object.

3.2 Controlled Vocabularies and Ontologies: from RDFS to OWL 2

The classes, properties, and relationships that correspond to the subjects, predicates, and objects of RDF triples are typically defined in *controlled vocabularies* (see Definition 3.2) written in *RDF Schema (RDFS)*,[2] an extension of RDF specially designed for defining atomic concepts and taxo-

---

[1] https://www.w3.org/TR/rdf11-concepts/

[2] https://www.w3.org/TR/rdf-schema/

nomical structures, or *ontologies*[3] implemented in the first or second version of the fully featured ontology language, the *Web Ontology Language* (OWL).[4]

Definition 3.2 (Controlled Vocabulary). A controlled vocabulary is a triple $V = (\mathrm{N}_C, \mathrm{N}_R, \mathrm{N}_I)$ of countably infinite sets of IRI symbols denoting atomic concepts (concept names or classes) (N_C), atomic roles (role names, properties, or predicates) (N_R), and individual names (objects) (N_I), respectively, where N_C, N_R, and N_I are pairwise disjoint sets.

3.2.1 Modeling with RDFS

While RDF is the cornerstone of the Semantic Web, it was designed to describe machine-interpretable statements, not to formally define terms of a knowledge domain. For this reason, the RDF vocabulary was extended by concepts required for creating controlled vocabularies and basic ontologies, resulting in the *RDF Schema Language*, which was later renamed as the *RDF Vocabulary Description Language* (*RDFS*). RDFS is suitable for defining terms of a knowledge domain and basic relationships between them [32]. RDFS-based vocabularies and ontologies can be represented as RDF graphs, in which the subjects and objects of triples are the nodes and the connections between them are the predicates.

The RDFS classes and properties form the *RDFS vocabulary* (*rdfsV*), which is a superset of the vocabulary of RDF. The namespace URI of RDFS is `http://www.w3.org/2000/01/rdfschema#`, which is abbreviated with the `rdfs:` prefix. The RDFS vocabulary defines class resources (`rdfs:Resource`), the class of literal values such as strings and integers (`rdfs:Literal`), the class of classes (`rdfs:Class`), the class of RDF datatypes (`rdfs:Datatype`), the class of RDF containers (`rdfs:Container`), and the class of container membership properties (`rdfs:ContainerMembershipProperty`). The properties of RDFS can express that the subject is a subclass of a class (`rdfs:subClassOf`), the subject is a subproperty of a property (`rdfs:subPropertyOf`), add a human-readable name for the subject (`rdfs:label`), declare a description of the subject resource

---

[3] The formal definition of ontology depends on the logical underpinning, some of the most common of which will be defined later.

[4] https://www.w3.org/TR/owl-overview/

(`rdfs:comment`), identify a member of the subject resource (`rdfs:member`), add information related to the subject resource (`rdfs:seeAlso`), and provide the definition of the subject resource (`rdfs:isDefinedBy`).

Properties can be declared to apply to only certain instances of classes by defining their domain and range that indicate the relationships between RDFS classes and properties and RDF data. The `rdfs:domain` predicate indicates that a particular property applies to instances of a designated class (the domain of the property), in other words declares the class of those resources that may appear as subjects in a triple with the predicate. The `rdfs:range` predicate indicates that the values of a particular property are instances of a designated class or its permissible values are of a certain datatype, i.e., the class or datatype of those resources that may appear as the object in a triple with the predicate, also known as the range of the property, as shown in Listing 3.1.

Listing 3.1 Using RDFS domain and range

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix health: <http://health-lifesci.schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
health:Patient rdf:type rdfs:Class .
health:healthCondition rdf:type rdfs:Property .
health:healthCondition rdfs:domain health:Patient .
health:healthCondition rdfs:range xsd:string .
```

Note. The `rdf:type` predicate is used to express “is a” relationship between a subject and an object, and is one of the most common predicates for making general RDF statements. The shorthand notation for `rdf:type` is simply `a`.

3.2.2 Modeling with OWL

While simple machine-readable ontologies can be created using RDFS, complex knowledge domains demand more capabilities, such as the following:

- Relationships between classes (union, intersection, disjointness, equivalence)
- Property cardinality constraints (minimum, maximum, exact number, e.g., a person has exactly one father)
- Rich typing of properties
- Characteristics of properties and special properties (transitive, symmetric, functional, inverse functional, e.g., `:A :hasAncestor :B` and `:B :hasAncestor :C` implies that `:A :hasAncestor :C`)
- Specifying that a given property is a unique key for instances of a particular class
- Domain and range restrictions for properties when they are used with a certain class
- Equality of classes, specifying that two classes with different URI references actually represent the same class
- Equality of individuals, specifying that two instances with different URI references actually represent the same individual
- Enumerated classes

The *Web Ontology Language* (intentionally abbreviated as *OWL* rather than WOL [33]) is a knowledge representation language that semantically extends RDF, RDFS, and its predecessor language, DAML+OIL. OWL was specially designed for creating web ontologies with a rich set of modeling constructors, addressing the ontology engineering limitations of RDFS. The development of the first version of OWL was started in 2002, and the second version, *OWL 2*, in 2008. OWL became a W3C Recommendation in 2004 [34], and OWL 2 was standardized in 2009 [35]. OWL 2 is a superset of OWL, so all OWL ontologies can be handled by both OWL and OWL 2 applications. The significant extensions of OWL 2 over OWL include: new constructors that increase expressivity; extended datatype support; extended annotation capabilities; *syntactic sugar* (shorthand notation for common statements) and three profiles (fragments or sublanguages). Each OWL 2 profile provides a different balance between expressive power and reasoning complexity, thereby providing more options for different implementation scenarios [36]:

- OWL 2 EL: designed for handling ontologies with very large numbers of properties and/or classes
- OWL 2 QL: aimed at applications with a very large instance data volume and a priority for query answering

- OWL 2 RL: designed for applications that require scalable reasoning with relatively high expressivity

The default namespace of OWL is `http://www.w3.org/2002/07/owl#`, which defines the OWL vocabulary. There is no MIME type dedicated to OWL, but the `application/rdf+xml` or the `application/xml` MIME type is recommended for XML-serialized OWL documents with the `.rdf` or `.owl` file extension. Turtle files use the `text/turtle` MIME type and the `.ttl` file extension.

OWL property declarations use `rdf:type` as the predicate and the property type as the object. In OWL, there are four different property types:

- Ontology properties (`owl:OntologyProperty`) include the ontology import property (`owl:imports`) and ontology versioning properties (`owl:priorVersion`, `owl:backwardCompatibleWith`, and `owl:incompatibleWith`)
- Annotation properties declare annotations, such as labels, comments, and related resources (`owl:AnnotationProperty`). Five annotation properties are predefined in OWL, namely `owl:versionInfo`, `rdfs:label`, `rdfs:comment`, `rdfs:seeAlso`, and `rdfs:isDefinedBy`.
- Object properties link individuals to other individuals (`owl:ObjectProperty`). The object property highest in the hierarchy is `owl:topObjectProperty`.
- Datatype properties link individuals to data values (`owl:DatatypeProperty`).

The *ontology header*, which provides general information about the ontology, employs ontology properties and annotation properties. OWL ontologies are identified globally in ontology files by declaring the URI as the subject, `rdf:type` as the predicate, and `owl:Ontology` as the object (see Listing 3.2).

Listing 3.2 Ontology declaration

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<http://example.com/cancerontology.owl> a owl:Ontology .
```

Using this resource identifier as the subject, general information defined in other ontologies can be reused in the ontology by using `owl:imports` as the predicate, and the resource identifier of the other ontology as the object (see Listing 3.3).

Listing 3.3 Importing an ontology

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<http://example.com/cancerontology.owl> owl:imports
<http://example.org/diseaseontology.owl> .
```

Further information about an ontology can be defined in the ontology header using the `owl:versionInfo`, `rdfs:label`, `rdfs:comment`, `rdfs:seeAlso`, and `rdfs:isDefinedBy` annotation properties.

The following sections demonstrate the most common class, property, and individual declaration types available in OWL 2.

3.2.2.1 Class Declarations

Classes are declared in OWL by using the `rdf:type` predicate and setting the object to `owl:Class`. Class hierarchy is declared in the form of subclass-superclass relationships using the `rdfs:subClassOf` predicate. Atomic classes can be combined into complex classes using `owl:equivalentClass`, which contains all instances of both classes (`owl:intersectionOf`), every individual that is contained in at least one of the classes (`owl:unionOf`), and instances that are not instances of a class (`owl:complementOf`) (see Listing 3.4).

Listing 3.4 Atomic class, subclass, and complex class declaration in OWL

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
dbpedia:Brain a owl:Class .
dbpedia:Frontal_lobe a owl:Class ; rdfs:subClassOf dbpedia:Brain .
dbpedia:Lobes_of_the_brain a owl:Class ; owl:equivalentClass [ a owl:Class ;
owl:unionOf ( dbpedia:Frontal_lobe dbpedia:Parietal_lobe dbpedia:Occipital_lobe
dbpedia:Temporal_lobe dbpedia:Limbic_lobe dbpedia:Insular_cortex ) ] .
```

Note. A series of RDF triples sharing the same subject can be abbreviated by stating the subject once, and omitting the subject from the rest of the triples by listing the predicate-object pairs separated by semicolons.

Not every individual can be an instance of multiple classes: some class memberships exclude membership in other classes, i.e., instances cannot belong to two disjoint classes, for example, men and women (see Listing 3.5).

Listing 3.5 Declaration of disjoint classes

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
[] a owl:AllDisjointClasses ; owl:members ( dbpedia:Woman dbpedia:Man ) .
```

3.2.2.2 Property Declarations

Ontology axioms employ object properties to make statements about the relationship between properties. Datatype properties define the datatype of a property, enumerate the permissible values of a property, or define an interval of permissible values for a property. Object and datatype properties are also used for defining a special subclass of a class based on particular property values and value ranges. Annotation properties are used to declare labels, comments, and ontology versioning information. In OWL DL, annotation properties are not allowed in axioms, only in the ontology header.[5]

Property hierarchy is declared in the form of subproperty-superproperty relationships using `rdfs:subPropertyOf`. For example, in a medical ontology, the `operatedBy` property can be defined as a subproperty of the `treatedBy` object property (see Listing 3.6).

---

[5] In OWL Full, there is no such restriction on the use of annotation properties.

Listing 3.6 Subproperty declarations

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex: <http://example.com/> .
ex:operatedBy a owl:ObjectProperty ; rdfs:subPropertyOf ex:treatedBy.
```

The target individuals of the `isTreatmentFor` object property are drugs, and the property is applied to disorders (see Listing 3.7).

Listing 3.7 Property restrictions

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ex: <http://example.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ex:isTreatmentFor a owl:ObjectProperty ; rdfs:domain ex:Drug ; rdfs:range ex:Disorder .
```

Permissible values for datatype properties can be defined not only by external datatypes, such as the XSD datatypes used in the previous examples, but also by custom datatypes defined specially for the domain of an ontology by constraining or combining existing datatypes.

Restrictions on existing datatypes define an interval of values, which can be left-open (no minimum element defined), right-open (no maximum element defined), open (neither minimum nor maximum element defined), left-closed (minimum element defined), right-closed (maximum element defined), or closed (both minimum and maximum element defined). The restrictions are declared using `owl:withRestrictions`, and the restriction type is typically declared using a combination of `xsd:minInclusive`, `xsd:minExclusive`, `xsd:maxInclusive`, and `xsd:maxExclusive` (see Listing 3.8).

Listing 3.8 Defining a new datatype by restricting permissible values of a de facto standard datatype

```
@prefix ex: <http://example.com/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

Listing 3.8 (Cont'd) Defining a new datatype by restricting permissible values of a de facto standard datatype

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ex:zeroone owl:equivalentClass
  [ a rdfs:Datatype ;
    owl:onDatatype xsd:float ;
    owl:withRestrictions (
       [ xsd:minInclusive "0"^^xsd:float ]
       [ xsd:maxInclusive "1"^^xsd:float ]
    )
  ] .
```

Some properties can have only one unique value for each individual. Such properties are called *functional properties*. Both object properties and datatype properties can be functional properties. For example, to express that only one value can be associated with the severity of medical conditions, the `hasSeverity` property in a medical ontology should be declared as a functional property (see Listing 3.9).

Listing 3.9 Functional property example

```
@prefix medical: <http://example.com/medical/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
medical:hasSeverity a owl:FunctionalProperty .
```

Property equivalence can be described using `owl:equivalentProperty`, which can be used for example in an ophthalmology ontology to express that optical power and lens power are the same (see Listing 3.10).

Listing 3.10 Property equivalence example

```
@prefix ophth: <http://example.com/ophthalmology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
ophth:opticalPower a owl:DatatypeProperty ; owl:equivalentProperty ophth:lensPower .
```

Existential quantification can capture incomplete knowledge, e.g., pneumonia is located in the lung (see Listing 3.11).

Listing 3.11 Existential quantification example

```
@prefix ex: <http://example.com/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
ex:Pneumonia owl:equivalentClass [ a owl:Restriction ; owl:onProperty ex:locatedIn ;
owl:someValuesFrom ex:Lung ] .
```

Universal quantification is suitable for describing a class of individuals for which all related individuals must be instances of a particular class, e.g., bacterial pneumonia is caused by a bacterium (see Listing 3.12).

Listing 3.12 Universal quantification example

```
@prefix ex: <http://example.com/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
ex:BacterialPneumonia a owl:Class ; owl:equivalentClass [ a owl:Restriction ;
owl:hasCause ex:hasMember ; owl:allValuesFrom ex:Bacterium ] .
```

In contrast to existential quantification and universal quantification, *property cardinality restrictions* can not only specify a restriction, but also a minimum, maximum, or exact number for the restriction. A cardinality constraint is called an *unqualified cardinality restriction* if it constrains the number of values of a particular property irrespective of the value type. In contrast, *qualified cardinality restrictions* constrain the number of values of a particular type for a property. As an example, Listing 3.13 expresses that interpreters speak at least two languages.

Listing 3.13 A qualified cardinality restriction with a left-closed interval

```
@prefix ex: <http://example.com/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
ex:Interpreter rdfs:subClassOf [ a owl:Restriction; owl:onProperty ex:speaks ;
owl:minQualifiedCardinality 2 ; owl:onClass ex:Language ] .
```

Listing 3.14 shows an example of a cardinality constraint for which the exact number is known, namely, that four-chambered hearts have two ventricles.

Listing 3.14 A qualified cardinality restriction with an exact number

```
@prefix ex: <http://example.com/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ex:FourChamberedHeart a [ a owl:Restriction ;
owl:qualifiedCardinality "2"^^xsd:nonNegativeInteger ;
owl:onProperty ex:hasVentricles ;
owl:onClass ex:Heart ] .
```

Inverse properties can be declared using `owl:inverseOf` as the predicate, as demonstrated in Listing 3.15.

Listing 3.15 Inverse property declaration

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

Listing 3.15 (Cont'd) Inverse property declaration

```
@prefix ex: <http://example.com/> .
ex:hasTreatment a owl:ObjectProperty ; owl:inverseOf ex:isTreatmentFor .
```

In cases where the direction of the property does not matter, i.e., the property and its inverse coincide, the property is called *symmetric* and can be declared using `owl:SymmetricProperty`, such as `:relativeOf a owl:SymmetricProperty .` If the direction of the property matters, the property is called *asymmetric* and can be declared using `owl:AsymmetricProperty`, as for example `:hasChild a owl:AsymmetricProperty .`

Two properties are called *disjoint* if there are no two individuals that are interlinked by both properties. For example, no one can be the parent and the spouse of the same person, i.e., `:hasParent owl:propertyDisjointWith :hasSpouse .`

The properties that relate everything to themselves are called *reflexive properties* and can be declared using `owl:ReflexiveProperty`, e.g., `foaf:knows a owl:ReflexiveProperty .` The properties by which no individual can be related to itself are called *irreflexive properties*, which can express for example that nobody can be his own parent, i.e., `:parentOf a owl:IrreflexiveProperty .`

Transitivity can be expressed using `owl:TransitiveProperty .` For example, declaring `basedOn` as a *transitive property* (`:basedOn a owl:TransitiveProperty .`) can be used to express that an advanced version of a surgical procedure is also based on the experiments the original version of that procedure was based on.

If a simple object property expression, which has no transitive subproperties, is not expressive enough to capture a property hierarchy, *property chains* can be used. For example, a property chain axiom can express that the child of a child is a grandchild (see Listing 3.16).

Listing 3.16 A property chain axiom

```
@prefix family: <http://example.com/family/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
family:grandchildOf owl:propertyChainAxiom ( family:childOf family:childOf ) .
```

The individuals that are connected by a property to themselves can be described via *self-restriction* using `owl:hasSelf` (see Listing 3.17).

Listing 3.17 Self-restriction

```
@prefix ex: <http://example.com/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ex:Narcissist owl:equivalentClass [ a owl:Restriction ; owl:onProperty ex:likes ;
owl:hasSelf "true"^^xsd:boolean ] .
```

3.2.2.3 Individual Declarations

OWL individuals can be declared using `rdf:type` as the predicate and `owl:NamedIndividual` as the object in an RDF triple. For example, Perthes disease can be added to a disease ontology as shown in Listing 3.18.

Listing 3.18 Individual declaration

```
@prefix disease: <http://example.com/disease/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
disease:PerthesDisease a owl:NamedIndividual .
```

To express that an individual is an instance of a class, a statement is made by simply adding the individual as the subject, `rdf:type` as the predicate, and the class as the object (see Listing 3.19).

Listing 3.19 Declaring an individual as an instance of a class

```
@prefix disease: <http://example.com/disease/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://schema.org/> .
disease:PerthesDisease a disease:Disease .
```

It is always the most specific class that should be declared, because the individual will automatically become an individual of all the superclasses of the class too.

The relationship between individuals can be expressed by simply using the property as the predicate in a statement, for example using the `relatedTo` property to describe the relationship between two eye disorders (see Listing 3.20).

Listing 3.20 Connecting individuals

```
@prefix ex: <http://example.com/> .
@prefix eye: <http://example.com/eye/> .
eye:retinalDetachment ex:relatedTo eye:Floater .
```

The *equality* and *inequality* of individuals can be expressed using `owl:sameAs` and `owl:differentFrom` (see Listing 3.21).

Listing 3.21 Declaring identical and different individuals

```
@prefix eye: <http://example.com/eyediseases/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
eye:Floater owl:sameAs eye:VitreousOpacity .
eye:Floater owl:differentFrom eye:VisualSnow .
```

The individuals that belong to the same class can be enumerated using `owl:oneOf` (see Listing 3.22).

Listing 3.22 Enumerating instances of a class

```
@prefix eye: <http://example.com/eyedisorders/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
eye:EyeDisorder a owl:Class ; owl:equivalentClass [ a owl:Class ; owl:oneOf ( eye:AMD
eye:Blepharitis eye:Cataract eye:Conjunctivitis eye:CornealAbrasion
eye:DiabeticRetinopathy eye:DryEye eye:Floater eye:Glaucoma eye:Pterygium
eye:RetinalDetachment eye:Uveitis ) ] .
```

3.2.3 Serialization

While all the previous examples were written in Turtle, OWL supports other serializations as well. OWL ontologies correspond to RDF graphs, i.e., sets of RDF triples. As with RDF graphs, OWL ontology graphs can be expressed in various syntactic notations. For the first version of OWL two special syntaxes were developed: the *XML presentation syntax*[6] and the *abstract syntax*.[7] Beyond the *RDF/XML syntax*,[8] the normative exchange syntax of both OWL and OWL 2, and the *Turtle syntax* used in the previous examples, OWL 2 also supports *OWL/XML*,[9] the *functional syntax*,[10] and the *Manchester syntax*.[11] To compare these syntaxes, a code snippet of an ontology is serialized in each syntax, defining an ontology file, prefixes for the namespaces, and the intersection of two classes. Listing 3.23 shows the RDF/XML serialization.

Listing 3.23 RDF/XML syntax example

```
<?xml version="1.0" encoding="UTF-8"?>
 <rdf:RDF xml:base="http://example.org/"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 <owl:Ontology rdf:about="tumor.owl" />
  <owl:Class rdf:about="Tumor">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="BenignTumor"/>
          <owl:Class rdf:about="PreMalignantTumor"/>
          <owl:Class rdf:about="MalignantTumor"/>
        </owl:intersectionOf>
```

---

[6] https://www.w3.org/TR/owl-xmlsyntax/

[7] https://www.w3.org/TR/2003/WD-owl-semantics-20030331/syntax.html

[8] https://www.w3.org/TR/owl-mapping-to-rdf/

[9] https://www.w3.org/TR/owl-xml-serialization/

[10] https://www.w3.org/TR/owl-syntax/

[11] https://www.w3.org/TR/owl2-manchester-syntax/

Listing 3.23 (Cont'd) RDF/XML syntax example

```
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
 </rdf:RDF>
```

RDF/XML is the only syntax all OWL 2-compliant tools must support, however, it is rather verbose for big ontologies and somewhat difficult to read for humans. This is why the more compact and easier-to-read RDF syntax, Turtle, is commonly used for representing RDF triples for OWL 2 ontologies, as shown in Listing 3.24.

Listing 3.24 OWL/Turtle example

```
@prefix : <http://example.org/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
<http://example.org/tumor.owl> a owl:Ontology .
:Tumor owl:equivalentClass [ a owl:Class ; owl:intersectionOf ( :BenignTumor
:PreMalignantTumor :MalignantTumor ) ] .
```

OWL/XML is an alternate exchange syntax for OWL 2 (see Listing 3.25).

Listing 3.25 OWL/XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<Ontology
  xml:base="http://example.org/"
  ontologyIRI="http://example.org/tumor.owl"
  xmlns="http://www.w3.org/2002/07/owl#">
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
   <EquivalentClasses>
    <Class IRI="Tumor" />
```

Listing 3.25 (Cont'd) OWL/XML example

```
    <ObjectIntersectionOf>
      <Class IRI="BenignTumor" />
      <Class IRI="PreMalignantTumor" />
      <Class IRI="MalignantTumor" />
    </ObjectIntersectionOf>
  </EquivalentClasses>
 </Ontology>
```

OWL/XML was designed for XML tools that use XQuery, XPath, and XSLT, for which RDF/XML would be difficult to process (beyond parsing and rendering).

The OWL 2 functional syntax is clean, adjustable, modifiable, and easy to parse (see Listing 3.26).

Listing 3.26 OWL 2 functional syntax example

```
Prefix(:=<http://example.org/>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
 Ontology(<http://example.org/tumor.owl>
  EquivalentClasses(
    :Tumor
    ObjectIntersectionOf( :BenignTumor :PreMalignantTumor :MalignantTumor )
  )
)
```

The functional syntax makes the formal structure of ontologies clear, and it is almost exclusively used for defining the formal OWL 2 grammar in the high-level structural specifications of the W3C. The OWL 2 functional syntax is compatible with UML, one of the most widely deployed general-purpose standardized modeling languages.

The less frequently used Manchester syntax is a compact, user-friendly syntax for OWL 2 DL that collects information about a particular class, property, or individual into a single construct called a frame. The Manchester syntax is easy to read and write, especially for those who are not experts in mathematical logic. Complex descriptions consist of short, meaningful English words, while eliminating the logical symbols and precedence rules represented in other syntaxes, as shown in Listing 3.27.

Listing 3.27 Manchester syntax example

```
Prefix: : <http://example.org/>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Ontology: <http://example.org/tumor.owl>
Class: Tumor
EquivalentTo: BenignTumor and PreMalignantTumor and MalignantTumor
```

3.3 Description Logics: Formal Grounding for OWL Ontologies

OWL ontologies can be formally grounded in *description logics (DL)*.[12] Description logics are formal knowledge representation languages that support different sets of mathematical constructors[13] to achieve a favorable trade-off between expressivity and computational complexity for the intended application.[14] For example, the $\mathcal{ALC}$ description logic supports atomic negation, concept intersection, universal restrictions, limited existential quantification, and complex class expressions using a combination of operators, such as conjunction. $\mathcal{ALC}$ extended with transitive roles is called $\mathcal{S}$. If all the previous constructors are extended with $\mathcal{H}$ (role hierarchy), $\mathcal{O}$ (enumerated concept individuals), $\mathcal{I}$ (inverse roles), and $\mathcal{N}$ (unqualified cardinality restrictions), the description logic is called $\mathcal{SHOIN}$, which roughly corresponds to OWL DL. Adding $\mathcal{R}$ (complex role inclusion, reflexivity and irreflexivity, and role disjointness) to the above and replacing $\mathcal{N}$ with $\mathcal{Q}$ (qualified cardinality restrictions) yields to $\mathcal{SROIQ}$, which is the description logic behind OWL 2 DL (see Definition 3.3). Those description logics that support datatypes, datatype properties, and data values also include a trailing $^{(\mathcal{D})}$ superscript in their names.

---

[12] There are three flavors of OWL, each constituting different compromises between expressivity and computational complexity: OWL Full, OWL DL, and OWL Lite. The "DL" in OWL DL refers to the description logic underpinning.

[13] The letters in description logic names refer to the set of available mathematical constructors, except when using a letter representing a superset of another letter, in which case the letter of the subset is omitted.

[14] Sikos, L. F. (2017) Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5

Definition 3.3 ($\mathcal{SROIQ}$ Ontology).[15] A $\mathcal{SROIQ}$ *ontology* is a set of role expressions R over a signature defined as R ::= $U \mid N_R \mid N_{\bar{R}}$, where U represents the universal role, N_R is a set of roles, and $N_{\bar{R}}$ is a set of negated role assertions. The concept expressions of a $\mathcal{SROIQ}$ ontology are defined as the set C ::= $N_C \mid (C \sqcap D) \mid (C \sqcup D) \mid \neg C \mid \top \mid \bot \mid \exists R.C \mid \forall R.C \mid \geqslant nR.C \mid \leqslant nR.C \mid \exists R.\mathit{Self} \mid \{N_I\}$, where n is a nonnegative integer, C and D represent concepts and R represents roles. Based on these sets, $\mathcal{SROIQ}$ axioms can be defined as general concept inclusions (GCIs) of the form $C \sqsubseteq D$ and $C \equiv D$ for concepts C and D (terminological knowledge, TBox), individual assertions of the form $C(N_I)$, $R(N_I, N_I)$, $N_I \approx N_I$, or $N_I \not\approx N_I$ (assertional knowledge, ABox), and role assertions of the form $R \sqsubseteq S$, $R \equiv S$, $R_1 \circ \ldots \circ R_n \sqsubseteq S$, *Asymmetric*($R$), *Reflexive*($R$), *Irreflexive*($R$), *Disjoint*($R$, S)[16] for roles R, R_i, and S (role box, RBox) [37].

Interpretation $\mathcal{I}$ consists of a set $\Delta^{\mathcal{I}}$ (the domain of $\mathcal{I}$) and an interpretation function $\cdot^{\mathcal{I}}$, which maps each atomic concept A to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each atomic role R to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

3.4 A Hybrid DL-Based Formalism for Video Event Representation

Ontology-based representation of video scenes and events indicates a promising direction in content-based video retrieval. However, the multimedia ontologies described in the literature often lack formal grounding, and none of them are suitable for representing complex video scenes. This issue can be partially addressed using SWRL[17] rules, which, however, can lead to undecidability. For this reason, a hybrid description logic-based architecture has been proposed that employs general, spatial, temporal, and fuzzy axioms for video scene representation and automated reasoning-based scene interpretation, while achieving a favorable tradeoff between expressivity and reasoning complexity.[18]

---

[15] The formal definition of an ontology depends on its logical underpinning, but the most expressive OWL 2 ontologies defined here are supersets of all the ontologies that utilize less expressive formalisms. Most OWL 2 ontologies do not exploit all the available mathematical constructors of the underlying logical underpinning.

[16] Often abbreviated with the first three letters as *Asy*(R), *Ref*(R), *Irr*(R), and *Dis*(R, S).

[17] Semantic Web Rule Language

[18] Sikos, L. F. (2017) Spatiotemporal reasoning for complex video event recognition in content-based video retrieval. In: Hassanien, A., Shaalan, K., Gaber, T., Tolba, M. (eds.) Proceedings of the International Conference on Advanced In-

3.4.1 Rationale

While video event representation requires an expressivity higher than that of OWL 2 due to the complexity of video events that cannot be described using terminological knowledge alone [38, 39, 40], even the mathematical constructors behind OWL 2 are not exploited in most multimedia ontologies.[19] A popular approach to describe video events is to use rule-based mechanisms, such as SWRL rules [41]. In contrast to OWL 2 formalisms, however, SWRL rules may break decidability. An alternative to rule-based formalisms for improving expressivity is formal grounding with spatial, temporal, or fuzzy description logics,[20] though not all of these are decidable either.

The most well-known spatial description logics are spatial extensions of the core description logics $\mathcal{ALC}$, the first two of which are $\mathcal{ALCRP}^{(\mathcal{D})}$ [42] and $\mathcal{ALCRP}^{3(\mathcal{D})}$ [43]. Other $\mathcal{ALC}$-based spatial description logics implement the *Region Connection Calculus (RCC)* with various levels of granularity, such as $\mathcal{ALC}\ (\mathcal{D}_{\mathcal{RCC}8})$ [44], $\mathcal{ALCI}_{\mathcal{RCC}1}$, $\mathcal{ALCI}_{\mathcal{RCC}2}$, $\mathcal{ALCI}_{\mathcal{RCC}3}$, $\mathcal{ALCI}_{\mathcal{RCC}5}$, and $\mathcal{ALCI}_{\mathcal{RCC}8}$ [45]. RCC describes the spatial relationship between two-dimensional objects in a plane as regions not only by their metrical and/or geometric attributes, but also by the qualitative spatial relationships between them [46].

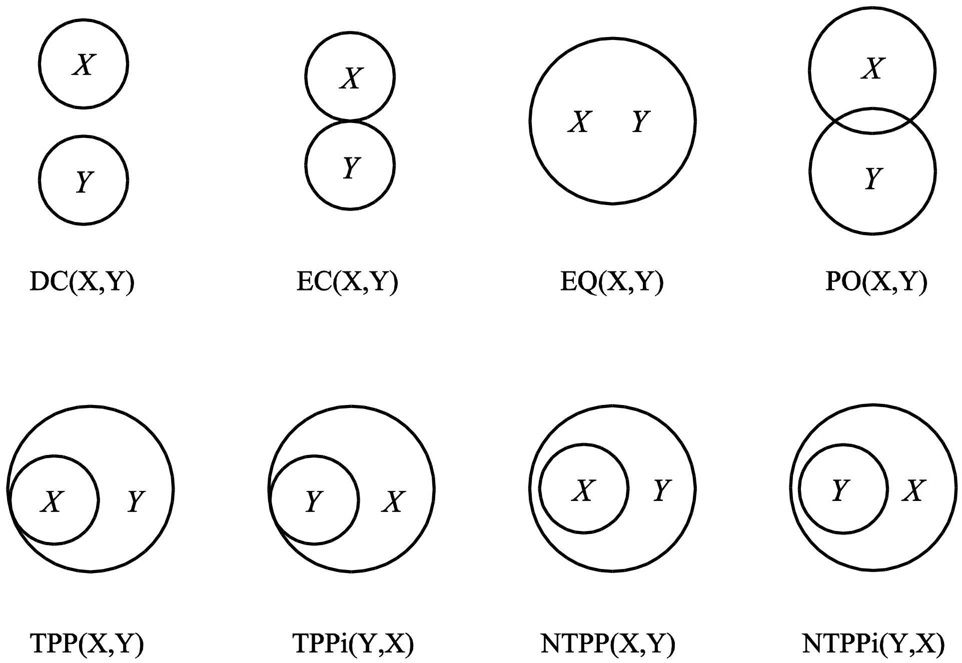
RCC8 consists of eight basic relations that are possible between two regions: disconnected (DC), externally connected (EC), equal (EQ), partially overlapping (PO), tangential proper part (TPP), tangential proper part inverse (TPPi), non-tangential proper part (NTPP), and non-tangential proper part inverse (NTPPi) (see Fig. 3.1).

---

telligent Systems and Informatics 2017. Advances in intelligent systems and computing, vol. 639, pp. 704–713. 3rd International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 9–11 September 2017. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-64861-3_66

[19] Sikos, L. F. (2016) A novel approach to multimedia ontology engineering for automated reasoning over audiovisual LOD datasets. In: Nguyễn, N. T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) Intelligent information and database systems, pp. 3–12. 8th Asian Conference on Intelligent Information and Database Systems, Đà Nẵng, Vietnam, 14–16 March 2016. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-49381-6_1

[20] Sikos, L. F. (2017) Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5


DC(X,Y) EC(X,Y) EQ(X,Y) PO(X,Y)

TPP(X,Y) TPPi(Y,X) NTPP(X,Y) NTPPi(Y,X)

Figure 3.1 Two-dimensional examples for the eight core RCC8 relations

Binary topological relations can also be described with Egenhofer relations [47].

$\mathcal{ALC}(\mathcal{CDC})$ is the extension of $\mathcal{ALC}$ with the *Cardinal Direction Calculus (CDC)* [48], which can also be combined with $\mathcal{ALC}(\mathcal{RCC})$, yielding to $\mathcal{ALC}(\mathcal{CDRCC})$. Some of these languages are suitable for annotating regions of interest (ROI) in videos, however, not all of them are decidable. For example, $\mathcal{ALCRP}^{(\mathcal{D})}$ and $\mathcal{ALCRP}^{3(\mathcal{D})}$ are decidable only if syntax restrictions are applied.

Temporal description logics are suitable for the formal representation of video actions and video event recognition via reasoning [49], and feature datatypes for time points, time intervals, or sets of time intervals [50]. Allen's interval algebra, which defines possible relations between time intervals and provides a composition table that can serve as a basis for reasoning about temporal descriptions of events [51], is commonly used in temporal description logics. It defines 13 base relations to capture the possible relations between two intervals, *i* and *j*:

- *i* takes place before *j*
- *i* meets *j*
- *i* overlaps with *j*
- *i* starts *j*
- *i* during *j*
- *i* finishes *j*

- *i* is equal to *j*
- the inverse of the first 6 relations (see Fig. 3.2)

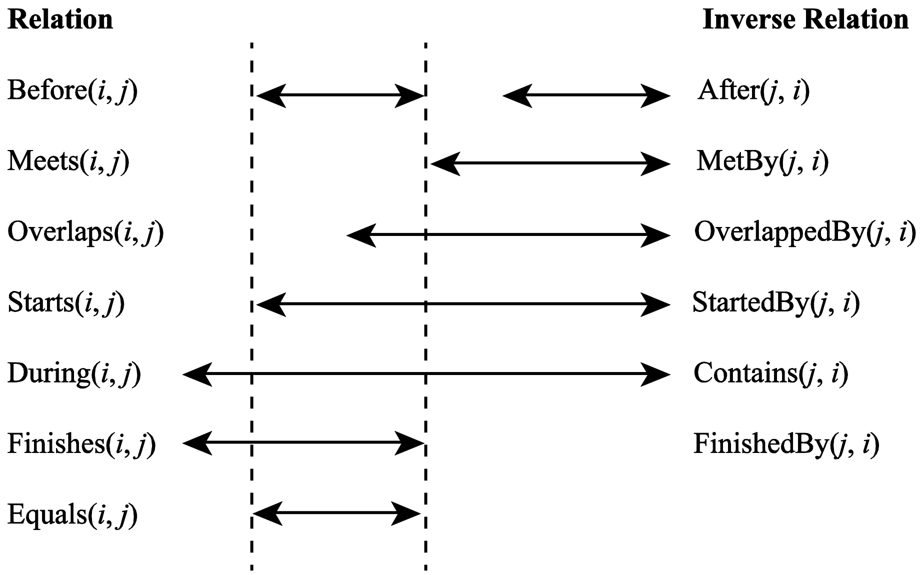


Figure 3.2 Allen's temporal operators

This calculus enables the formal description of temporal facts to be used for automated reasoning. Relations between intervals are formalized as sets of base relations. The composition of relations between intervals turns Allen's interval algebra into a relation algebra.

Note. Allen's interval algebra can be used not only for temporal intervals, but also for spatial configurations, for which the relations are interpreted as relative positions of spatial objects.

Standard temporal logic operators are usually combined with general description logics by adding temporal operators as additional concept constructors and/or applying the temporal operators to the TBox, the ABox, and roles. Since adding temporal operators to description logics can yield to undecidability, such as recursively non-enumerable formalisms, most research focus is on combining the propositional fragment of temporal logics with description logics. The basic temporal description logic $\mathcal{TL}$-$\mathcal{F}$ is composed of the temporal language $\mathcal{TL}$, which expresses interval temporal networks, and a non-temporal feature description logic $\mathcal{F}$. More expressive temporal description logics include $\mathcal{TLU}$-$\mathcal{FU}$, which adds disjunction to both the temporal and the non-temporal sides of $\mathcal{TL}$-$\mathcal{F}$, and $\mathcal{TL}$-$\mathcal{ALCF}$, which extends the non-temporal side of $\mathcal{TL}$-$\mathcal{F}$ with roles and full propositional calculus [52]. $\mathcal{T}$-$\mathcal{ALC}$ is the temporal extension of the core description logic $\mathcal{ALC}$ with time intervals, which are suitable for, among other things, reasoning over actions of video events [53]. $\mathcal{TL}$-$\mathcal{ALC}$ employs the temporal operators ○ (next) and $\mathcal{U}$ (until) applied to concepts and formulae [54]. $\mathcal{TL}$-$\mathcal{ALCF}$ is composed of the interval-based temporal logic $\mathcal{TL}$ and the non-temporal description logic $\mathcal{ALCF}$, and was designed for knowledge domains that feature objects with properties that vary over time [55]. $\mathcal{ALC}$-$\mathcal{LTL}$ employs $\mathcal{ALC}$ axioms to replace propositional variables for describing properties of system states [56]. The combination of the description logic $\mathcal{ALC}$ and Prior's *Tense Logic* resulted in the $\mathcal{XALC}$ and $\mathcal{BALC}_I$ temporal description logics, the first of which features the next time operator, and the second one a restricted version of the next time, any time, and some time operators [57]. DLR_{US} is the extension of the description logic *DLR* with the temporal operators $\mathcal{U}$ (until) and $\mathcal{S}$ (since) [58]. $\mathcal{TL}$-$\mathcal{SI}$ consists of the temporal logic $\mathcal{TL}$ and the description logic $\mathcal{SI}$ [59]. There are 16 temporal description logics based on *DL-Lite*, featuring different levels of expressivity via allowing or restricting various types of temporal concepts, including $\mathcal{U}$ (until), $\mathcal{S}$ (since), ○F (next time), and ○P (previous time) operators in quantitative evolution constraints, and the □F (always in the future) and □P (always in the past) operators in qualitative evolution constraints [60]. The temporal extension of $\mathcal{SHIN}^{(\mathcal{D})}$ which, when implemented in OWL, is called tOWL, provides time points, relations between time points, intervals, and timeslices [61]. $\mathcal{DL}$-$\mathcal{CTL}$ extends description logics with propositional branch-time logic (CTL), and was designed to specify temporal properties on state transition systems [62]. The $\mathcal{EL}$ family of description logics has been combined with *Computational Tree Logic (CTL)* fragments, including ◇*E* (possibly eventually) and ○*E* (possibly at the next state), resulting in the temporal description logics $CTL^{E\diamond}_{\mathcal{EL}}$, $CTL^{E\circ}_{\mathcal{EL}}$, and $CTL^{E\circ,E\diamond}_{\mathcal{EL}}$ [63]. $HS\text{-}Lite^{H}_{horn}$ is the combination of the *Halpern-Shoham temporal interval logic (HS logic)* with the $DL\text{-}LiteH_{horn}$ description logic [64]. More recently, a metric interval-based temporal description logic has also been introduced [65].

As with spatial description logics, in temporal description logics it is also a challenge to push the limits of expressivity without breaking decidability, because using just one rigid role and the concept constructors ⊓, ∃*R*.*C*, and ◇F yields to undecidability [66]. There are essentially three options to maintain decidability in temporal description logics [67]:

1) disallow rigid roles and temporal operators on roles,
2) weaken the temporal component to the “undirected” temporal operators $\boxast$ and $\diamondast$,
3) allow arbitrary temporal operators on $\mathcal{ALC}$ axioms only, but not on concepts or roles.

Despite the large number of spatial and temporal description logics, there are very few mentions of spatiotemporal description logics in the literature, one of which is $\mathcal{ALCRP}(S_2 \oplus T)$ [68].

Fuzzy description logics can be used for video content indexing by expressing the uncertainty of concept and event depiction [69]. Notable fuzzy description logics include f-$\mathcal{ALC}$, $\mathcal{ALCQ}_F^+$, f-$\mathcal{ALCIQ}$, f-$\mathcal{SHIN}$, f-$\mathcal{SHOIN}$, f-$\mathcal{SHOIQ}$, *C*-$\mathcal{SHOIN}$, and f-$\mathcal{SROIQ}$, which extend the corresponding general description logics with fuzzy features. For example, f-$\mathcal{SHOIN}$ is the fuzzy extension of the $\mathcal{SHOIN}$ description logic, and can be briefly defined as follows [70]. Consider an alphabet of concept names (C), abstract role names (R_A), concrete role names (R_D), abstract individuals (I_A), concrete individuals (I_D), and concrete datatypes (D). The set of $\mathcal{SHOIN}^{(\mathcal{D})}$ roles is defined by $R_A \cup \{R^- \mid R \in R_A\} \cup R_D$, where R^- is called the inverse role of R. Let $A \in C$, R, $S \in R_A$, where S is a simple role, $T_i \in R_D$, d is a datatype, $o, o_1, \ldots, o_k \in I_A$, c, $n \in (0,1]$, $p \in \mathbb{N}$, and $k \in \mathbb{N}^*$. Based on these sets, the f-$\mathcal{SHOIN}^{(\mathcal{D})}$ concepts are defined inductively by the production rule shown in Definition 3.4.

Definition 3.4 (f-$\mathcal{SHOIN}^{(\mathcal{D})}$ Concept). The f-$\mathcal{SHOIN}^{(\mathcal{D})}$ concepts are defined by the rule $C, D \rightarrow \top \mid \bot \mid A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \exists R.C \mid \forall R.C \mid \geqslant pS \mid \leqslant pS \mid \exists T.u \mid \forall T.u \mid \geqslant pT \mid \leqslant pT \mid \{o\} \mid \{o_1, \ldots, o_k\} \mid R(o)u \rightarrow d \mid \{c\}$

The semantics of f-$\mathcal{SHOIN}$ are provided by the fuzzy interpretation of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, ^{\mathcal{I}})$ and the interpretation of the concrete (datatype) domain $D = (\Delta_D, ^{\mathcal{D}})$ (see Definition 3.5).

Definition 3.5 (f-$\mathcal{SHOIN}^{(\mathcal{D})}$ Interpretation). An f-$\mathcal{SHOIN}$ interpretation is defined by a quadruple of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \Delta_{\mathcal{D}}, ^{\mathcal{I}}, ^{\mathcal{D}})$, where the abstract domain $\Delta^{\mathcal{I}}$ is a nonempty set of objects, the datatype domain $\Delta_{\mathcal{D}}$ is the domain of interpretation of all datatypes (disjoint from $\Delta^{\mathcal{I}}$) consisting of data values, and $^{\mathcal{I}}$ and $^{\mathcal{D}}$ are two fuzzy interpretation functions, which map an abstract individual a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, a concrete individual c to an element $c^{\mathcal{D}} \in \Delta_{\mathcal{D}}$, a concept name A to a function

$A^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0,1]$, an abstract role name R to a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0,1]$, a datatype d to a function $d^{\mathcal{D}} : \Delta_{\mathcal{D}} \rightarrow [0,1]$, and a concrete role name T to a function $T^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta_{\mathcal{D}} \rightarrow [0,1]$.

This makes it possible to assign any degree between 0 and 1 to any object or object pair of a fuzzy concept. Using this interpretation, the semantics of f-$\mathcal{SHOIN}$ concepts can be summarized as shown in Table 3.1.

Table 3.1 Syntax and semantics of f-$\mathcal{SHOIN}^{(\mathcal{D})}$ concepts

Constructor	Syntax	Semantics
Top concept	$\top$	$\top^{\mathcal{I}}(a) = 1$
Bottom concept	$\bot$	$\bot^{\mathcal{I}}(a) = 0$
Data value	c	$c^{\mathcal{I}} = c^{\mathcal{D}}$
Datatype	d	$d^{\mathcal{I}}(y) = d^{\mathcal{D}}(y)$
Conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}}(a) = t(C^{\mathcal{I}}(a),D^{\mathcal{I}}(a))$
Disjunction	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}}(a) = u(C^{\mathcal{I}}(a),D^{\mathcal{I}}(a))$
Negation	$\neg C$	$(\neg C)^{\mathcal{I}}(a) = c(C^{\mathcal{I}}(a))$
Nominal	$\{o\}$	$\{o\}^{\mathcal{I}}(a) = 1$ iff $o^{\mathcal{I}} = a$, otherwise $\{o\}^{\mathcal{I}}(a) = o$
One of	$\{o_1, \ldots, o_k\}$	$\{o_1, \ldots, o_k\}^{\mathcal{I}}(a) = 1$ if $a \in \{o_1^{\mathcal{I}}, \ldots, o_k^{\mathcal{I}}\}$, 0 otherwise
Fills	$R(o)$	$(R(o))^{\mathcal{I}}(a) = R^{\mathcal{I}}(a,o^{\mathcal{I}})$
Existential quantification	$\exists R.C$	$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b\in\Delta^{\mathcal{I}}} t(R^{\mathcal{I}}(a,b),C^{\mathcal{I}}(b))$
Universal quantification	$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b\in\Delta^{\mathcal{I}}} J(R^{\mathcal{I}}(a,b),C^{\mathcal{I}}(b))$
At-least restriction	$\geqslant pS$	$(\geqslant pS)^{\mathcal{I}}(a) = \sup_{b_1,\ldots,b_p\in\Delta^{\mathcal{I}}} t\left(\underset{i=1}{\overset{p}{t}} S^{\mathcal{I}}(a,b_i), \underset{i<j}{t}\{b_i \neq b_j\}\right)$
At-most restriction	$\leqslant pS$	$(\leqslant pS)^{\mathcal{I}}(a) = \inf_{b_1,\ldots,b_{p+1}\in\Delta^{\mathcal{I}}} J\left(\underset{i=1}{\overset{p+1}{t}} S^{\mathcal{I}}(a,b_i), \underset{i<j}{u}\{b_i = b_j\}\right)$

Table 3.1 (Cont'd) Syntax and semantics of f-$\mathcal{SHOIN}^{(\mathcal{D})}$ concepts

Constructor	Syntax	Semantics
Inverse role	R^{-}	$(R^{-})^{\mathcal{I}}(b,a) = R^{\mathcal{I}}(a,b)$
Datatype exists	$\exists T.d$	$(\exists T.d)^{\mathcal{I}}(a) = \sup_{y \in \Delta_{\mathcal{D}}} t(T^{\mathcal{I}}(a,y),d^{\mathcal{I}}(y))$
Datatype value	$\forall T.d$	$(\forall T.d)^{\mathcal{I}}(a) = \inf_{y \in \Delta_{\mathcal{D}}} \mathrm{J}(T^{\mathcal{I}}(a,y),d^{\mathcal{I}}(y))$
Datatype at-least	$\geqslant pT$	$(\geqslant pT)^{\mathcal{I}}(a) = \sup\limits_{y_1,\ldots,y_p \in \Delta_D} t\left(\underset{i=1}{\overset{p}{t}} R^{\mathcal{I}}(a,y_i), \underset{i<j}{t}\{y_i \neq y_j\}\right)$
Datatype at-least	$\leqslant pT$	$(\leqslant pT)^{\mathcal{I}}(a) = \inf\limits_{y_1,\ldots,y_{p+1} \in \Delta_D} \mathrm{J}\left(\underset{i=1}{\overset{p+1}{t}} R^{\mathcal{I}}(a,y_i), \underset{i<j}{u}\{y_i \neq y_j\}\right)$
Datatype nominal	$\{c\}$	$\{c\}^{\mathcal{I}}(y) = 1$ iff $c^{\mathcal{D}} = y$, $\{c\}D(y) = o$ otherwise

Some of the aforementioned description logics have been used over the decades for video annotation. The idea of formalizing video events as the combination of description logic concepts, grouping roles, and temporal logic formalism dates back to the late 1990s [71]. Since then, constrained videos have been described using description logic formalisms [72], and depicted sequences and objects defined in the form of DL axioms [73]. Some approaches combined the results of low-level feature extraction with a DL-based ontology to describe video events [74]. However, the limitations of general description logics in video descriptions became obvious, and various attempts have been made to push the expressivity boundaries of such formalisms. Some examples include dynamic description logics that purposefully combine a general description logic with a temporal logic for video event detection [75], very expressive description logics with probabilistic reasoning [76], fuzzy logic-based scene recognition [77], and formal knowledge models for the recognition of specific video events, such as human actions [78].

The following sections present a novel approach for spatiotemporal reasoning over video contents by introducing a formal knowledge representation mechanism that combines general, spatial, temporal, and fuzzy constructors, and was specially designed for video content representation.[21]

---

[21] Sikos, L. F. (2017) Spatiotemporal reasoning for complex video event recognition in content-based video retrieval. In: Hassanien, A., Shaalan, K., Gaber, T., Tolba, M. (eds.) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017. Advances in intelligent systems and computing, vol. 639, pp. 704–713. 3rd International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 9–11 September 2017. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-64861-3_66

3.4.2 Development

The approaches for description logic-based formalisms in the literature focus mainly on image representation and scene interpretation for images via reasoning over DL axioms, some implement temporal logics for audio description, and others spatial representations for geometric data and spatial segmentation for the regions of interest.[22] Fuzzy knowledge representations are used to express likelihood or the degree of uncertainty.[23] Most of these approaches have been developed independently, however, many of them yield to undecidability, and cannot be used in parallel due to incompatible constraints.

When designing a hybrid knowledge representation architecture for video events aimed at addressing the limitations of previous approaches, the following aims have been formulated. First of all, the formalism should be compatible with OWL 2 DL. In order to achieve this, most of the temporal and fuzzy knowledge representations described in the literature had to be excluded, as many of them cannot be implemented directly in OWL. As shown earlier, not all spatial-temporal formalisms are decidable, which, however, is another desirable feature for the knowledge representation of videos. Therefore, our approach combines carefully crafted general description logic grammar suitable for conceptual modeling [79] with RCC-compliant spatial constructors [80], interval-based temporal constructors [81], and fuzzy concepts and fuzzy roles [82], all of which meet the above criteria.

Considering that the sets of points on the plane delimited by a continuous boundary curve (i.e., the regions of interest) can correspond to concepts depicted in video scenes, our architecture defines the RCC relations $DC(C,D)$, $EQ(C,D)$, $EC(C,D)$, $PO(C,D)$, $TPP(C,D)$ and its inverse, and $NTPP(C,D)$ and its inverse as DL axioms of the form $\langle C \sqsubseteq \neg D\rangle$, $\langle C \equiv D\rangle$, $\langle \forall R.C \sqsubseteq \exists R.\neg D , C \sqcap D\rangle$, $\langle \forall R.C \sqcap \forall R.D, C \sqcap \neg D, \neg C \sqcap D\rangle$, $\langle C \sqsubseteq D , C \sqcap \exists R.\neg D\rangle$, and $\langle C \sqsubseteq \forall R.D\rangle$. The temporal part implements an interval-based temporal logic that represents temporal constraint networks based on

---

[22] Sikos, L. F. (2016) RDF-powered semantic video annotation tools with concept mapping to Linked Data for next-generation video indexing. Multimedia Tools and Applications, 76(12):14437–14460. doi: 10.1007/s11042-016-3705-7

[23] Sikos, L. F. (2018) Ontology-based structured video annotation for content-based video retrieval via spatiotemporal reasoning. *Invited chapter.* In: Kwaśnicka, H., Jain, L. C. (eds.) Bridging the semantic gap in image and video analysis. Intelligent Systems Reference Library, vol. 145, pp. 97–122. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-73891-8_6

Allen's temporal relations, and maps DL concept expressions to time intervals, thereby enabling reasoning over video events. Not all fuzzy knowledge representations found in the literature can be implemented in OWL; in fact, they often extend the standard to be able to handle vagueness and uncertainty. In contrast, the fuzzy part of our formalism is the realization of an OWL 2-compliant fuzzy knowledge representation framework that supports N-ary relations.

Assume three countably finite and pairwise disjoint sets of atomic concepts (N_C), atomic roles (N_R), and individual names (N_I), denoted by Internationalized Resource Identifiers (IRIs). The set of concept expressions in our formalism is defined as $C ::= N_C \mid \top \mid \bot \mid C \sqcap D \mid C \sqcap \neg D \mid \neg C \sqcap D \mid C \sqcup D \mid E \sqcap F \mid E@X \mid E[Y]@X \mid \diamond(\overline{X})\overline{T}_C.E \mid p \downarrow q \mid C(p) \mid \neg C \mid \forall R.C \mid \exists R.C \mid \forall T.\mathrm{d} \mid \exists T.\mathrm{d} \mid \{\alpha/a\} \mid \geqslant mS.C \mid \leqslant mS.C \mid \geqslant mT.\mathrm{d} \mid \leqslant mT.\mathrm{d} \mid \exists S.Self \mid mod(C) \mid \alpha \cdot C \mid (\alpha \cdot C) + \ldots + (\alpha_k \cdot C_k)$, where C and D represent (possibly complex) fuzzy concepts, R represents (possibly complex) abstract fuzzy roles, E and F represent temporal concepts, X and Y represent temporal variables, $E@X$ denotes temporal concept E at interval X, $\diamond(\overline{X})\overline{T}_C.E$ represents temporal existential quantifiers, p and q are features, $p \downarrow q$ means agreement, $C(p)$ represents selection, S represents simple fuzzy roles, T represents concrete fuzzy roles, d represents fuzzy concrete predicates, $\alpha \in [0,1]$, m is a positive integer, k is a rational number, and *mod* is a fuzzy modifier. Temporal constraints are defined in the form $T_C \rightarrow (X(R)Y) \mid (X(R)\ \#) \mid (\#\ (R)Y)$, where # is a distinguished temporal variable that serves as the reference interval.

The set of role expressions is defined as $R ::= N_R \mid R^{-} \mid T \mid U \mid mod(R)$, where N_R is an atomic fuzzy role, R is a (possibly complex) abstract fuzzy role, T is a concrete fuzzy role, U represents the universal role, and *mod* is a fuzzy modifier.

Using the above sets, an axiom is either a general concept inclusion of the form $\langle C \sqsubseteq D \rhd \alpha\rangle$, $\langle C \sqsubseteq \neg D \rhd \alpha\rangle$, or $\langle C_1 \equiv C_m\rangle$ for (possibly complex) fuzzy concepts C and D, where $\rhd \in \{\geqslant, >\}$, disjoint concept declaration of the form $C_1 \sqcap C_m \sqsubseteq \bot$ or disjoint union of concepts of the form $\coprod_{1<m} C_1 \sqsubseteq C$, $C_1 \sqcap C_m \sqsubseteq \bot$, an individual assertion of the form $\langle C(a) \bowtie \alpha\rangle$, $\langle R(a,b) \bowtie \alpha\rangle$, $\langle \neg R(a,b) \bowtie \alpha\rangle$, $\langle T(a,v) \bowtie \alpha\rangle$, $\langle \neg T(a,v) \bowtie \alpha\rangle$, $\langle a \approx b\rangle$, or $\langle a \not\approx b\rangle$, where a, b are abstract individuals and $\bowtie \in \{\geqslant, >, \leqslant, <\}$, or a role assertion of the form $R_1 \equiv \ldots R_m$, $T_1 \equiv \ldots T_m$, $\exists R.\top \sqsubseteq C$, $\top \sqsubseteq \forall T.\mathrm{d}$, $\top \sqsubseteq \forall R.C$, $\top \sqsubseteq \leqslant 1S.\top$, $\top \sqsubseteq \leqslant 1T.\top$, $R \equiv R^{-}$, $R \circ R \sqsubseteq R$, $\mathrm{dis}(S_1, \ldots, S_m)$, $\mathrm{dis}(T_1, \ldots, T_m)$, $\langle R_1 \ldots R_m \sqsubseteq R \rhd \alpha\rangle$, $\langle T_1 \sqsubseteq T_2 \rhd \alpha\rangle$, $\top \sqsubseteq \exists R.\mathrm{Self}$, $\forall R.C \sqsubseteq \exists R.\neg D$, $\forall R.C \sqcap \forall R.D$, $C \sqcap \exists R.\neg D$, $\forall R.D$, or Allen's relations of the form $R_A(C, D)$, where $R_A \in \{$b, m, o, s, d, f, a, mi, oi, si, di, fi$\}$.

3.4.3 Importance in Spatiotemporal Reasoning

Ontology-based knowledge representation is known to be efficient in the formal representation of audio and visual contents, however, multimedia ontologies often lack formal grounding. While knowledge representation formalisms could be combined for video event representation, the literature indicates the extended use of rule-based systems for this purpose, which often breaks decidability.

Our formalism supports role negation at the ABox level, enabling us to check whether the RCC relations hold or infer the disconnectivity of regions. To retain decidability, the temporal part of the presented formalism supports existential temporal quantifiers only, but universal temporal quantifiers are not permitted. In other words, it implements a fragment of the interval-based HS-logic. Fuzzy concepts and fuzzy roles are formalized in a way that they support OWL 2 implementations. Experiments show the potential of the resulting knowledge representation formalism in video scene interpretation via spatiotemporal reasoning. The only limiting factor is that not all reasoners can handle temporal and fuzzy axioms, but this holds for other formalisms used for extending the expressivity of OWL as well, such as SWRL rules. Nevertheless, the presented formalism can answer mixed SPARQL queries[24] over general, spatial, temporal, and fuzzy relation types. This is an advancement compared to former approaches, for which formalizing complex video events by combining primitive actions has been feasible only for well-defined knowledge domains with constrained actions and environment, such as for news videos and sports videos.

3.5 Summary

This chapter introduced the reader to knowledge engineering and the difference between unstructured, semistructured, and structured data. The core Semantic Web standards have been described, such as RDF, RDFS, and OWL. The importance of the formal grounding of OWL ontologies with description logics has also been explained, and a novel hybrid description logic-based formalism introduced for spatiotemporal annotation and reasoning.

---

[24] SPARQL is a very powerful query language specially designed for structured data querying. https://www.w3.org/TR/sparql11-query/

Chapter 4

Ontology Implementations

> "The limits of your language
> are the limits of your world."
>
> — *Ludwig Wittgenstein*

The benefits of formal video representation cannot be exploited in multimedia applications until the definitions of terms for the depicted knowledge domain are implemented in an OWL ontology. While there is a wide variety of multimedia ontologies, selecting the most appropriate one for a particular medical video scene is not a trivial task, particularly when there are overlapping ontologies, most of which are not standardized. Some biomedical ontologies may be excellent for representing background knowledge, but these have to be complemented by ontologies purpose-built for medical imaging in order to obtain a useful ontological representation for medical video scenes.

4.1 Common Vocabularies and Ontologies for Video Representation

The declaration of a video clip depicting a medical condition in a machine-readable format requires the formal definition of video clips and their features to be retrieved from a vocabulary or ontology, such as the Clip vocabulary from Schema.org, which is suitable for declaring the file format, language, encoding, and other properties of video clips (`schema:Clip`). The definition of the "depicts" relationship can be found in the Video Ontology (VidOnt) (`video:depicts`). The definition of "medical condition" can be used from `health-lifesci:MedicalCondition`,[1] which defines typical properties of medical conditions, including, but not limited to, cause, epidemiology, expected prognosis, possible complications, possible treatment, risk factor, and stage. Since many multimedia ontologies provide very generic or very specific concept definitions only, video descriptions usually utilize more than one ontology.

The vocabularies of core audio and video metadata standards have originally been created in XML or XML Schema (XSD),[2, 3, 4] which made them machine-readable, but not machine-interpretable. *Semantic Web* standards, such as RDF, RDFS, and OWL, can overcome this limitation, which resulted in several attempts for the RDFS or OWL mapping of ID3,[5] Dublin Core,[6] TV-Anytime,[7] MPEG-7, or a combination of these [83]. Hunter's MPEG-7 ontology was the first of its kind; it modeled the core parts of MPEG-7 in OWL Full, complemented by DAML+OIL constructs [84]. Inspired by this mapping, Tsinaraki et al. created another MPEG-7 ontology, but with full coverage for the MPEG-7 Multimedia Description Scheme (MDS) [85]. *Rhizomik* (MPEG-7Ontos),[8] the first complete MPEG-7 ontology, was generated using a transparent mapping from XML to RDF combined with mapping XSD to OWL [86]. The *Visual Descriptor Ontology (VDO)* was an OWL DL ontology, which covered the visual components of MPEG-7 [87]. The *Multimedia Structure Ontology (MSO)* defined basic multimedia concepts from the MPEG-7 MDS [88]. Oberle et al. created an ontological framework to formally model the MPEG-7 descriptors and export them into OWL [89]. The *Multimedia Content Ontology (MCO)* and the *Multimedia Descriptors Ontology (MDO)* cover the MPEG-7 MDS structural descriptors, and the visual and audio parts of MPEG-7 [90]. The *Core Ontology for Multimedia (COMM)* mapped selected parts of the MPEG-7 vocabulary to OWL

---

[1] It is a common practice to abbreviate terms using the *namespace mechanism*, which uses a prefix instead of full (and often symbolic) URIs. For example, `schema:` abbreviates `http://schema.org/`, therefore `schema:Clip` stands for http://schema.org/Clip, `video:depicts` abbreviates `http://purl.org/ontology/video/depicts`, and `health-lifesci:` is the prefix for `http://health-lifesci.schema.org/`, i.e., `health-lifesci:MedicalCondition` stands for `http://health-lifesci.schema.org/MedicalCondition`.

[2] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=34230

[3] http://purl.org/NET/mco-core, http://purl.org/NET/mco-ipre

[4] http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=39864

[5] http://www.semanticdesktop.org/ontologies/2007/05/10/nid3/

[6] http://dublincore.org/2012/06/14/dcterms.rdf

[7] http://rhizomik.net/ontologies/2005/03/TVAnytimeContent.owl

[8] http://rhizomik.net/ontologies/2005/03/Mpeg7-2001.owl

[91]. Although COMM is one of the very few multimedia ontologies that exploited all the mathematical constructors available in the underlying formal language at the time of its release, it is not without flaws, and does not support spatiotemporal reasoning.

On top of these ontologies, several OWL ontologies have been created for de facto standards and many without standards alignment.[9] W3C's *Ontology for Media Resources*[10] provides a core vocabulary with standards alignment to be used in online media resource descriptions. The *Multimedia Metadata Ontology (M3O)*[11] was designed to integrate the core aspects of multimedia metadata [92]. The *Linked Movie Database*[12] was designed for describing common concepts and properties of Hollywood movies, such as actor, director, etc. Unfortunately, the naming convention of the Linked Movie Database is not optimal, and the ontology is not suitable for describing videos other than films, for example, music video clips, tutorial videos, teasers, commercials, news videos, and sports videos. The *Video Ontology (VidOnt)*[13], which will be described in detail in Chapter 6, is a core reference ontology that integrates viewpoints of de facto standard and standard video-related upper ontologies and domain ontologies with important concepts and roles, resulting in the most expressive video ontology to date. VidOnt addresses many of the aforementioned issues, provides a novel concept and role hierarchy, defines terms for spatiotemporal annotation of complex video scenes, and integrates low-level and high-level video descriptors with standards alignment. The *3D Modeling Ontology*[14] can be used for describing characters and objects of computer animations, including computer generated imagery (CGI) and computer-aided design (CAD).[15] The *STIMONT* ontology can describe emotions associated with video scenes [93].

---

[9] Sikos, L. F., Powers, D. M. W. (2015) Knowledge-driven video information retrieval with LOD: from semi-structured to structured video metadata. In: Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, pp. 35–37. Melbourne, VIC, Australia, 23 October 2015. New York: ACM. doi: 10.1145/2810133.2810141

[10] http://www.w3.org/TR/mediaont-10/

[11] http://m3o.semantic-multimedia.org/ontology/2009/09/16/annotation.owl

[12] http://www.linkedmdb.org

[13] http://videoontology.org/vidont.ttl

[14] http://3dontology.org/3d.ttl

[15] Sikos, L. F. (2017) 3D model indexing in videos for content-based retrieval via X3D-based semantic enrichment and automated reasoning. *Invited paper*. In: 22nd International Conference on 3D Web Technology. Brisbane, QLD, Australia, 5–7 June 2017. New York: ACM. doi: 10.1145/3055624.3075943

Wordnet[16] and *OpenCyc*[17] are two well-established upper ontologies[18] that can be used for describing a variety of concepts, including those depicted in videos. Alternatively, ontologies specifically designed for this purpose, such as the *Large-Scale Concept Ontology for Multimedia (LSCOM)* [94] or the ontology of Zha et al. [95] can also be used.

The spatiotemporal annotation of video events can employ spatial ontologies, such as ontologies based on the aforementioned RCC8 calculus [96], temporal ontologies, such as the *SWRL Temporal Ontology*,[19] fuzzy ontologies, such as the *Video Semantic Content Model (VISCOM)* [97], formally grounded in not only general, but also in spatial, temporal, and fuzzy description logics,[20] and Media Fragment URI 1.0 identifiers.[21]

Schema.org provides de facto standard definitions for a variety of knowledge domains, which also include coverage for concepts and properties that frequently appear in multimedia contents. For example, audio resources can be described with `schema:bitrate`, `schema:encodingFormat`, and `schema:duration`. Similarly, videos can be described using `schema:video` and `schema:VideoObject`. Seasons, film series, episodes of series, and movies can be annotated with `schema:CreativeWorkSeason`, `schema:MovieSeries`, `schema:Episode`, and `schema:Movie`. Music and movie genres can be defined using `schema:genre`. The age rating of videos can be declared using `schema:childMinAge`.

The terms of the previous ontologies, when embedded in the website markup, can be indexed by all major search engines, including Google, Yahoo!, and Bing [98].

4.2 Common Medical Ontologies

Similar to other domains, in biomedicine the formal representation of concepts, properties, and correlations started with semistructured, rather than structured knowledge organization systems. For example, the *Digital Imaging and Communications in Medicine (DICOM)*, the international standard that provides interoperability between medical imaging devices and technologies (ISO 12052), is

---

[16] http://wordnet-rdf.princeton.edu/ontology

[17] https://sourceforge.net/projects/texai/files/open-cyc-rdf/1.1/

[18] They correspond to two knowledge bases of the same name.

[19] http://swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl

[20] Not all of these formalisms can be implemented in OWL 2, leading to proprietary extensions.

[21] https://www.w3.org/TR/media-frags/

available in XML. The *Open Microscopy Environment*,[22] which provides semantics for microscopy images and associated metadata, is available in XSD. Introduced in the 1960s, the N-Triples serialization of the well-known curated vocabulary, the *Medical Subject Headings (MeSH)*[23] of the U.S. National Library of Medicine has just recently been created, and is still in the beta phase.

OWL ontologies are used in the medical domain for representing healthcare processes and pharmaceutical products, providing personalized home care assistance, detecting and assessing adverse drug reactions, exposing expert knowledge captured from heterogeneous resources, modeling product features, and planning and managing medical services [99]. Common ontology types used in biomedicine include upper ontologies, domain ontologies, core reference ontologies, and application ontologies [100]. Biomedical ontologies are primarily used in the annotation of medical imaging resources and data description with well-defined scientific terms to be used for medical information retrieval, thereby bridging heterogeneous resources that use proprietary terminology and encoding.

However, the increasing number and complexity of biomedical ontologies make it difficult to select the ones to use in a particular scenario. While there is a system, BiOSS, to assist biomedical ontology selection via evaluating the suitability of an ontology to an application area by term coverage, semantic richness, and whether the ontology is a de facto standard [101], the variety of biomedical ontologies is an obstacle in data integration. To counteract this problem, biomedical ontology collections are maintained [102], such as the *NCBO BioPortal*,[24] *OpenClinical*,[25] the *Unified Medical Language System (UMLS)*,[26] *Open Biomedical Ontologies (OBO Foundry)*,[27] and *Ontobee*.[28] Ontologies typically have to meet certain criteria to be included in these; the OBO Foundry collection, for example, requires, among others, versioning support, textual description for all terms, and adequate documentation.

---

[22] https://www.openmicroscopy.org

[23] https://www.nlm.nih.gov/mesh/

[24] https://bioportal.bioontology.org/ontologies

[25] http://www.openclinical.org/ontologies.html

[26] https://www.nlm.nih.gov/research/umls/

[27] http://obofoundry.org

[28] http://www.ontobee.org

Because not all biomedical ontologies listed in these ontology collections are related to healthcare and medical practice, such as many of those that cover life science, biology, chemistry, or biochemistry, only some of them are included in the following sections. These can be used for either providing background knowledge in the form of TBox axioms, or annotating medical images and videos.

4.2.1 Biomedical Ontologies for Representing Background Knowledge

The *Human Disease Ontology*[29] is an ontology for describing the classification of human diseases organized by etiology. It consists of axioms on anatomy, cells, infectious agents, and phenotypes. The *Ontology for General Medical Science*[30] defines the diagnosis and treatment of diseases related to pathological entities, such as carcinomas. The *Cardiovascular Disease Ontology*[31] defines entities related to cardiovascular diseases. The *Infectious Disease Ontology*[32], as its name suggests, defines infectious diseases. The *Genomic Epidemiology Ontology*[33] (GenEpiO) enumerates terms that are used to identify, document, and research foodborne pathogens and associated outbreaks. The *Monarch Disease Ontology*[34] merges several monarch disease resources and was generated semi-automatically from multiple ontologies. The *Symptom Ontology*[35] defines the terminology of disease symptoms, including changes in function, sensation, or appearance perceived by patients and which might indicate a disease. The *Oral Health and Disease Ontology*[36] is suitable for the formal representation of the content of dental practice health records. The *Prescription of Drugs Ontology*[37] defines entities related to drug prescription.

The *Common Anatomy Reference Ontology*[38] (CARO) is an upper ontology that facilitates interoperability between existing anatomy ontologies for different species. It has been extended in the

---

[29] https://raw.githubusercontent.com/DiseaseOntology/HumanDiseaseOntology/master/src/ontology/doid.owl

[30] https://raw.githubusercontent.com/OGMS/ogms/master/src/ontology/ogms.owl

[31] https://raw.githubusercontent.com/OpenLHS/CVDO/master/cvdo.owl

[32] https://raw.githubusercontent.com/infectious-disease-ontology/infectious-disease-ontology/master/releases/2016-07-05/ido-merged.owl

[33] https://raw.githubusercontent.com/GenEpiO/genepio/master/genepio.owl

[34] https://raw.githubusercontent.com/monarch-initiative/monarch-disease-ontology/master/src/mondo/mondo.owl

[35] https://raw.githubusercontent.com/DiseaseOntology/SymptomOntology/master/symp.owl

[36] https://raw.githubusercontent.com/alanruttenberg/ohd-ontology/master/src/ontology/ohd-2016-06-27-merged-inferred.owl

[37] https://raw.githubusercontent.com/OpenLHS/PDRO/master/PDRO.owl

[38] https://raw.githubusercontent.com/obophenotype/caro/master/src/ontology/caro.owl

Anatomical Entity Ontology,[39] which is a comprehensive collection of anatomical structures. The *Vocabulary of Human Developmental Anatomy*[40] is a controlled vocabulary of stage-specific anatomical structures of the developing human. The *Human Developmental Stages Ontology*[41] defines the life cycle stages of humans. The *Obstetric and Neonatal Ontology*[42] is a controlled vocabulary that can be used to represent the data from electronic health records (EHRs) used in the care of pregnant women and their babies.

The *NCI Thesaurus OBO Edition*[43] (NCIt) is a core reference ontology for the cancer domain, covering cancer-related diseases, findings, and abnormalities. The *ACGT Master Ontology* (ACGT-MO) provides a comprehensive coverage of medical terms in the domain of cancer research and management, with special emphasis on mammary carcinoma (breast cancer), Wilms' tumor (nephroblastoma), and rhabdoid tumors [103].

4.2.2 Vocabularies and Ontologies for the Semantic Annotation of Medical Multimedia

While there are general-purpose ontologies that can be used for the structured annotation of medical images, such as the *Web Annotation Ontology*,[44] biomedical ontologies have also been developed in a number of fields related to medical imaging at the macroscopic (organs, disease cells, specimens), microscopic (cell morphology, spatial relations, regions of interest), and molecular level (PET CT, SPECT CT) [104].

The *Annotation and Image Markup (AIM) Ontology* represents entities associated with medical images, including anatomic structures researchers see in images (e.g., human organs), observations radiologists make about images (e.g., opacity and density of structures), anatomic location, and context [105]. *RadLex*[45] is a comprehensive ontology that defines unified radiology terms to be used for standardized indexing and retrieval of radiological images. The *Radiology Gamuts Ontology (RGO)*[46]

---

[39] http://ontologies.berkeleybop.org/aeo.owl

[40] http://ontologies.berkeleybop.org/ehdaa2.owl

[41] http://ontologies.berkeleybop.org/hsapdv.owl

[42] https://raw.githubusercontent.com/ontoneo-project/Ontoneo/master/ontoneo.owl

[43] http://purl.obolibrary.org/obo/ncit.owl

[44] https://www.w3.org/ns/oa

[45] http://radlex.org

[46] http://gamuts.net

represents differential diagnoses in radiology, and can be used for the semantic indexing and retrieval of radiology images created using cardiac, chest, diagnostic, gastrointestinal, genitourinary, head and neck, musculoskeletal, obstetric, and pediatric radiology, neuroradiology, CT, MRI, breast imaging, ultrasound imaging, oncologic imaging, or vascular imaging [106]. The *Magnetic Resonance Imaging Ontology*, for example, was specially designed for the semantic description of MRI images and image sequences [107]. The *Biological Imaging Methods Vocabulary*[47] is a controlled vocabulary of sample preparation, visualization, and imaging methods used in biomedical research. The *OntoNeuroBase* application ontology was designed for semantic annotations in neuroimaging to help explore brain structure and function [108].

The *Ontology for Biobanking*[48] is suitable for the annotation and modeling of biobank repositories and administering biobanks, in which several entity types are relevant to imaging, such as specimens and their preparation methods (e.g., staining). By using this ontology, data and entities such as samples and images can be formally described and associated with patient data, such as demographics. The *Quantitative Imaging Biomarker Ontology*[49] describes various concepts in quantitative imaging biomarkers. The *Ontology for Biomedical Investigations (OBI)* contains some terms relevant to medical imaging and pathology, including imaging assay and staining.

4.3 Structured Video Data Deployment

RDF triples, which can use arbitrary terms from RDFS vocabularies and OWL ontologies, are typically serialized in structured datasets and purpose-built graph databases in a mainstream RDF serialization format, such as RDF/XML or Turtle.

4.3.1 Lightweight Video Annotations

The RDF triples of both videos and background knowledge associated with videos can be deployed as lightweight annotations embedded in the website markup in RDFa, HTML5 Microdata, and

---

[47] http://ontologies.berkeleybop.org/fbbi.owl

[48] https://raw.githubusercontent.com/biobanking/ontology/20170728/src/obib.owl

[49] http://purl.bioontology.org/ontology/QIBO

JSON-LD. RDFa (“RDF in attributes”) expresses RDF in markup attributes that are not part of the core (X)HTML vocabularies. HTML5 Microdata extends the HTML5 markup with structured metadata. JSON-LD (JavaScript Object Notation for Linked Data) adds structured data to the markup as a script code block. Note that RDFa and JSON-LD can be used in most markup language versions and variants, while HTML5 Microdata can be used in (X)HTML5 only, and that all these annotation formats have their own syntax. For example, the vocabulary is declared with the `vocab` attribute in RDFa, the `itemtype` attribute in Microdata, and `context` in JSON-LD (see Table 4.1).

Table 4.1 Structured data serialized in RDFa, HTML5 Microdata, and JSON-LD

Markup without semantic annotation

```
Bones are part of the skeletal structure. One of their major
functions is support.
```

Markup with RDFa

```
<div vocab="http://health-lifesci.schema.org/" typeof="Bone">
  Bones are part of the <span property="partOfSystem">skeletal
  structure</span>. One of their major functions is <span
  property="function">support</span>.
</div>
```

Markup with HTML5 Microdata

```
<div itemscope="itemscope"
itemtype="http://health-lifesci.schema.org/Bone">
  Bones are part of the <span itemprop="partOfSystem">skeletal
  structure</span>. One of their major functions is <span
  itemprop="function">support</span>.
</div>
```

Markup with JSON-LD

```
<script type="application/ld+json">
  {
    "@context": "http://health-lifesci.schema.org",
    "@type": "Bone",
    "partOfSystem": "skeletal structure",
    "function": "support"
  }
</script>
```

4.3.2 Video Representations in Graph Databases

To leverage the power of RDF, structure video data may be stored in graph databases rather than traditional relational databases. A graph database is a database that stores RDF statements and implements graph structures for semantic queries, using nodes, edges, and properties to represent and retrieve data. A few graph databases are based on relational databases, while most are purpose-built from the ground up for storing and retrieving RDF statements.

There are two important properties of graph databases that determine efficiency and implementation potential. The first one is the storage, which can be native graph storage or a database engine that transforms an RDF graph to relational, object-oriented, or general-purpose database structures. The other main property is the processing engine. True graph databases implement so-called index-free adjacency, whereby connected nodes are physically linked to each other in the database. Because every element contains a direct pointer to its adjacent element, no index lookups are necessary. Graph databases store arbitrarily complex RDF graphs by simple abstraction of the graph nodes and relationships. Unlike other database management systems, graph databases do not use inferred connections between entities using foreign keys, as in relational databases, or other data, such as the ones used in MapReduce. The computational algorithms are implemented as graph compute engines, which identify clusters and answer queries.

Some of the most widely deployed, high-performance graph databases are AllegroGraph, Neo4j, Blazegraph, Virtuoso, Stardog, 4Store, Jena TDB, RDF4J, and GraphDB. However, using a native graph database engine is not the only option to store triples and quads. Some examples are Oracle Spatial and Graph with Oracle Database, Jena with PostgreSQL, and 3Store with MySQL.

One of the main advantages of graph databases over relational databases and NoSQL stores is performance [109]. Graph databases are typically thousands of times more powerful than conventional databases in terms of indexing, computing power, storage, and querying. In contrast to relational databases, where the query performance on data relations decreases as the dataset grows, the performance of graph databases remains relatively constant.

While relational databases require a comprehensive data model about the knowledge domain upfront, graph databases are inherently flexible, because graphs can be extended with new nodes and new relationship types effortlessly, while subgraphs merge naturally with their supergraph.

Because graph databases implement freely available standards such as RDF for data modeling and SPARQL for querying, the storage is usually free of proprietary formats and third-party dependencies. Another big advantage of graph databases is the option to use arbitrary external vocabularies and schemas, while the data is available programmatically through Application Programming Interfaces (APIs) and powerful queries.

All graph databases designed for storing RDF triples are called triplestores or subject-predicate-object databases, however, the triplestores that have been built on top of existing commercial relational database engines (such as SQL-based databases) are typically not as efficient as the native triplestores with a database engine built from scratch for storing and retrieving RDF triples. The performance of native triplestores is usually better, due to the difficulty of mapping the graph-based RDF model to SQL or NoSQL queries.

The advantages of graph databases are derived from the advantageous features of RDF, OWL, and SPARQL. RDF data elements are globally unique and linked, leveraging the advantages of the graph structure. Adding a new schema element is as easy as inserting a triple with a new predicate. Graph databases also support ad hoc SPARQL queries. Unlike the column headers, foreign keys, or constraints of relational databases, the entities of graph databases are categorized with classes; predicates are properties or relationships; and they are all part of the data. Due to the RDF implementation, graph databases support automatic inferencing for knowledge discovery. The data stored in these databases can unify vocabularies, dictionaries, and taxonomies through machine-readable ontologies. Graph databases are commonly used in semantic data integration, social network analysis, and Linked Open Data applications.

It is not always possible to interpret RDF statements without a graph identifier. For example, if a given name is used as a subject, it is out of context if we do not state the patient we want to describe. If, however, we add the web site address of the same person to each triple that describes the same person, all components become globally unique and dereferenceable. A quad is a subject-predicate-object triple coupled with a graph identifier. The identified graph is called a named graph.

4.3.3 Semantic Enrichment of Videos with Linked Data

Records of isolated video databases, particularly when locked down and using proprietary data formats, are inefficient in data access, sharing, and reuse.

To enable semantic queries across diverse resources, structured data can be published according to best practices, called *Linked Data*, by using *typed RDF links* and unique identifiers [110]. The four principles of Linked Data are the following [111]:

1. Uniform Resource Identifiers (URIs), i.e., strings of ASCII characters of the form `scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]`, should be used to represent real-world concepts and entities. By assigning a dereferenceable URI to each resource, they become individually identifiable (as opposed to an application-specific identifier, such as a database key or incremental numbers).
2. The URIs of RDF resources should be HTTP URIs, so that the resource names can be found on the Internet.
3. The resource URIs should provide useful information using Semantic Web standards, such as RDF.
4. Links to other URIs should be included, enabling users and software agents to discover related information.

Creating links between the structured datasets of the Semantic Web is called *interlinking*, which makes isolated datasets part of the *LOD Cloud*,[50] in which all resources are linked to one another. These links enable semantic agents to navigate between data sources and discover additional resources. The most common predicates for interlinking are `owl:sameAs` and `rdfs:seeAlso`, but any predicate can be used. In contrast to hyperlinks between web pages, LOD links utilize typed RDF links between resources.

Linked Data with an explicitly stated open license is called *Linked Open Data (LOD)*. A meaningful collection of RDF triples that complies with Linked Data principles and is published with an open license is called an *LOD dataset*, which is the second type of structured data resource beyond OWL ontologies for term definitions of semantic video descriptions. The LOD-based semantic en-

---

[50] http://lod-cloud.net

richment of videos is employed by video repositories, hypervideo applications, and video streaming portals, such as YouTube [112].

4.4 Summary

This chapter enumerated the most common vocabularies and ontologies used in formal video representation in general, and the most common biomedical ontologies for medical video annotation. The options for structured data deployment have been listed, including triplestores and lightweight annotation in the website markup. The principles of Linked Data and the benefits of LOD datasets have also been described.

Chapter 5
Ontology-Based Structured Video Annotation

> “The key to artificial intelligence has
> always been the representation.”
>
> — *Jeff Hawkins*

The semantic enrichment of videos with structured annotations employing ontology terms can be used for efficient content-based video indexing and retrieval, and for knowledge discovery based on the fusion of formally described background knowledge about the depicted concepts and low-level feature descriptors. While most high-level descriptors can be associated with a video scene manually or collaboratively only, some structured video annotations can be made semi-automatically using purpose-built software tools.

5.1 Semantic Video Annotation

State-of-the-art structured video annotation incorporates multimedia signal processing and formally grounded knowledge representation, video feature extraction, machine learning, ontology engineering, and multimedia reasoning.

5.1.1 Feature Extraction for Concept Mapping

A wide range of well-established algorithms exists for automatically extracting low-level video features, for example, fast color quantization to extract the dominant colors [113] or Gabor filter banks to extract homogeneous texture descriptors [114]. There are also advanced algorithms for video content analysis, such as the Viola-Jones and Lienhart-Maydt object detection algorithms [115, 116], and the SIFT, SURF, and ORB keypoint detection algorithms [117, 118, 119]. The corresponding descriptors can be used as positive and negative examples in machine learning, such as support vector machines (SVM) and Bayesian networks, for keyframe analysis, face recognition, and video scene understanding.

While useful, many automatically extracted low-level video features are inadequate for representing video semantics. For example, annotating the dominant color or color distribution of a frame does not provide the meaning of the visual content. In contrast, high-level descriptors are suitable for video concept mapping, but they often rely on human knowledge, experience, and judgment. However, manual video concept tagging is very time-consuming, might be biased, too generic, or inappropriate, which has led to the introduction of collaborative semantic video annotation, where multiple users annotate the same resources and improve each other's annotations [120]. User-supplied annotations can be curated using natural language processing to eliminate duplicates and typos, and filter out incorrectly mapped concepts. The integrity of manual annotations captured as structured data can be confirmed automatically using LOD definitions. Research results for high-level concept mapping in constrained videos, such as medical videos [121] or sport videos [122], are already promising, however, concept mapping in unconstrained videos is still a challenge [123].

The next section details multimedia ontology engineering best practices to create machine-interpretable high-level descriptors and reuse de facto standard definitions to formally represent human knowledge suitable for the automated interpretation of video contents.

5.1.2 Knowledge Representation of Video Scenes

Logical formalization of video contents can be used for video indexing, scene interpretation, and video understanding. Concept definitions and factual data might be used from any kind of structured data source, such as controlled vocabularies, ontologies, and LOD datasets. As mentioned before, two of the most well-established commonsense knowledge bases and their corresponding upper ontologies that can be used for describing the concepts depicted in videos are Wordnet and OpenCyc. There are also more specific ontologies for this purpose, such as the aforementioned Large-Scale Concept Ontology for Multimedia (LSCOM). The spatiotemporal annotation of video events requires even more specialized ontologies, such as the SWRL Temporal Ontology, along with other formalisms and standards, such as Media Fragment URI 1.0 identifiers.

If the concepts to be described belong to a knowledge domain not covered by existing ontologies, one can create a new ontology by formally defining the classes, properties, and their relationships, preferably in OWL, with a logical underpinning in description logics. DL-based ontologies do not specify a particular interpretation based on a default assumption; instead, they consider all possible cases in which the axioms are satisfied.

5.1.3 Ontology-Based Video Indexing and Retrieval

The spatiotemporal annotation of medical videos is useful for identifying a video scene of a medical procedure, such as Nd:YAG[1] laser vitreolysis for treating bothersome vitreous opacities that negatively affect the quality of life [124], with temporal segmentation, annotating the region of interest, such as an eye floater, as a moving region with spatiotemporal segmentation, and describing the medical video scene and the concept depicted in the scene. By using Media Fragment URI 1.0 identifiers, the spatiotemporal segmentation can be done by specifying the positions in Normal Play Time format according to RFC 2326,[2] and the region of interest represented by its minimum bounding box, as shown in Fig. 5.1.

---

[1] Neodymium-doped yttrium aluminum garnet

[2] https://www.ietf.org/rfc/rfc2326.txt

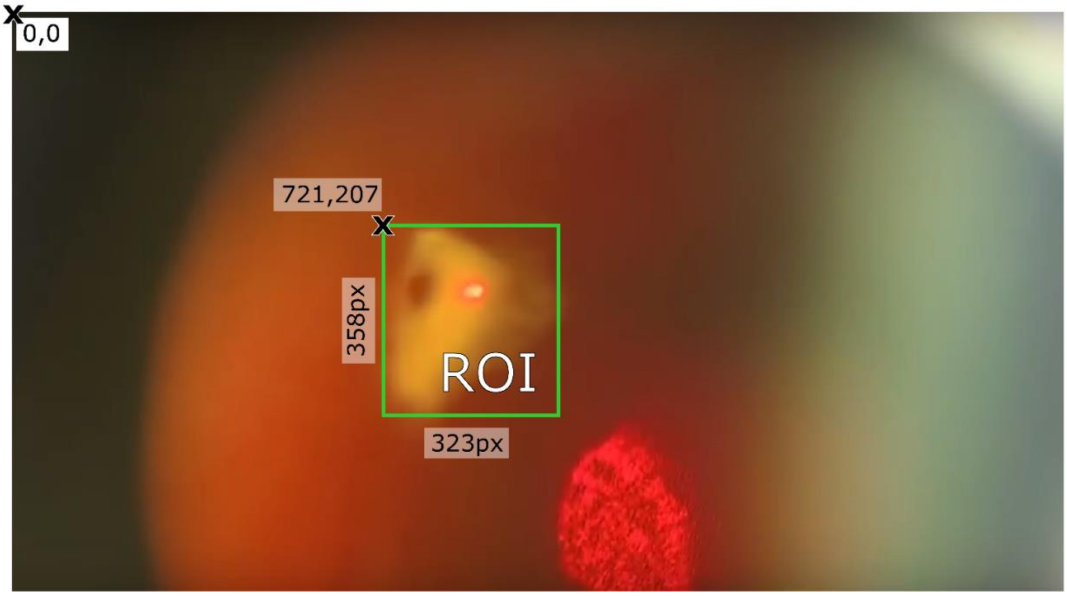


Figure 5.1 Spatial annotation of the region of interest depicting a Weiss ring floater by the top left corner coordinates and dimensions of its minimum bounding box. Video by James H. Johnson[3]

Using a description logic formalism, this video scene can be represented as shown in Listing 5.1.

Listing 5.1 Spatiotemporal description of a video scene with DL formalism

```
Video(LASERVITREOLYSISVIDEO)
Scene(FLOATERSCENE)
sceneFrom(FLOATERSCENE, LASERVITREOLYSISVIDEO)
hasStartTime(FLOATERSCENE, 00:00:00)
duration(FLOATERSCENE, 50)
hasFinishTime(FLOATERSCENE, 00:00:50)
depicts(FLOATERSCENE, LaserVitreolysis)
StillRegion(FLOATERSCENEROI)
depicts(FLOATERSCENEROI, Floater)
```

This formal description can be written in any RDF serialization. Listing 5.2 shows the Turtle serialization of the above example.

---

[3] https://www.youtube.com/watch?v=hZibbnjiGuY

Listing 5.2 Spatiotemporal description of a video scene in Turtle

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix mpeg7: <http://mpeg7.org/> .
@prefix snomedct: <http://purl.bioontology.org/ontology/SNOMEDCT/> .
@prefix temporal: <http://swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl> .
@prefix video: <http://purl.org/ontology/video/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://example.com/floater.mp4> a mpeg7:Video .
<http://example.com/floater.mp4#t=0:00:00,0:00:50> a
 video:Scene ; video:temporalSegmentOf <http://example.com/floater.mp4> ;
 video:sceneFrom <http://example.com/floater.mp4> ;
 temporal:hasStartTime "00:00:0"^^xsd:time ;
 temporal:duration "PT00M00S50"^^xsd:duration ;
 temporal:hasFinishTime "00:00:50"^^xsd:time .
<http://example.com/floater.mp4#t=0:00:48,0:00:50&xywh=721,207,358,323> a
 mpeg7:StillRegion ;
 video:spatioTemporalSegmentOf <http://example.com/floater.mp4> ;
 video:depicts snomedct:15013002 , dbpedia:Floater .
```

The formal definition of the medical term “vitreous floater” used in this video scene description is retrieved from SNOMED CT and DBpedia. MPEG-7, VidOnt, and the SWRL Temporal Ontology are utilized for spatiotemporal segmentation, and XSD for the datatype declarations. The namespaces for these ontologies are defined using `@prefix` as usual, and the corresponding prefixes are used in the RDF triples. Temporal segments are identified for the `floater.mp4` video file by the starting and ending position separated by a comma, which is preceded by `#t=`. The URI of the video is, similar to any URI in Turtle, delimited by < and >. The spatiotemporal segment for the region of interest extends this by the top left coordinates and dimensions of the minimum bounding box of the region depicting the vitreous opacity separated by commas, and preceded by `&xywh=` in the URI.

Note. A shorthand notation for RDF triples that share the same object is to state the subject and the predicate once only, followed by a comma-separated list of objects.

5.1.4 Primary Application Areas

While our primary focus here is the medical domain, multimedia ontologies can be used for high-level scene interpretation in a wide range of applications, such as event detection [125], moving object detection and tracking [126], and even human intention detection [127]. High-level scene interpretation is suitable for, among others, classification, video surveillance [128], intelligent video analytics, and real-time activity monitoring [129]. Most of these tasks are performed by reasoning over the video contents to recognize situations and temporal events based on human knowledge formally described as ontology concepts, roles, individuals, and rules. By representing fuzzy relationships between the context and depicted concepts of video contents, both deductive and abductive reasoning can be performed [130].

5.2 Structured Video Annotation Tools

To identify industry requirements of structured video annotations in terms of practicality, user interface considerations, standards alignment, semantics, and automation, a comprehensive literature review has been conducted.[4] Because the medical image annotation tools cannot handle the temporal data needed for medical video annotation, it is essential to differentiate between those tools that are specially designed for images and those that are video annotation tools. Note that some image annotation tools might support image sequences as well, and can open sets or series of images for annotation (see Fig. 5.2).

---

[4] Sikos, L. F. (2016) RDF-powered semantic video annotation tools with concept mapping to Linked Data for next-generation video indexing. Multimedia Tools and Applications, 76(12):14437–14460. doi: 10.1007/s11042-016-3705-7

Figure 5.2 Annotation of regions of interest in the Medical Annotation Tool[5] for an image that is part of an image sequence

To reason over symptoms present in multiple patient records, the semantic annotations of medical multimedia resources should hold provenance data, along with timestamps for the video frame, or intervals for the video shot, that depicts the related concepts, actions, and visual features. For this reason, in contrast to papers on multimedia annotation tools that mix the annotation of still images and videos, or do not differentiate between semi-structured and structured output, here only the milestones of structured video annotation tools are enumerated, and their limitations highlighted.

5.2.1 A Retrospective Survey of Semantic Video Annotation Tools

Veggie, one of the first video annotation tools to generate RDF output, was introduced by Hunter and Newmarch in 1999 [131]. The Java application produced Dublin Core-based metadata descriptions and video summaries for MPEG-1 videos.

In 2002, Heggland developed *OntoLog*, an application for searching and browsing temporal media metadata by leveraging metadata exchange using RDF, and SMIL for interoperability between

---

[5] https://www.youtube.com/watch?v=BWbhtjyMdEI

different media players [132]. The software supported high-level descriptors not only for entire videos, but also for video shots and frames. OntoLog incorporated RDFS for representing depicted concepts and the relationships between them.

Vannotea, also released in 2002, and updated in 2006, was a prototype system for synchronous indexing, browsing, and real-time collaborative annotation and commentary of MPEG-2 videos (see Fig. 5.3).

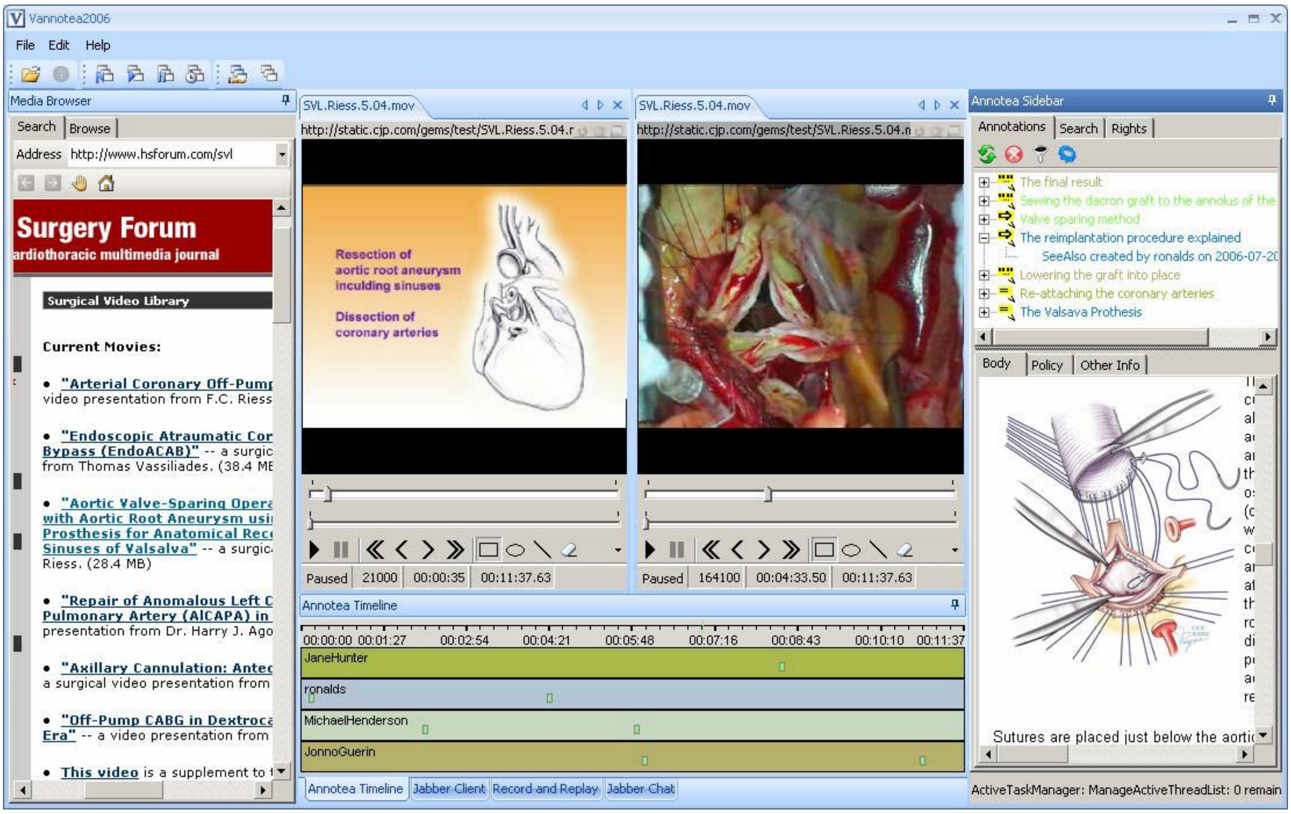


Figure 5.3 Vannotea, an early implementation of structured video annotation tools[6]

Vannotea was based on W3C's *Annotea*,[7] and used RDF for knowledge representation and XPointer to link the annotations to the video resources.

Advene[8] *(Annotate Digital Video, Exchange on the Net)*, also released in 2002, was developed over a decade, and is still available to download today. In Advene, users can annotate video fragments at arbitrary positions, save the semantically enriched videos, and play the videos with the associated

---

[6] http://www.itee.uq.edu.au/eresearch/projects/vannotea

[7] https://www.w3.org/2001/Annotea/

[8] http://advene.org

semantics. Being a proprietary binary format, the native file format of Advene is not ideal. Nevertheless, the software is Linked Data-ready, because every annotation, relation, and view is identified by a URI and RDF/XML output is supported. The software does not incorporate multimedia ontologies beyond FOAF and Dublin Core though.

OntoMedia[9] was developed in 2006 for large multimedia collection management using Semantic Web technologies. The graphical user interface of this standalone Java application offered easy metadata indexing and video retrieval. OntoMedia accepted any input media supported by QuickTime or the Java Media Framework, and could generate RDF and relational database output.

Also in 2006, Bertini and his colleagues developed the *Multimedia Ontology Manager* (MOM) to combine multimedia ontology engineering with automatic annotation, and generate textual and auditory commentary for video sequences [133]. The automatic video annotation was performed for entire video clips by using similarity checking between visual ontology concepts and extracted clips, and for video sequences by using composite concept patterns. Video clip sequences were annotated with predefined articulated sentences curated by the RACER reasoner.

Annomation,[10] published in 2008 as a collaborative Linked Data-driven narrative hypervideo application, allowed users to semantically annotate video resources using controlled vocabularies defined in the LOD Cloud. It was restricted to predefined videos hosted by the service, and the semantic annotations were saved in a local repository, making them inaccessible to external semantic agents.

In 2009, the *LEMO Annotation Framework* was released, providing a uniform, multimedia-enabled annotation model. LEMO addressed video fragments using the MPEG-21 Part 17 (Fragment Identification of MPEG Resources) standard, and exposed data as Linked Data [134].

IMAS, also published in 2009, was a web-based annotation tool for media resources that generated annotations using a set of proprietary ontologies [135]. IMAS imported images and videos from a media repository, but did not support media fragments. The output of IMAS was suitable for producers only, rather than general-purpose online publishing.

---

[9] http://www.ontomedia.de

[10] http://annomation.open.ac.uk/annomation

SemWebVid[11] was an Ajax web application released in 2010, which automatically generated YouTube[12] video descriptions in RDF, taking manually added tags and closed captions into account. SemWebVid implemented natural language processing APIs to analyze the descriptors, and mapped the results to LOD concepts, using the DBpedia, Uberblic, Any23, and rdf:about APIs, and the now-discontinued Sindice API. Provenance data was color-coded, which was an original idea, however, the resulting text was not always easy to read [136]. The application implemented YouTube Data API v2, which has been replaced by the backward-incompatible YouTube Data API v3 in April 2015. Consequently, SemWebVid is not working anymore.

Also in 2010, Choudhury and Breslin introduced a framework to annotate and retrieve online videos with light semantics, and integrate structured video annotations into the Linked Open Data Cloud by reusing important terms from Dublin Core, FOAF, and SKOS [137]. In the same year, the *Europeana Connect Media Annotation Suite* (ECMAS) was released, which used both plain text and semantic tags for the knowledge representation of depicted concepts [138].

Pan, an ontology-based online video annotation tool to import and edit OWL ontologies with MPEG-7 alignment, was also developed in 2010 [139]. Pan can browse videos, provide a mechanism for the user to select concepts from an ontology structure, add and edit annotations, and load previously saved annotations. The annotations are managed by another tool, Orione, an ontology-based search engine. Pan is not future-ready, because it was written in Adobe Flex and ActionScript 3, i.e., it requires the Flash plugin, which is now deprecated in favor of HTML5 and JavaScript.

YUMA was an annotation software released in 2011, which supported image, audio, and video files [140]. YUMA suggested DBpedia and GeoNames terms, and exported the results to RDF using a proprietary vocabulary, along with LEMO and Open Annotation.

The *ConnectME* toolset was released in 2012, comprising of an HTML5-based semantically enriched video player and an online video annotation tool. The ConnectME framework identified, annotated, and deployed video concepts as Linked Data [141]. The user interface displayed timestamps next to the video player, along with the corresponding labels and LOD URIs, although using prefixes would have made the URIs more compact, easier to read, and easier to fit in the pro-

---

[11] http://tomayac.com/semwebvid/

[12] https://www.youtube.com

gram window (more space would have been reserved for the video player, the search box, and the explorer).

SemTube, a YouTube video annotation prototype, was also released in 2012. It expressed the context of YouTube videos in RDF/OWL and OAC [142]. SemTube used RDF, Linked Data, SPARQL, and RESTful APIs for data import and export. The data retrieval from SemTube annotations supported keyword-based and Linked Data-powered faceted search, and SPARQL queries. One of the preferred LOD datasets for concept mapping in both SemTube and SemWebVid was Freebase, which was discontinued in 2015, with some of its articles transferred to Wikidata.[13]

Many of the annotation tools discussed above were built with proprietary APIs, which have been changed over the years, breaking the functionality of the original program code. The original version of those tools that have not been updated to reflect these API changes stopped working partially or completely. Also, support is limited for most of the software prototypes, which often had a domain name registered at the time of their release, but have later been discontinued. Veggie, OntoLog, OntoMedia, MOM, the LEMO Annotation Framework, IMAS, ECMAS, ConnectME, SemTube, YUMA, and Vannotea are not available online anymore, while SemWebVid and Annomation were available at the time of writing, but were not working.

5.2.2 State-of-the-Art Structured Video Annotation Tools

The *TV Metadata Generator*[14] was released by Eurecom as part of the LinkedTV project in 2011. Based on the local or online input video file, TV-Anytime or EXMARaLDA metadata files, or SRT subtitle files, the software automatically converts television content metadata into RDF. However, the software cannot generate RDF based on the video content alone, and is basically limited to the serialization of existing textual data as structured data. The *LinkedTV Editor*[15] provides a user interface for broadcasting services, which uses the automatically generated annotations of LinkedTV for the rapid generation of contextual information queues.

---

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

[14] http://linkedtv.eurecom.fr/tv2rdf/

[15] http://editortoolv2.linkedtv.eu/

Open Video Annotation[16] is based on open source JavaScript libraries, such as Video.js,[17] Annotator,[18] and RangeSlider.[19] The developers claim that the software is compliant with W3C's Open Annotation data formats. Open Video Annotation was designed to provide an intuitive interface for semantic tagging and the playback of semantically enriched videos; users can take notes on the timeline, view existing annotations, and play annotated video fragments individually.

MyStoryPlayer is a video player capable of the semantic enrichment of multi-angle videos, and was specifically designed for educational videos. It provides an interface for interactive user annotations to be used in action, gesture, and posture analysis, with a focus on the formal representation of relationships between depicted elements in RDF [143]. MyStoryPlayer powers the website of the European eLibrary for Performing Arts (ECLAP),[20] and provides not only general and technical metadata, such as title and duration, but also timestamp-based data, which can be used to annotate presentations, human dialogues, and arbitrary video events. In MyStoryPlayer, metadata and classification are coupled with timestamp-based snapshot comments.

5.2.3 Comparison of Structured Video Annotation Tools

Semantic video annotation tools differ in terms of characteristics and functionality due to the following technical features:

- *Expressivity.* The semantic richness of annotations is determined by the expressivity of the controlled vocabularies and ontologies used for the knowledge representation of the depicted concepts. Some tools are restricted to proprietary controlled vocabulary terms, while others do not provide suggestions but accept arbitrary data.
- *Annotation level.* Annotation software usually specializes in particular types of metadata (technical, administrative, licensing), content descriptors (high-level descriptors), multimedia descriptors (low-level descriptors), structural descriptors (spatial, temporal, and spatiotemporal descriptors), or a combination of these.

---

[16] https://gteavirtual.org/ova/

[17] http://videojs.com

[18] http://annotatorjs.org

[19] https://github.com/andreruffert/rangeslider.js

[20] http://www.eclap.eu

  - *Low-level descriptor support.* Capability to annotate automatically extractable low-level features of videos, such as motion trajectory.
  - *High-level descriptor support.* Capability to precisely annotate depicted concepts and individuals, such as a person, a car, or a building.
  - *Spatial segmentation support.* Enables working with a portion of the media (Region of Interest, ROI) to represent information about the depicted space, for example to annotate a tumor in a medical video or an actor in a movie.
  - *Temporal segmentation support.* Enables frame sequence segmentation within videos to represent time and events, such as video scenes or a goal in a soccer match video.
- *Standards alignment.* Standards make it possible for various platforms and computer systems to communicate with each other and exchange data efficiently, regardless of their structural and functional differences. Standards alignment determines whether standards and de facto standards are implemented (e.g., MPEG-7, Dublin Core, Open Annotation). Video annotation software prototypes may use proprietary formats and mechanisms, which are difficult to implement in large-scale, heterogeneous multimedia systems. Poor standard support, including proprietary vocabulary use, negatively affects interoperability. Open standards are likely to be implemented globally, so they should be preferred.
- *Supported input and output data formats.* Some annotation tools are designed for a particular video compression or codec only (MPEG-1, MPEG-2, MPEG-4 AVC/H.264, MPEG-H HEVC/H.265 etc.), or accept nothing else but YouTube videos by URL. Ideally, the set of supported formats would include at least the current industry standard video file formats. Some video annotation software can handle any kind of video file format, as long as the related codecs are installed on the system.
- *Signal processing integration.* By integrating signal processing algorithms to video annotation tools, the annotation of low-level features becomes seamless, although the majority of automatically extracted low-level descriptors cannot be used for high-level scene interpretation, as mentioned earlier.
- *Linked Data support.* Supporting Linked Data is crucial for semantic multimedia applications. Linked Data provides unique URIs for each video object, media fragment,

keyframe, and ROI, along with a mechanism to interlink depicted concepts with arbitrary definitions from the LOD Cloud, and differentiates media files from web resources that convey information about them. Linked Data support is crucial for future multimedia applications.

- *Automation.* While manual annotations can be the most sophisticated and accurate annotations, they depend on the experience and background of the user, can be misspelt and ambiguous, do not always incorporate the most relevant keywords, and might be biased by personal preferences. Semi-automatic (supervised) and automated (unsupervised) annotation would be desired to address the above issues of manual annotations and to efficiently generate annotations to the rapidly growing number of online videos.
- *Provenance data support.* Storing data source information (preferably by using the PROV Ontology)[21] is beneficial for video annotations derived from diverse data sources. Provenance data makes data quality assessment easier, can be used to find similar or related resources, and makes LOD concept interlinking more efficient.
- *RDF output.* All structured video annotation software must support RDF output in a standard serialization, such as RDF/XML or Turtle. HTML5 Microdata, RDFa, and JSON-LD, which can be directly embedded in the website markup, are also desirable.
- *Architecture.* Web-based semantic video annotation tools are preferred to their desktop counterparts due to benefits such as platform-independence, interoperability, and global availability.
- *Built-in Video Player.* Ideally, video annotation tools are embedded to a video player for seamless annotation and hypermedia playback.

Less objective features include user-friendliness, documentation quality and coverage, user support (examples, tutorial videos, contact), long-term availability, licensing, and whether the software is open source.

The following sections compare structured video annotation tools from the four main perspectives: standards support, input and output data formats, concept mapping sources, and spatiotemporal fragmentation.

---

[21] https://www.w3.org/TR/prov-o/

5.3 Standards Alignment

Several multimedia and web standards are required, and often implemented, in structured video annotation tools to provide backward- and forward-compatibility and interoperability. Standards are vital to gain widespread use, obtain optimality in terms of file structure and code length, and consider global needs. The most common international standards in semantic video annotation are DVD-Video (media and format are defined by multiple standards, e.g., ISO/IEC 16448:2002[22] and ECMA-267,[23] ISO/IEC 25434:2008),[24] MPEG-7 (ISO/IEC 15938),[25] MPEG-21 Part 17 (ISO/IEC 14496-17:2006),[26] Uniform Resource Locators (IETF RFC 1738),[27] and Dublin Core (ISO 15836:2009).[28] The technical specifications used by semantic video annotation tools that have not been standardized officially by a standardization body yet are used globally are known as de facto standards; these include W3C recommendations, such as RDF and SKOS, Open Annotation, and Media Fragment URIs.

5.4 Linked Data Support

While Linked Data output is expected from semantic video annotation tools, dependence on a particular LOD dataset can be a major design issue. A good example is the now-discontinued SemTube, which implemented Freebase as the primary LOD dataset for interlinking, which became obsolete and was succeeded by Wikidata.[29] However, the still popular DBpedia and GeoNames were the primary LOD datasets of Annomation, ConnectME, and YUMA, all of which have also been discontinued. This suggests that the long-term viability of LOD dataset URLs generated by semantic video annotation tools does not guarantee the success of these tools.

---

[22] http://standards.iso.org/ittf/PubliclyAvailableStandards/c035641_ISO_IEC_16448_2002%28E%29.zip

[23] http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-267.pdf

[24] https://www.iso.org/standard/51140.html

[25] https://www.iso.org/standard/34228.html

[26] https://www.iso.org/standard/39478.html

[27] https://www.ietf.org/rfc/rfc1738.txt

[28] https://www.iso.org/standard/52142.html

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

5.5 Ontology Use

The primary concept mapping sources vary greatly among structured video annotation tools, and include ontologies such as Dublin Core,[30] the Ontology for Media Resources,[31] FOAF,[32] Open Annotation,[33] and Representing Content in RDF.[34] Some tools also allow arbitrary ontologies so that they are not limited to the concepts of the primary concept mapping sources (see Table 5.1).

Table 5.1 Ontology use of semantic video annotation tools

Tool	Primary Vocabularies and Ontologies	Arbitrary Ontology
Advene	Dublin Core, FOAF	+
Annomation	Dublin Core, FOAF, SKOS	–
ConnectME	Proprietary, Ontology for Media Resources, Annotation Ontology, Dublin Core, FOAF, Open Annotation, Representing Content in RDF	+
IMAS	SALERO ontologies	+
LEMO	Proprietary	–
LinkedTV Editor	LinkedTV, Ontology for Media Resources, Annotation Ontology, Dublin Core, FOAF, Open Annotation, Representing Content in RDF	–

---

[30] http://dublincore.org/documents/dcmi-terms/
[31] https://www.w3.org/TR/mediaont-10/
[32] http://xmlns.com/foaf/spec/
[33] http://www.openannotation.org/ns/
[34] https://www.w3.org/2011/content

Table 5.1 (Cont'd) Ontology use of semantic video annotation tools

Tool	Primary Vocabularies and Ontologies	Arbitrary Ontology
Open Video Annotation	Annotation Ontology, Dublin Core, FOAF, Open Annotation, Representing Content in RDF	–
SemTube	–	+
SemVidLOD	VidOnt, Schema, Dublin Core, FOAF	+
SemWebVid	–	–
TV Metadata Generator	LinkedTV, Ontology for Media Resources, Annotation Ontology, Dublin Core, FOAF, Open Annotation, Representing Content in RDF	–
Vannotea	Dublin Core	–
YUMA	LEMO, Annotation Ontology, Dublin Core, FOAF, Open Annotation, Representing Content in RDF	–

As shown above, not all annotation tools allow arbitrary ontologies, which is a huge limitation even if standardized ontologies are used as the primary concept mapping sources. However, arbitrary ontology support is not necessarily sufficient to gain global adoption, as was the case of Advene, ConnectME, IMAS, and SemTube.

5.6 Spatiotemporal Annotation Support

Structured video annotation tools support either spatial or temporal annotations, both spatial and temporal annotations, or neither (see Table 5.2).

Table 5.2 Spatiotemporal annotation support of semantic video annotation tools

Tool	Spatial Fragmentation	Temporal Fragmentation
Advene	Proprietary	Proprietary
Annomation	–	Proprietary
ConnectME	Media Fragment URI	Media Fragment URI
IMAS	–	–
LEMO	MPEG-21 Part 17	MPEG-21 Part 17
LinkedTV Editor	Media Fragment URI	Media Fragment URI
MyStoryPlayer	–	–
Open Video Annotation	–	Media Fragment URI
SemTube	Proprietary	Proprietary
SemVidLOD	Media Fragment URI	Media Fragment URI
SemWebVid	–	–
TV Metadata Generator	Media Fragment URI	Media Fragment URI
Vannotea	Proprietary	Proprietary
YUMA	Proprietary	Proprietary

The most common spatiotemporal annotation format in semantic video annotation is W3C’s Media Fragment URI. LEMO used the MPEG-21 Part 17 standard instead for spatiotemporal segmentation. Some tools (Advene, Annomation, and Vannotea) implemented proprietary mechanisms that cannot be processed by any other software tool but the ones that introduced them.

5.7 Summary

This chapter demonstrated the formal knowledge representation of video scenes and content-based video indexing and retrieval using ontology terms. The milestones of semantic video annotation tools were highlighted, and the key factors in selecting tools for video annotation listed.

Chapter 6
Video Ontology Engineering

> "In formal logic, a contradiction is the signal of defeat,
> but in the evolution of real knowledge it marks
> the first step in progress toward a victory."
>
> — *Alfred N. Whitehead*

To address the limitations of multimedia ontologies in video object representation and scene interpretation, two new ontologies have been developed. The first one, the Video Ontology, is a core reference ontology for video contents, while the second one, the MPEG-7 Ontology, was developed from scratch with formal grounding, is the ultimate OWL 2 ontology for all domains that utilize MPEG-7 terms.

6.1 The Video Ontology (VidOnt)

The feasibility and efficiency of automated reasoning relies on the accurate conceptualization and comprehensive description of relations between concepts, predicates, and individuals [144]. Despite the known benefits of multimedia reasoning in video classification, scene interpretation, and content-based video retrieval, most multimedia ontologies lack the constructors necessary for expressing axioms upon which complex inference tasks could be performed.[1] Also, the correlations between video properties, which could be exploited in video annotation and classification, have not been

---

[1] Sikos, L. F., Powers, D. M. W. (2015) Knowledge-driven video information retrieval with LOD: from semi-structured to structured video metadata. In: Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, pp. 35–37. Melbourne, VIC, Australia, 23 October 2015. New York: ACM. doi: 10.1145/2810133.2810141

captured by previous approaches. To address the reasoning limitations of previous multimedia ontologies, the *Video Ontology (VidOnt)*[2] has been introduced (see A.3). This defines fundamental videography concepts, video characteristics, and the relationship between them.[3] For the first time among video ontologies, VidOnt exploits all mathematical constructors of OWL 2, rather than a small subset of constructors available in the first version of OWL.

6.1.1 Rationale

Ontology-based reasoning is infeasible without expressive constructors, most of which are not implemented in previous multimedia ontologies. For example, the Visual Descriptor Ontology (VDO), which was published as an "ontology for multimedia reasoning" [145], actually has a rather limited reasoning potential, because it implements the constructors of the basic description logic $\mathcal{ALH}$, resulting in a very low description logic expressivity.

The video terms defined by multimedia vocabularies and ontologies are very much isolated, and their relations are not defined formally. Based on the explicit statements about the basic terminology of video production provided by previous multimedia ontologies, it is infeasible for reasoning services to infer sophisticated statements. None of the previous video ontologies features formal grounding with high enough expressivity to capture complex relations of videos. In fact, the video terminology defined in previous ontologies is limited to low-level multimedia descriptors (most of which do not convey information about the video content) and basic classification schemata for Hollywood movie classification.

To address these issues, VidOnt

1. provides a comprehensive coverage of videography concepts by
   a) defining important concepts not defined in any other video or video-related ontology,
   b) adding taxonomical structure to standard and de facto standard terminology,
   c) defining concept relations across standard vocabularies and ontologies;

---

[2] http://vidont.org

[3] Sikos, L. F. (2016) A novel approach to multimedia ontology engineering for automated reasoning over audiovisual LOD datasets. In: Nguyễn, N. T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) Intelligent information and database systems, pp. 3–12. 8th Asian Conference on Intelligent Information and Database Systems, Đà Nẵng, Vietnam, 14–16 March 2016. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-49381-6_1

2. defines domains, ranges, and datatypes for video properties, which inherently adds rich semantics about concept specificity, role applicability, and permissible values for roles, as opposed to vocabularies that declare terms with just some or none of these;
3. captures fundamental correlations and rules for reasoning over video metadata;
4. integrates the RCC8 Region Connection Calculus (Randell et al., 1992) and Allen's relations (1983) for spatiotemporal knowledge representation.[4]

As a result, VidOnt is suitable for the knowledge representation and annotation of video characteristics and capturing high-level semantics, upon which multimedia reasoning and Linked Open Data (LOD) interlinking can be performed. VidOnt acts as a mediator between upper ontologies, domain ontologies, and application ontologies used for the semantic enrichment of videos, automated scene interpretation, and video understanding.

6.1.2 Formal Grounding

VidOnt has been formally grounded in the $\mathcal{SROIQ}^{(\mathcal{D})}$ description logic (see Chapter 3), and implemented in OWL 2 by translating its axioms to the corresponding RDF triples. Some general examples for this transparent translation are demonstrated in Table 6.1.

Table 6.1 Examples for translating description logic axioms to OWL

DL Axiom	OWL Implementation
$a \approx b$	`:a owl:sameAs :b .`
$a \not\approx b$	`:a owl:differentFrom :b .`
$C \sqsubseteq D$	`:C rdfs:subClassOf :D .`
$C(a)$	`:a rdf:type :C .`
$R(a, b)$	`:a :R :b .`
$R^{-}(a, b)$	`:b :R :a .`

a, *b* individuals, *C*, *D* concepts, *R* role

---

[4] Sikos, L. F. (2018) VidOnt: a core reference ontology for reasoning over video scenes. *Invited paper*. Journal of Information and Telecommunication. doi: 10.1080/24751839.2018.1437696

In contrast to previous video ontologies, VidOnt defines not only terminological and assertional axioms, but also a role box. As you will see, constructors not exploited in previously released multimedia ontologies, in particular the role box axioms, significantly extend the application potential in video data integration, knowledge management, and ontology-based automated reasoning.

6.1.2.1 Terminological Knowledge

The concepts and roles of VidOnt have been defined in a hierarchy that incorporates de facto standard structured definitions. Terminological knowledge (TBox) is captured in VidOnt by defining the relationship of classes and properties as subclass axioms and subproperty axioms, and defining unions, intersections, and equivalent classes (see Definition 6.1).

Definition 6.1 (TBox). A TBox $\mathcal{T}$ is a finite collection of concept inclusion axioms and concept equivalence axioms.

For example, TBox axioms can express that keyframes are frames and that lectors are equivalent to narrators, as shown in Table 6.2.

Table 6.2 Expressing terminological knowledge with TBox axioms

DL Syntax	OWL Implementation
Keyframe $\sqsubseteq$ Frame	`video:Keyframe rdfs:subClassOf` `video:Frame .`

Lector $\equiv$ Narrator	`video:Lector`
	`owl:equivalentClass`
	`video:Narrator .`

Similar to other OWL ontologies, the terms of VidOnt can be deployed in fully-featured knowledge representations in any RDF serialization, such as Turtle or RDF/XML, or as lightweight markup annotations in RDFa, JSON-LD, or HTML5 Microdata [146].

VidOnt exploits the fact that conditional statements can be expressed as disjunction [147], as the truth table of "IF *p* THEN *q*" and "NOT *p* OR *q*" are identical (see Equation 6.1).

$$p \rightarrow q = (\sim p \vee q) \qquad \text{(Eq. 6.1)}$$

This approach allowed the formal definition of core video production knowledge without SWRL rules, such as by stating the following:

- if a disc release is a Blu-Ray release, the video mode is either 720p, 1080i, or 1080p;
- if a disc release is an Ultra HD Blu-ray disc release, it uses the 2160p video mode;
- if the video mode of a video is 2160p, the frame height is 2160 pixels or less (depending on the aspect ratio of the video).[5]

6.1.2.2 Assertional Knowledge

Individuals and their relationships are represented using ABox axioms (see Definition 6.2).

Definition 6.2 (ABox). An ABox $\mathcal{A}$ is a finite collection of concept assertion axioms of the form $C(a)$, role assertion axioms of the form $R(a,b)$, negated role assertions of the form $\neg R(a,b)$, equality statements of the form $a \approx b$, and inequality statements of the form $a \not\approx b$, where $a, b \in \mathrm{N}_I$ individual names, $C \in \mathrm{C}$ represents concept expressions, and $R \in \mathrm{R}$ represents role expressions, as demonstrated in Table 6.3.

Table 6.3 Asserting and describing individuals with ABox axioms

DL Axiom	OWL Implementation
ComputerAnimation(JOINT-REPLACEMENTVIDEO)	`<http://example.com/jointreplacement.mp4> a video:ComputerAnimation .`
depicts(JOINTREPLACEMENTVIDEO, JOINT)	`<http://example.com/jointreplacement.mp4> video:depicts ex:Joint .`

---

[5] 2160p videos with an aspect ratio wider than 1.78:1 might also be called 2160p videos even if their frame height is smaller than 2,160 pixels (such as a 4K Blu-Ray release with an aspect ratio of 2.40:1).

Table 6.3 (Cont'd) Asserting and describing individuals with ABox axioms

DL Axiom	OWL Implementation
$\mathrm{VIDEOCLIP} \approx \mathrm{CLIP}$	`Schema:Clip owl:sameAs po:Clip .`[6]

$\mathrm{JOINTREPLACEMENTVIDEO} \not\approx \mathrm{ANATOMYOFJOINTVIDEO}$	`<http://example.com/jointreplacement.mp4>`
	`owl:differentFrom`
	`<http://example.com/anatomyofjoint.mp4> .`

6.1.2.3 Role Box

Most multimedia ontologies described in the literature correspond to formalisms in which the knowledge base is defined as the combination of a TBox and an ABox.[7] Role box (RBox) axioms are supported in very expressive description logics only, such as $\mathcal{SROIQ}$ (see Definition 6.4), to collect all statements related to roles and the interdependencies between roles (see Definition 6.3).

Definition 6.3 (RBox). A rule box (RBox) $\mathcal{R}$ is a role hierarchy, a finite collection of generalized role inclusion axioms of the form $R \sqsubseteq S$, role equivalence axioms of the form $R \equiv S$, complex role inclusions of the form $R_1 \circ R_2 \sqsubseteq S$, and role disjointness declarations of the form $Dis(R, S)$, where R and S represent roles, and transitivity axioms of the form $R^+ \sqsubseteq R$, where R^+ is a set of transitive roles.

Definition 6.4 ($\mathcal{SROIQ}$ Knowledge Base). A $\mathcal{SROIQ}$ knowledge base $\mathcal{KB}$ is a triple $\langle \mathcal{T}, \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{T}$ is a set of terminological axioms (TBox), $\mathcal{A}$ is a set of assertional axioms (ABox), and $\mathcal{R}$ is a role box (RBox).

Some examples for role box axioms are shown in Table 6.4.

---

[6] The po prefix abbreviates the namespace http://purl.org/ontology/po/

[7] Sikos, L. F. (2016) A novel approach to multimedia ontology engineering for automated reasoning over audiovisual LOD datasets. In: Nguyễn, N. T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) Intelligent information and database systems, pp. 3–12. 8th Asian Conference on Intelligent Information and Database Systems, Đà Nẵng, Vietnam, 14–16 March 2016. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-49381-6_1

Table 6.4 Role box axioms

DL Syntax	OWL Implementation
covers ⊑ topologicallyRelated	`video:covers rdfs:subPropertyOf` `video:topologicallyRelated .`
Disjoint(left, right)	`video:left owl:propertyDisjointWith` `vidont:right .`
basedOn ∘ basedOn ⊑ basedOn	`video:basedOn a owl:TransitiveProperty .`

6.1.2.4 Increasing Expressivity

While $\mathcal{SROIQ}^{(\mathcal{D})}$, the description logic behind OWL 2 DL, is very expressive, it is inadequate for the spatiotemporal annotation and formal description of video events. While some OWL 2 axioms correspond to rules, such as class inclusion and property inclusion, some classes can be decomposed as rules, and property chain axioms provide rule-like axioms, there are statements that cannot be expressed in OWL 2. For example, a rule head with two variables cannot be represented as a subclass axiom, or a rule body that contains a class expression cannot be described by a subproperty axiom. To add additional expressivity to OWL 2 DL, ontologies can be extended with rule-based formalisms (see Definition 6.5), such as SWRL rules.

Definition 6.5 (Rule). A rule R is given as $H \leftarrow B_1, \ldots, B_n$ $(n \geqslant 0)$, where $H, B_1, \ldots, B_n$ are atoms, H is called the head (conclusion or consequent) and $B_1, \ldots, B_n$ the body (premise or antecedent).

SWRL rules, however, might break the decidability of an ontology. For this reason, VidOnt employs *DL-safe rules* [148] wherever possible (see Definition 6.6).

Definition 6.6 (DL-Safe Rule). Let $\mathcal{KB}$ be a $\mathcal{SROIQ}^{(\mathcal{D})}$ knowledge base, and let N_P be a set of predicate symbols such that $\mathsf{N}_C \cup \mathsf{N}_{R_a} \cup \mathsf{N}_{R_c} \subseteq \mathsf{N}_P$. A DL-atom is an atom of the form $A(s)$, where $A \in \mathsf{N}_C$, or of the form $R(s, t)$, where $R \in \mathsf{N}_{R_a} \cup \mathsf{N}_{R_c}$. A rule R is called DL-safe if each variable in R occurs in a non-DL atom in the rule body.

In VidOnt, SWRL rules are used only to formalize complex video events, which can be used for example to recognize scenes of a particular medical procedure for automated classification.

6.1.3 Evaluation

VidOnt was validated with industry-leading reasoners, namely HermiT[8] and FaCT++,[9] to ensure that it is meaningful, correct, minimally redundant, and richly axiomatized. The implementation potential of the ontology has been evaluated through knowledge representation and recognition of video scenes of 100 videos, including video scenes and video clips from well-established video datasets, such as the 2016 TRECVID Multimedia Event Detection dataset,[10] medical animations, feature films, including movie clips from Hollywood2,[11] randomly selected YouTube videos,[12] and camera recordings. The evaluation focused on the intended and actually represented semantics to highlight the strengths of video annotations using VidOnt, and identified limitations of general DL-based and rule-based formalisms in video scene representation.

6.1.3.1 Case Study 1: Professional Video Production

VidOnt provides a comprehensive set of properties for describing the technical characteristics of both uncut and edited video files, most of which are not available in any other multimedia ontology. As an example, consider a report with a doctor who recently patented a new medical procedure, recorded with a Canon XC15 camcorder in 4K UHD (QFHD) with a frame rate of 24fps (cinema mode), a constant video bitrate of 205Mbps, using the YCbCr color space and 4:2:2 chroma subsampling, the H.264/MPEG-4 AVC video codec and 48kHz LPCM audio in an MXF container, which has been imported to Adobe Premier Pro for post-production (see Listing 6.1).

---

[8] http://www.hermit-reasoner.com

[9] http://owl.man.ac.uk/factplusplus/

[10] https://www.nist.gov/itl/iad/mig/med-2016-evaluation

[11] http://www.di.ens.fr/~laptev/actions/hollywood2/

[12] https://www.youtube.com

Listing 6.1 Annotating advanced video file characteristics

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix schema: <http://schema.org/> .
@prefix video: <http://purl.org/ontology/video/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://example.com/patentreport/> a video:Video ;
dc:title "Patent Report" ;
video:directedBy "John Smith" ;
video:producedBy "Jane Doe" ;
video:runningTime "12"^^xsd:positiveInteger ;
schema:Country dbpedia:Australia ;
schema:inLanguage "en-Au"^^xsd:language ;
video:recordedUsing "Canon XC15" ;
video:resolution "QFHD" ;
video:videoBitrate "205"^^xsd:positiveInteger ;
video:videoBitrateType dbpedia:Constant_bitrate ;
video:colorSpace dbpedia:YCbCr ;
video:chromaSubsampling dbpedia:4:2:2 ;
video:colorTemperature "5700"^^xsd:positiveInteger ;
video:videoCompression dbpedia:H.264/MPEG-4_AVC ;
video:frameRate "24"^^xsd:positiveInteger ;
video:TVSystem dbpedia:PAL ;
video:refreshRate "50"^^xsd:positiveInteger ;
video:is3D "no"^^xsd:string ;
video:videoFileFormat dbpedia:Material_Exchange_Format ;
video:videoEditingSoftware dbpedia:Adobe_Premiere_Pro ;
video:audioSamplingRate "48"^^xsd:float .
```

Note that de facto standard properties, such as the country and language of release defined by Schema.org, are reused as per Semantic Web best practice, to benefit from well-established definitions and maximize interoperability [149].

Notice that several other properties could be added to the previous description, which are, however, correlated. The formal description of such concept and role relationships collected in VidOnt makes it possible to infer new statements automatically from the previous knowledge representation. For example, standard QFHD videos have an aspect ratio of 1.78:1, use 2160p video mode, a frame width of 3,840 pixels, a frame height of 2,160 pixels, and progressive scanning (see Listing 6.2).

Listing 6.2 Statements inferred using definitions of correlated properties

```
<http://example.com/patentreport/>
video:aspectRatio "1.78" ;
video:videoMode dbpedia:2160p ;
video:scanning "progressive" ;
video:frameWidth "3840"^^xsd:positiveInteger ;
video:frameHeight "2160"^^xsd:positiveInteger .
```

VidOnt also considers the differences between video camera recordings and disc releases, thereby preventing false deductions via reasoning. For example, standard 4K Blu-ray releases exclusively employ the H.265/MPEG-H Part 2 (HEVC) encoding, while 4K video cameras still often use H.264/MPEG-4 AVC instead, meaning that the video codec of 4K videos cannot be assumed to be HEVC. Similarly, disc releases have a standardized maximum bitrate, which is often exceeded by the bitrate offered by video cameras. Therefore, the video bitrate cannot be deduced from the video mode.

6.1.3.2 Case Study 2: Broadcasting

There are a limited number of multimedia ontologies for broadcasting annotations, such as the BBC Programmes Ontology,[13] EBU CCDM,[14] and TV Anytime,[15] none of which is suitable for advanced

---

[13] http://purl.org/ontology/po/

reasoning over broadcasted video content. For this reason, VidOnt provides a comprehensive coverage for the specifications of broadcasting, such as the broadcast format, the transmit power, the color zone, the transmission standard, and the delivery format. This enables the annotation of terrestrial, cable, or satellite television broadcasts (telecasts), and webcasts.

As an example, consider a TV advertisement broadcasted in an area with the best reception conditions (CEA yellow zone) using the Digital Video Broadcasting Second Generation Terrestrial (DVB-T2) standard with a 4/5 Forward Error Correction (FEC) for controlling errors in the data transmission. The broadcasting is happening in the UHF TV band. The method of data encoding is orthogonal frequency division multiplex (OFDM), which uses multiple carrier frequencies to represent the HDTV channel. The commercial advertiser is identified using the industry standard Ad-ID identifier (see Listing 6.3).

Listing 6.3 Broadcasting annotation

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix schema: <http://schema.org/> .
@prefix video: <http://purl.org/ontology/video/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://example.com/telecastexample> a video:telecastVideo ;
video:TVSystem dbpedia:PAL ;
video:is3D "no"^^xsd:string ;
video:colorZone "yellow" ;
video:transmissionStandard dbpedia:DVB-T2 ;
video:FEC "4/5" ;
video:TVband dbpedia:Ultra_high_frequency ;
video:TVchannel "42.1" ;
video:TVnetwork "XYZ" ;
schema:City dbpedia:Adelaide ;
schema:State dbpedia:South_Australia ;
```

---

[14] https://www.ebu.ch/metadata/ontologies/ebuccdm/20120915/CCDM_Core.owl

[15] http://rhizomik.net/ontologies/2005/03/TVAnytimeFormat.owl

Listing 6.3 (Cont'd) Broadcasting annotation

```
schema:Country dbpedia:Australia ;
schema:inLanguage "en-us"^^xsd:language ;
video:liveDate "2017-05-20"^^xsd:date ;
video:frequencyAssignment "43"^^xsd:float ;
video:modulation dbpedia:OFDM ;
video:Ad-ID  "SWCD5007000H" .
```

Based on the specification of the TV system, the refresh rate and the framerate can be inferred, while the DVB-T2 transmission standard clearly indicates that the TV station broadcasting the advertisement is digital (see Listing 6.4).

Listing 6.4 Inferred broadcasting properties

```
<http://example.com/telecastexample>
video:refreshRate "50"^^xsd:positiveInteger ;
video:frameRate "25"^^xsd:float ;
video:TVstationType "DTV" .
```

6.1.4 Importance

The vocabulary of VidOnt has been aligned with standards such as MPEG-7, EBU CCDM, and Dublin Core, and defined with a novel taxonomic structure. To push the expressivity boundaries of its logical underpinning, VidOnt employs rule-based formalisms.

VidOnt provides a comprehensive set of machine-interpretable concept definitions and axioms for mainstream and cutting-edge video technologies and standards, such as the 4K UHD video mode and the H.265/MPEG-H HEVC codec. This enables the machine-interpretability of the correlations between video mode and scanning, standard disc releases and video resolution, aspect ratio and maximum frame height, and so on.

The efficiency and practical applicability of the presented ontology have been evaluated through case studies with promising results. However, the definition of complex video events in the form of rule-based formalisms performed quite differently in constrained and unconstrained videos.

Beyond straightforward implementations in video scene representation, indexing, and retrieval, VidOnt can also be used to improve existing classifications by identifying video data incorrectly entered into a video database, and to avoid incorrect data entries in the future. In fact, using the fundamental statements of the ontology provides the option to add a dropdown list of all the possible options for entering video characteristics into video databases.

6.2 The Ultimate MPEG-7 Ontology

As mentioned earlier in Chapter 4, there is a large number of MPEG-7 ontologies available in the literature. Despite these OWL mappings of the MPEG-7 vocabulary and MPEG-7-aligned ontologies, the entire MPEG-7 vocabulary is covered by very few ontologies, none of which have been formally grounded. As a result, they have modeling issues, such as incorrect concept hierarchy and conceptual ambiguity, and fail to capture complex relations between concepts, roles, and individuals.

6.2.1 Problem Statement

Despite several attempts to map MPEG-7 terms to OWL, all previously released mappings are limited [150]. They either inherited issues directly from the original MPEG-7 vocabulary or failed to apply ontology engineering best practices. What makes developing MPEG-7 ontologies really challenging is the structural complexity of the standard: the MPEG-7 XML schemas define 1,976 terms in the form of 1,182 elements, 417 attributes, and 377 complex datatypes, many of which do not correspond directly to classes and properties. In addition, the standard is not freely available. Its documentation is not detailed enough to apply a top-down ontology engineering approach, while the term definitions in the XSD files do not put the concepts and roles into context, thereby preventing a bottom-up development approach.

The most common problems of previous MPEG-7 ontologies are detailed in the following sections.

6.2.1.1 Poor Specification

The initial step of ontology engineering is specifying the intended applications, however, many of the previous MPEG-7 ontologies do not fall into any known ontology category, meaning that their knowledge domain and scope have not been defined clearly. While the knowledge domain of MPEG-7 is "multimedia," the terminology of the standard cannot be captured as a domain ontology, because the multimedia contents to describe with MPEG-7 terms span multiple knowledge domains. Considering the MPEG-7 terminology a knowledge domain, like most previous MPEG-7 ontology authors did, leads to poor conceptualization and an ontology granularity inadequate for capturing the semantics of multimedia resource properties. MPEG-7 covers a wide range of concepts and properties related to image, audio, video, and 3D model creation, editing, manipulation, segmentation, and use, making it necessary to create an upper ontology for the MPEG-7 standard.

The computer programs that generate and use many of the low-level descriptors defined in MPEG-7 can process the corresponding data even if they are written in XML (using terms from the vocabulary of the original MPEG-7 XSD files). Therefore, while the XML-OWL translation is feasible, such as via XSLT, the OWL mapping of most low-level descriptors is not particularly useful, unless their definition goes well beyond a simple axiom that specifies whether they are classes or properties, and if they are complemented by rich semantics.

6.2.1.2 Inadequate Knowledge Acquisition

The majority of MPEG-7 ontologies had issues with knowledge acquisition, resulting in conceptualizations that capture a representation of the MPEG-7 specification rather than actual multimedia knowledge domains. The taxonomical structure of these ontologies failed to put MPEG-7 terms into context, which prevented practitioners from deploying them globally. Even COMM, one of the most expressive OWL mappings of MPEG-7, defines the document of the MPEG-7 specification itself, and not those multimedia concepts and properties that can be used for describing multimedia objects. This is clearly indicated by the concepts of the ontology, such as the *color index* parameter of the *Dominant Color Descriptor*, which is defined as a class in COMM, making it impossible to capture the corresponding property values for multimedia resources. In addition, the concept hierarchy is based on sets of terms from the MPEG-7 specification, so that the aforementioned `color-index`, defined as a concept in COMM, is declared to be a subclass of `unsigned-12-vector-dim-3`, which represents the class of vectors with three numbers of the datatype `mpeg7:unsigned12`, i.e., vectors of three 12-bit nonnegative integers. From the modeling point of view, these vectors are obviously not equivalent to any concept that could be used for expressing rich semantics for multimedia resources. The `unsigned-12-vector-dim-3` class is defined as a subclass of `vector`, which is a subclass of the `abstract-region` class, a subclass of `region`. The `region` class is a subclass of `abstract`, which is defined as a subclass of `particular`. This ontology design issue can be best demonstrated by formalizing these classes in a description logic concept hierarchy, as shown in Listing 6.5.

Listing 6.5 Knowledge acquisition issue demonstrated by formal grounding[16]

color-index ⊑ unsigned-12-vector-dim-3
unsigned-12-vector-dim-3 ⊑ integer-vector
integer-vector ⊑ vector
vector ⊑ abstract-region
abstract-region ⊑ region
region ⊑ abstract
abstract ⊑ particular

Beyond this taxonomical hierarchy issue, COMM does not define superclasses for core multimedia content types, such as image, audio, and video, i.e., the knowledge domain covered by COMM is not a collection of multimedia subdomains, as it should be, but a set of multimedia terms derived from the MPEG-7 specification. On top of this, the terms are defined using multiple namespaces across multiple files, mixed with custom terms not aligned with the MPEG-7 standard.

---

[16] Sikos, L. F. (2017) Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5

Other MPEG-7 ontologies have similar design issues, the root cause of which is the reliance on the MPEG-7 XSD files for defining the concepts and creating the concept hierarchy. In fact, some MPEG-7 classes and properties are not practical to use—in their original form—in formal multimedia descriptions on the Semantic Web. Therefore, a fully standard-compliant conceptualization not solely based on the MPEG-7 XSD files would be better than the previous approaches for the knowledge acquisition when creating an MPEG-7 ontology.

6.2.1.3 Lack of Formal Grounding

The majority of design issues of previous MPEG-7 ontologies could have been prevented by formal grounding. Since all these ontologies have been created automatically by mapping terms from XSD files to OWL, and/or via visual concept hierarchy editing, none of them has been properly grounded in description logics. The lack of formal grounding is evidenced by serious design issues, such as the following:

- Incorrect concept and datatype definitions: many MPEG-7 OWL mappings define concepts as object properties, and datatype properties and datatypes as concepts;
- Incomplete definitions: many MPEG-7 OWL mappings define object properties without a domain or range, datatype properties without the range of permissible values, and classes without a taxonomical structure;
- Conceptual ambiguity: semantically equivalent descriptors representing the same information are defined by multiple classes and entities, as a result of incorrect modeling;[17]
- Semantic interoperability issues: lack of explicit axiomatization of the intended semantics, directly inherited from the MPEG-7 specification, which does not define formal semantics for the descriptors;
- Syntactic interoperability issues: unaddressed syntactic variability;
- Missing concept hierarchy: many MPEG-7 concepts can be put into context by defining them as subclasses/superclasses, which is missing from the MPEG-7 XSD files and inherently from all OWL mappings based on them;

---

[17] Not to be confused with purposefully denoting different entity types with the same identifier, called punning, which is used for metamodeling.

- Unclear computational properties: although DL expressivity can be automatically calculated (as seen in the ontology editor Protégé when opening an RDFS or OWL file), upon which computational properties can be determined, reasoning complexity and decidability are better determined via formal grounding.

6.2.1.4 Low DL Expressivity and Poor Semantics

A common problem with previously released MPEG-7 ontologies is that they underutilize the mathematical constructors available in the implementation language, OWL. In other words, they use only a small subset of the allowed constructors, making them unable to capture complex relationships and sophisticated multimedia semantics. If these ontologies had been formally grounded in a description logic, this would have been clearly indicated by a low description logic expressivity.

Despite that semistructured definitions are serialized in a structured format, an XSD to OWL mapping does not add rich semantics to concepts. This is particularly true in the case of the MPEG-7 vocabulary, which features low-level descriptors that do not directly correspond to the meaning (semantics) of multimedia contents (as mentioned earlier in Chapter 1, for example, that based on the automatically extracted dominant colors, what is in a picture cannot be inferred).

Obviously, a list of terms without a comprehensive formal description is insufficient for multimedia reasoning. While finding the right trade-off between expressivity and computational complexity is crucial in ontology engineering, extremely low expressivity might prevent reasoning altogether. COMM uses $\mathcal{SHOIN}^{(\mathcal{D})}$, the most expressive DL at the time of its release (in OWL DL), however, the current standard, OWL 2 DL, supports far more constructors, such as complex role inclusion axioms, reflexive, asymmetric, irreflexive, and disjoint roles, universal role, self-constructs, negated role assertions, and qualified number restrictions. Also, using constructors in incorrect concept and role definitions results in poor semantics, regardless of the high expressivity. All the other MPEG-7 mappings are far less expressive, as seen in Table 6.5, which compares the expressivity of all RDFS and OWL ontologies that are based on, or aligned with, MPEG-7, and are available online at the time of writing.

Table 6.5 DL expressivity of MPEG-7 ontologies

Ontology	Language	DL Expressivity
COMM	OWL[18]	$\mathcal{SHOIN}^{(\mathcal{D})}$
Rhizomik	OWL[19]	$\mathcal{ALUN}^{(\mathcal{D})}$
SWIntO	RDFS[20]	$\mathcal{ALH}^{(\mathcal{D})}$
VDO	OWL[21]	$\mathcal{ALH}$

The table shows that the description logic expressivity of previous MPEG-7 mappings is quite low, except for COMM. Due to various design issues, however, only a few terms can be used from COMM to represent multimedia resources and perform reasoning-based scene interpretation. In fact, the relatively high expressivity of COMM comes from the definition of relationships between classes, which should be properties, because in their current form they cannot capture the corresponding property values. While one might argue that defining additional properties (e.g., `hasDominantColor`) would make it possible to implement the concepts and roles with the current ontology granularity, doing so would diverge from the standard MPEG-7 XML schemas and would almost double the size of the already large ontology file.

6.2.1.5 Inheriting and Introducing Issues via XSD-OWL Translation

Most OWL mappings of MPEG-7 inherited issues from the XSD files of the standard. The Rhizomik Ontology,[22] for example, was created using the XML Semantics Reuse methodology, which combined an XSL-based XML Schema to OWL mapping (XSD2OWL)[23] with a transparent mapping from XML to RDF (XML2RDF).[24] The "transparent" XSD-OWL mapping was based on the translation rules shown in Table 6.6.

---

[18] http://multimedia.semanticweb.org/COMM/visual.owl

[19] http://rhizomik.net/ontologies/2005/03/Mpeg7-2001.owl

[20] http://smartweb.dfki.de/ontology/swinto0.3.1.rdfs

[21] https://raw.githubusercontent.com/gatemezing/MMOntologies/master/ACEMEDIA/acemedia-visual-descriptor-ontology-v09.rdfs.owl

[22] http://rhizomik.net/ontologies/2017/05/Mpeg7-2001.owl

[23] http://rhizomik.net/html/redefer/#XSD2OWL

[24] http://rhizomik.net/html/redefer/#XML2RDF

Table 6.6 XSD-OWL translation rules implemented in XSD2OWL [151]

XSD	OWL

`element|attribute`	`rdf:Property`
	`owl:DatatypeProperty`
	`owl:ObjectProperty`

`element@substitutionGroup`	`rdfs:subPropertyOf`
`element@type`	`rdfs:range`
`complexType|group|attributeGroup`	`owl:Class`
`complexType//element`	`owl:Restriction`
`extension@base|restriction@base`	`rdfs:subClassOf`
`@maxOccurs` `@minOccurs`	`owl:maxCardinality` `owl:minCardinality`
`sequence` `choice`	`owl:intersectionOf` `owl:unionOf`

To demonstrate the output of this translation, first take a closer look at the definition of the color space datatype from the XSD file of MPEG-7 (see Listing 6.6).

Listing 6.6 Definition of the `ColorSpace` datatype in the MPEG-7 XSD file

```
<complexType name="ColorSpaceType" final="#all">
  <sequence>
    <element name="ColorTransMat" minOccurs="0">
      <simpleType>
        <restriction>
          <simpleType>
            <list itemType="mpeg7:unsigned16"/>
          </simpleType>
          <length value="9"/>
```

Listing 6.6 (Cont'd) Definition of the `ColorSpace` datatype in the MPEG-7 XSD file

```
        </restriction>
      </simpleType>
    </element>
  </sequence>
  <attribute name="colorReferenceFlag" type="boolean" use="optional" default="false"/>
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="string">
        <enumeration value="RGB"/>
        <enumeration value="YCbCr"/>
        <enumeration value="HSV"/>
        <enumeration value="HMMD"/>
        <enumeration value="LinearMatrix"/>
        <enumeration value="Monochrome"/>
      </restriction>
    </simpleType>
  </attribute>
</complexType>
```

This definition has been translated to OWL as shown in Listing 6.7.

Listing 6.7 OWL translation of Listing 6.6 in the Rhizomik Ontology

```
<owl:Class rdf:about="&mpeg7;ColorSpaceType">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="&mpeg7;ColorTransMat"/>
          <owl:allValuesFrom rdf:resource="&xsd;string"/>
        </owl:Restriction>
```

Listing 6.7 (Cont'd) OWL translation of Listing 6.9 in the Rhizomik Ontology

```
        <owl:Restriction>
          <owl:onProperty rdf:resource="&mpeg7;ColorTransMat"/>
            <owl:maxCardinality rdf:datatype=
             "&xsd;nonNegativeInteger">1</owl:maxCardinality>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&mpeg7;colorReferenceFlag"/>
      <owl:allValuesFrom rdf:resource=
       "&mpeg7;http://www.w3.org/2001/XMLSchema#boolean"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&mpeg7;type"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
…
<owl:DatatypeProperty rdf:about="&mpeg7;ColorTransMat"/>
…
<owl:DatatypeProperty rdf:about="&mpeg7;colorReferenceFlag"/>
```

This mapping not only did not give additional semantics to what was defined in the original XSD file (as it should have done, considering the higher expressivity of the Web Ontology Language), but it actually lost information from the original definition. The length restriction of exactly 9 characters has been lost during the translation, along with the color spaces supported by MPEG-7 defined in the XSD file, namely, RGB, YCbCr, HSV, HMMD, LinearMatrix, and Monochrome. These are not defined anywhere in the OWL file of the Rhizomik Ontology, and the permissible values of the type property have been restricted to `xsd:string`, not the actual string literals. The datatype properties defined in the Rhizomik Ontology do not have associated datatypes that would define their permissible values, because the rules listed in Table 6.6 translated the MPEG-7 datatypes to OWL classes.

The naming convention inherited from the standard is not ideal for the formal representation of multimedia resources, because many of the concept and property names cannot be interpreted without context. From the semantics point of view, a property such as `hasColorSpace` with permissible values defined using the previous datatype would be more useful, as it would serve as an easy-to-use predicate in RDF triples when describing a multimedia resource.[25]

6.2.1.6 No Stable Namespace and Inadequate Documentation

Most of the previous MPEG-7 ontologies never had a dedicated website or a permanent namespace URI. The ones that had a webpage buried deeply in a research group's folder structure disappeared over time, invalidating all RDF statements on the Internet that implemented their namespace URI and prefix. Out of the nine well-cited MPEG-7 ontologies, only four are available today (COMM, Rhizomik, SWIntO, and VDO), and five are not (Harmony, MCO, MDO, SCDO, and DS-MIRF).

Ideally, an ontology website is set up using content negotiation, thereby serving the ontology file for semantic agents, and a dedicated webpage about a requested MPEG-7 term for web browsers. Also, an ontology URI should be a permanent URI which is kept short for practicality reasons. This is quite the opposite of what has been done for the previous MPEG-7 ontologies. In fact, most of them used symbolic, rather than physical, URIs. By providing a compact and permanent namespace URI, complemented by well-written and frequently updated documentation, an ontology becomes much more appealing to practitioners. This way, implementers can access a general description,

---

[25] However, adding new or altering original MPEG-7 terms would break standard-alignment, and defining new terms would significantly increase the ontological commitment.

important technical characteristics, contact persons, information about the anticipated audience and applications, and sample code for implementation.

6.2.1.7 Inadequate Evaluation

Previous MPEG-7 ontologies contained errors such as contradictions and inconsistencies. The (un)satisfiability of these ontologies should have been checked using reasoners such as HermiT, FaCT++, Pellet,[26] or Racer.[27] These tools are based on highly optimized tableau algorithms, which attempt to construct an abstraction of a tree-like model $\mathcal{I}$ (a completion graph chosen in a particular iteration) that satisfies all axioms of the ontology. Also, testing the ontologies in real-life implementations could have highlighted issues that have not been identified in the earlier stages of ontology development.

6.2.2 A Novel Approach to Capture MPEG-7 Semantics

Because the transparent XML-RDF/XSD-OWL mappings did not give satisfactory results for previous MPEG-7 ontologies, a different approach has been used to create a correctly modeled MPEG-7 ontology. This approach set ontology engineering aims by analyzing and addressing the issues of previous MPEG-7 ontologies as follows:

- The ontology cannot rely on the MPEG-7 XSD files alone. One of the main limitations of previous approaches was the reliance on transparent mapping, thereby utilizing a very limited subset of constructors available in OWL 2. In other words, they implemented the semistructured MPEG-7 vocabulary in a structured ontology language, however, they did not define additional semantics to what is described in the MPEG-7 XSD files. Those OWL ontologies that define terms in a taxonomical structure are not richer semantically than the corresponding XSD file that defines the same taxonomy.

---

[26] https://github.com/stardog-union/pellet

[27] http://www.ifis.uni-luebeck.de/~moeller/racer/

- The ontology should not be created by directly implementing it in OWL. Instead, to ensure model correctness, consistency, and to provide precise reasoning complexity measures, the ontology first has to be formally grounded in a description logic formalism.
- All relations, simple or complex, between concepts and roles have to be captured using DL axioms and implemented as complex OWL constructors. By doing so, the ontology will be suitable for complex reasoning tasks (unlike previous MPEG-7 ontologies). This is a huge task and requires solid theoretical background and hands-on skills in image processing, audio and video authoring, and 3D modeling. Most authors of previous MPEG-7 ontologies did not define relations between concepts and roles just because they are not captured in the original XSD files, however, this is the very reason why previous MPEG-7 ontologies are very limited in multimedia reasoning: software agents simply cannot infer core multimedia component interdependencies based on a concept list or hierarchy alone.
- The concept hierarchy requires concepts beyond the ones defined by the standard in order to put MPEG-7 terms into context when exposed through a namespace, so that software agents can infer useful, explicit statements and calculate semantic relatedness through concept subsumption and typed links, find related concepts by traversing a subgraph of the LOD Cloud, and so on. These new concepts have to be annotated with provenance data so that they are clearly segregated from the standard MPEG-7 terminology.
- The ontology has to be exposed through a namespace URI via content negotiation so that by requesting a URI with any MPEG-7 term within this domain, browsers can display the documentation of the corresponding MPEG-7 term and semantic agents can retrieve the ontology file. This is a massive task but well worth the effort, considering that many previous ontologies have symbolic URIs only and very few of them have a stable namespace, often with an excessively long URI.

6.2.3 Ontology Engineering

To address the aforementioned issues, a novel OWL 2 ontology has been introduced, called the *MPEG-7 Ontology* (with a capital O), according to ontology engineering best practices, with formal grounding in description logics, and with full standards compliance (see A.1).

Because the aim of the proposed ontology is to cover the MPEG-7 vocabulary and formally describe the relations between concepts and roles, determining ontology granularity was straightforward. The ontology was designed to be an upper ontology, which can be used in combination with domain ontologies for the formal description of, and reasoning over, multimedia contents. Anticipated application areas include multimedia analysis, content-based multimedia retrieval, and intelligent multimedia applications, such as video recommender systems and hypervideo applications.

Instead of a bottom-up or top-down approach used by previous MPEG-7 ontology developers, the presented ontology has been created using a hybrid approach. For knowledge acquisition, not only the XSD files have been used (like many previous MPEG-7 ontology developers did), but also the ISO standard itself, along with the principal concept list of MPEG-7.[28]

While the MPEG-7 vocabulary can be extracted from the XSD files, determining which terms correspond to concepts and which ones to roles can be challenging. This is the crucial step of ontology engineering all the previous MPEG-7 ontology developers struggled with for two reasons. First, all relied on the XSD files of MPEG-7, which do not explicitly state what is a class and what is a property. Moreover, they extensively use `xsd:complexType` to define MPEG-7 terms, which is an element that contains multiple XML elements and/or attributes by definition, but there is no restriction on its semantics.[29] Second, the model of the MPEG-7 XSD files is not complete, therefore the OWL mappings of these files are not complete either. Therefore, the term list of the proposed ontology has been constructed by analyzing the defined XML elements of the XSD files of MPEG-7 and matching them with concepts described in the standard. Based on this analysis and the assessment of term use, the terms have been conceptualized with formal grounding in description logic syntax.

This approach resulted in concept definitions for all the MPEG-7 concept terms, including the ones that have been incorrectly defined in all the previous MPEG-7 ontologies. The concept hierarchy has been created with practical considerations in mind, particularly where the XSD definitions gave no clues about the taxonomical structure.

---

[28] https://www.w3.org/2001/05/mpeg7/W3413_principal_concepts_list.doc

[29] See the OWL XML Schema at https://www.w3.org/2007/OWL/wiki/OWL_XML_Schema

Most role definitions have been derived from the elements and attributes of the MPEG-7 XSD files, however, some elements have been defined as concepts, because they do not hold any values but correspond to a collection of multimedia resource individuals. These have been incorrectly mapped to properties in all the previous MPEG-7 ontologies, and only the proposed ontology captures the semantics correctly, thereby enabling the use of these MPEG-7 concepts in RDF statements.

A compact URI has been selected to be the namespace URI of the proposed ontology, namely, `http://mpeg7.org/`. At the time of writing, this is the only namespace URI that exposes the entire MPEG-7 vocabulary, which is complemented by a dedicated and frequently updated webpage for each MPEG-7 term. This documentation aims at promoting the correct term usage by providing descriptions for MPEG concepts and roles, along with code samples to help practitioners with implementation.

6.2.4 Evaluation

The proposed ontology has been evaluated for conceptual modeling, correctness, consistency, code optimality, standard compliance, and implementation potential, and has been compared to some of the most well-known and still available OWL mappings of MPEG-7.

6.2.4.1 Checking Consistency, Modeling Correctness, and Code Optimality

The integrity of the MPEG-7 Ontology was verified via reasoning in HermiT and FaCT++. It has also been assessed in terms of compliance with the five core ontology engineering principles of Gruber, the researcher who introduced ontologies in the context of artificial intelligence [152]:

- Clarity: the intended semantics of the MPEG-7 terms have been defined with machine-interpretable constraints for semantic agents and concise descriptions with contextual information for humans. Although the atomic concepts and roles defined in the ontology characterize subsets of the multimedia domain, their definition is independent from the modeling context.

- Coherence: the integrity of the MPEG-7 Ontology has been checked with industry-leading reasoners, namely, HermiT, FaCT++, and Pellet, thereby guaranteeing that RDF statements contradicting MPEG-7 definitions will never be inferred from the axioms of the ontology.
- Minimal encoding bias: in the MPEG-7 Ontology, the semantics of multimedia concepts have been captured at the knowledge level independent from any symbol-level encoding. The ontology engineering has been conducted using open standards, including RDF, RDFS, OWL, and Turtle, rather than proprietary specifications, serializations, or file formats.
- Minimal ontological commitment: the proposed ontology has been designed to be as lightweight as possible, while providing full coverage of the MPEG-7 terminology.
- Extendibility: the axioms of the ontology can be refined without changing the core concept or role hierarchy. The proposed ontology has been created according to the MPEG-7 specification, and can be used in combination with domain ontologies and interlinked with LOD datasets.

6.2.4.2 Testing Implementation Potential: a Case Study

The implementation potential of the proposed ontology has been evaluated in a case study, which involved annotation of video frames and spatiotemporal video segments. The descriptors and their properties were expressed in the form of RDF triples. Owing to the popularity, simplicity, and compactness of the Turtle serialization, along with the fact that it makes RDF triples easy to read, Turtle has been selected to demonstrate this implementation, rather than the more verbose RDF/XML or any of the less common RDF serializations (see Listing 6.8).

Listing 6.8 Video frame annotation with MPEG-7 Ontology terms

```
@prefix mpeg7: <http://mpeg7.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://example.com/video.mp4#t=smpte:01:14:40:03> a mpeg7:Frame ;
mpeg7:temporalDecomposition <http://example.com/video.mp4> ;
mpeg7:width "3840"^^xsd:positiveInteger ;
mpeg7:height "1620"^^xsd:positiveInteger ;
```

Listing 6.8 (Cont'd) Video frame annotation with MPEG-7 Ontology terms

```
mpeg7:DominantColor "233 14 52 0.65 1"^^mpeg7:DominantColorType ;
mpeg7:YACCoeff5 "16 12 15 12 17" ;
mpeg7:CbACCoeff2 "22 17" ;
mpeg7:CrACCoeff2 "16 14" ;
mpeg7:YDCCoeff "50"^^mpeg7:unsigned6 ;
mpeg7:CbDCCoeff "34"^^mpeg7:unsigned6 ;
mpeg7:CrDCCoeff "30"^^mpeg7:unsigned6 ;
mpeg7:ColorTemperature "warm"^^mpeg7:ColorTemperatureType ;
mpeg7:BinCounts "5 2 1 3 3 4 3 3 4 3 6 0 5 1 3 4 3 2 3 1 2 3 1 4 4 3 0 4 5 6 2 1 5 5 1
2 4 1 7 7 3 2 6 4 2 2 5 7 1 6 6 7 3 7 4 1 1 1 2 0 4 2 1 2 5 4 4 5 3 3 2 2 0 2 3 5 1 3 5
3" .
```

In Turtle, resource URIs are delimited by less than and greater than symbols. Each set of statements about a single subject has a trailing full stop. RDF triples sharing the same subject are abbreviated by stating the subject once, followed by semicolon-separated predicate-object pairs. If there are multiple objects declared for the same subject-predicate pair, they are separated by commas. Datatypes are declared by appending `^^` and a datatype identifier. URIs are abbreviated using the namespace mechanism, which concatenates the prefix URI and the ontology term. The previous example declares three namespaces using `@prefix`: the proposed namespace, the RDF namespace for expressing "is a" relationships, and the XML Schema namespace for standard datatypes.

The first part of the code describes the third frame of the `video.mp4` video file at 1:14:40, which is declared to be a 3,840×1,620-pixel frame using the MPEG-7 terminology through the namespace of the proposed ontology. This corresponds to a frame of a 4K Ultra HD Blu-ray video with an aspect ratio of 2.37:1. Such representation of video frames can be used, for example, for multimedia retrieval using the query language SPARQL, such as when searching for screenshots of 4K movies in a video repository (see Listing 6.9).

Listing 6.9 A SPARQL example for multimedia retrieval using terms of the proposed ontology

```
PREFIX mpeg7: <http://mpeg7.org/>
SELECT ?title
WHERE {
      ?screenshot a mpeg7:Frame ;
      mpeg7:Title ?title ;
      mpeg7:width ?width .
      FILTER (?width = 3840) }
ORDER BY ASC(?title)
```

The *dominant color descriptor (DCD)* of the frame is defined with the following values of the `DominantColorType` datatype, as per MPEG-7 specifications: the first three numbers (`233`, `14`, and `52`) represent the vector of the corresponding color space component values, the fourth value (`0.65`) is the ratio of the number of pixels corresponding to the color and the total number of pixels in the frame normalized to a value within the range [0, 1], and the last value (`1`) represents the spatial coherency, i.e., the overall spatial homogeneity of the dominant color in the frame calculated as the linear combination of the individual spatial coherence values weighted by the corresponding percentage non-uniformly quantized to 5 bits.[30] Note that the MPEG-7 standard allows a maximum of eight dominant colors for a single frame, but typically 3–4 colors provide a good characterization.

The *color layout descriptor (CLD)*, which captures the spatial distribution of colors in the frame, is created by partitioning the frame into 8×8 blocks, deriving the average color of each block, and performing discrete cosine transform (DCT) to the luminance (Y), the blue chrominance (Cb), and the red chrominance (Cr). This yields to three sets of 64 signal amplitudes, expressed by the DCT coefficients DCTY, DCTCb, and DCTCr. The DCT coefficients include DC coefficients and AC coefficients,[31] i.e., coefficients that have zero frequency in both dimensions (the mean value of the waveform), and coefficients that have non-zero frequencies. These DCT coefficients are then quantized and zig-zag scanned.

---

[30] 31 is the highest and 1 is the lowest confidence, and 0 means that spatial coherence is not computed.

[31] The names AC and DC are derived from the historical use of the DCT for analysing electrical currents, where AC represents alternating current and DC means direct current.

In our example, the first three coefficient properties are declared by multiple 5-bit property values of the type `unsigned5`, i.e., nonnegative integers in the range [0, 31]. Since the length of the coefficient values is indicated by a trailing number in MPEG-7, YACCoeff5 is declared by five `unsigned5` values, while `CbACCoeff2` and `CrACCoeff2` are declared by two `unsigned5` values. The 6-bit values of the last three coefficients, `YDCCoeff`, `CbDCCoeff`, and `CrDCCoeff`, are of the `unsigned6` type, i.e., nonnegative integers in the range [0, 63].

The color temperature descriptor (CTD) of MPEG-7 describes the perceptual temperature feeling of an image, which can be declared using the value `hot`, `warm`, `moderate`, or `cool`, as defined by the `mpeg7:ColorTemperatureType` datatype. In the example, the color temperature of the video frame is declared warm.

The *edge histogram descriptor (EHD)* is calculated by dividing the video frame into 4×4 subframes. Each subframe consists of five bins, each of which represents different edge types, namely vertical, horizontal, 45° diagonal, 135° diagonal, and non-directional edges. The subframes are further divided into non-overlapping, small image blocks to extract all five edge types. If a block contains an edge, the counter of the corresponding edge type is increased by one. If there is no edge in a block, i.e., the region contains only monotonous gray levels, no histogram bin is increased. The five bin values are normalized by the total number of blocks in the subframe, and finally these normalized bin values are quantized. The total number of bins is 4×4×5, i.e., 80 (the number of subframes multiplied by the number of edge types). In the example, the normalized and quantized bins are represented using the `Bincounts` MPEG-7 property, which are declared by 80 values of the `unsigned3` datatype, more specifically, 80 nonnegative integers in the range [0, 7].

The second part of the evaluation has been conducted on videos, as demonstrated in Listing 6.10.

Listing 6.10 Temporal video segment annotation with MPEG-7 Ontology terms

```
<http://example.com/video.mp4#t=1:14:40,1:16:20> mpeg7:temporalDecomposition
<http://example.com/video.mp4> ;
mpeg7:TemporalTransition "PT1N10F"^^mpeg7:VisualTimeSeriesType ;
mpeg7:MotionActivity "3"^^mpeg7:MotionActivityType .
```

The `mpeg7:TemporalDecomposition` predicate declares the temporal segment of `video.mp4`. The `mpeg7:TemporalTransition` predicate declares temporal frame feature transition with value(s) of the `VisualTimeSeriesType` datatype of MPEG-7, which supports two attributes: media duration and an optional offset attribute.

The *Motion Activity Descriptor (MAD)* represents the intensity of activity or action pace of the video segment. It has four parameters: 1) activity intensity, 2) dominant direction of activity, 3) spatial distribution of activity (number of active regions), and 4) temporal distribution of activity (variation of activity over time). Only the first parameter, intensity, is required, the other three values are optional, as shown in the example.

The triples of the case study have been visualized as an RDF graph, which is shown in Fig. 6.1.

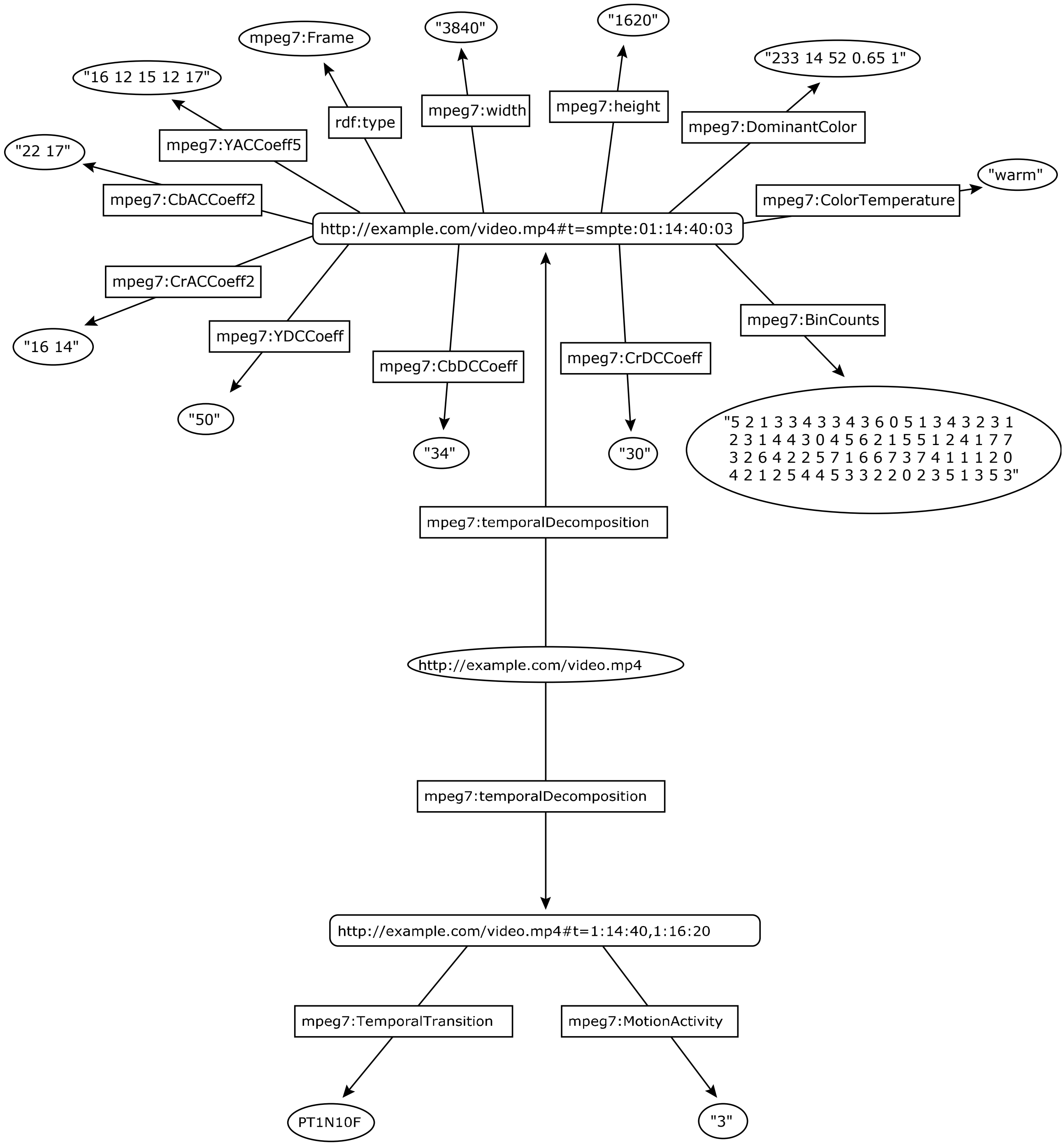


Figure 6.1 RDF graph of the case study

The annotation of the same temporal segments has been attempted with three well-known MPEG-7 OWL mappings, namely, VDO, Rhizomik, and COMM. The MPEG-7 terms used in the previous examples have been checked in these ontologies and compared to the terminology of the proposed ontology (see Table 6.7).

Table 6.7 Comparison of MPEG-7 term definitions

MPEG-7 Term	VDO	Rhizomik	COMM	Proposed Ontology
`Frame`	Not defined	Object property without domain or range (!)	Not defined	Class
`width`	Not defined	Datatype property without range	Not defined	Datatype property with range
`height`	Not defined	Datatype property without range	Not defined	Datatype property with range
`DominantColor`	*Class* (!)	Not defined	Not defined	Property with range
`DominantColorType`	Not defined	Class (!)	*Class* (!)	Datatype
`YACCoeff5`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range
`CbACCoeff2`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range
`CrACCoeff2`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range
`YDCCoeff`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range
`CbDCCoeff`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range
`CrDCCoeff`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range
`unsigned6`	Not defined	Not defined	*Class* (!)	Datatype
`ColorTemperature`	Not defined	Not defined	Not defined	Property with range
`ColorTemperatureType`	Not defined	Not defined	Not defined	Datatype
`BinCounts`	Not defined	Datatype property without range	*Class* (!)	Datatype property with range

Table 6.7 (Cont'd) Comparison of MPEG-7 term definitions

MPEG-7 Term	VDO	Rhizomik	COMM	Proposed Ontology
`TemporalDecomposition`	Not defined	Object property without domain or range	Not defined	Object property with domain and range
`TemporalTransition`	Not defined	Not defined	Not defined	Property with range
`VisualTimeSeriesType`	Not defined	Class (!)	Not defined	Datatype
`MotionActivity`	*Class* (!)	Not defined	Not defined	Property with range
`MotionActivityType`	Not defined	Class (!)	Not defined	Datatype

Class/property names in italic are not compliant with MPEG-7 terminology. Exclamation marks in parenthesis indicate incorrect modeling.

In the case study, the comparison of ontologies has indicated incorrect modeling and/or incomplete term definitions in previous MPEG-7 ontologies, such as datatypes have been defined as classes, domain and range have not been assigned to property definitions, and many terms are simply omitted.

For example, line 1444 of COMM's OWL mapping of MPEG-7 Visual,[32] i.e., `<owl:Class rdf:about="#dominant-color-descriptor">`, defines dominant color as a class, with the comment that it "corresponds to the 'DominantColorType' specified in part 3, page 29–33." This and similar conceptualization errors prove that the authors defined the technical specification of MPEG-7 as an OWL ontology, rather than defining MPEG-7 multimedia terms correctly in OWL. (Also note that the terminology is not standard-aligned, uses kebab-case, which is not the standard casing in ontology engineering.) In Rhizomik, `Frame` is an object property, despite the fact that from the modeling point of view video frame is a concept, not a property, and despite having the properties defined by the MPEG-7 specification for which `Frame` could be the domain. While these properties could be defined as subproperties of `Frame`, it would still prevent statements about video frames, and allow the use of `Frame` as a predicate only, i.e., to express that an image is a "frame of" a certain video file. This, however, can be described as a temporal segment, while the Rhizomik approach cannot ex-

---

[32] http://multimedia.semanticweb.org/COMM/visual.owl

press “is a” relationship for video frames, because it does not define the `Frame` concept. By defining `Frame` as a concept, both representations are possible.

While one might argue that defining a term as a class instead of a property may be correct (depending on the granularity of the ontology), defining MPEG-7 datatype properties as classes yields to a representation model in which several datatype property values cannot be captured, because only subclasses and individuals can be declared for classes, and a class cannot hold data values like a datatype property can. For example, the bin counts of an edge histogram would ideally be declared by 80 3-bit nonnegative integers, as shown in Listing 6.8, yet COMM defined `BinCounts` as a class, making it impossible to represent the corresponding values. In VDO, `BinCounts` is not even defined. Among the analyzed ontologies, only the proposed ontology defines the `TemporalTransition` property.

In contrast to VDO, Rhizomik, and COMM, the proposed ontology follows the MPEG-7 specification, and takes into account knowledge representation and data modeling considerations. As a result, it can be used to formally describe any MPEG-7 concept for media resources and correctly represent property values for datatype properties.

6.2.5 Novelty and Impact

In contrast to previous OWL mappings of MPEG-7, the proposed ontology captures the semantics of the standard via high expressivity while maintaining decidability. The MPEG-7 Ontology is an upper ontology suitable for the representation of images, audio, video, and to a limited extent, 3D scenes, along with still and moving regions of interest. It can also be utilized in spatial, temporal, and spatiotemporal reasoning over multimedia contents. It exposes the entire vocabulary of MPEG-7 through a compact, permanent URI, and unlike its predecessors, it provides a physical URI for each term. This approach provides excellent documentation for practitioners, thereby helping the deployment of MPEG-7 terms.

The applicability of the ontology to real-life scenarios has been tested via case studies, one of which has been briefly described. In the case studies, the proposed ontology outperformed previous MPEG-7 ontologies in terms of modeling correctness, MPEG-7 term coverage, and definition completeness. As a result, the proposed MPEG-7 Ontology can be used to annotate multimedia concepts and properties previous OWL mappings of the MPEG-7 XSD files cannot. Being an upper ontology, it can be combined with specific domain ontologies to further detail semantics of multimedia contents. Due to its comprehensiveness, the MPEG-7 Ontology can be considered the ultimate multimedia ontology for annotating low-level multimedia descriptors, complementing feature extraction and suitable for machine learning-based classifications, as well as feature-based and content-based multimedia indexing and retrieval.

6.3 Summary

The unique characteristics of videos impose special requirements for video ontologies. The majority of upper and domain ontologies introduced for representing multimedia contents in the last decade and a half, with or without MPEG-7 alignment, lack complex role inclusion axioms and rule-based definitions of common video events, and are limited to highly specific terminological and assertional knowledge. For this reason, most of these formalisms are actually controlled vocabularies or taxonomies only, not fully-featured ontologies, hence they are not suitable for advanced multimedia reasoning. To address the above issues, concepts, roles, individuals, and relationships of professional video production, video distribution, and common video events have been formally modeled using $\mathcal{SROIQ}^{(\mathcal{D})}$, one of the most expressive decidable description logics, and implemented as an OWL 2 ontology called the Video Ontology (VidOnt). This ontology is the only standard-aligned core reference ontology for video production and spatiotemporal annotation that not only collects standard and de facto standard videography terminology, but also defines terms of practical importance to be used in RDF triples for describing audiovisual content.

A well-designed OWL ontology for the MPEG-7 standard has long been desired, yet not developed for nearly two decades due to modeling difficulties and ontology engineering challenges. For this reason, a novel ontology has been proposed, which is the first formally grounded MPEG-7 ontology with proper conceptual modeling. It has been implemented in OWL 2, verified against ISO/IEC 15938, checked for consistency with industry-leading reasoners, and evaluated through case studies.

Chapter 7
Medical Video Interpretation

> “Contradiction is not a sign of falsity,
> nor the lack of contradiction
> a sign of truth.”
>
> — *Blaise Pascal*

Although there are multiple options to represent and interpret videos, some of them are limited to well-defined knowledge domains and constrained environments. Temporal description logic axioms enable the temporal representation of video content at the abstraction level, as opposed to previous approaches that make this possible only at the interpretation level. Also, reasoning over these temporal DL axioms can be used for video event recognition without SWRL rules. To exploit even more semantics in video understanding, hidden 3D information can be utilized by the spatiotemporal annotation of objects and their 3D characteristics.

7.1 Ontology-Based Reasoning

Patient records can be semantically enriched with medical semantics and data obtained via machine learning to be used in intelligent medical applications. Medical image taxonomy can be used for classification, anatomy and features for anomaly analysis, and pathology or disease for semantic tagging. Machine learning can provide categorical examples for classification, annotated references for anomaly analysis, and multimodal context data for tagging. The medical expert knowledge formally described using this data yields to actionable information, such as potentially depicted concepts with normalized certainty values of concept depiction, for computer-aided diagnosis, patient-centered medicine, computerized triage, and coincidental diagnosis [153].

Rules have been developed for tumor assessment via reasoning, such as to classify the longest diameter from a pair of length annotations for a lesion, classification of findings as pathologic or non-pathologic represented using ontology terms, temporal classification of the valid time of lesions, and classification of image findings on baseline images as measurable or non-measurable [154].

Low-level MRI image data combined with rule-based formalisms and ontologies are proven to be efficient for the semantic annotation of anatomic structures via reasoning [155].

Reasoning over the semantic annotation of tumor images can be used for automated classification of cancer tumors [156]. However, differentiating cancerous and non-cancerous tumors does not have to rely on image semantics alone; statistical features and wavelet transform features may also be processed to complement the semantic features used for this purpose [157].

Automatic anatomy detectors and OWL reasoning processes can be used to preprocess medical images automatically and provide accurate input to further, more complex reasoning services [158].

7.2 Video Event Recognition via Reasoning over Temporal DL Axioms

As opposed to the combination of a general description logic, a temporal ontology, and unique, URL-based identifiers for representing video events (as seen earlier in Listing 5.1 and 5.2), the hybrid description logic formalism introduced as part of this thesis gives a whole new representation for the same scene, as shown in Listing 7.1.

Listing 7.1 Video event representation with the proposed hybrid DL-based formalism

Video(LASERVITREOLYSISVIDEO) $\rhd$ 1.0
Scene(FLOATERSCENE) $\rhd$ 1.0
sceneFrom(FLOATERSCENE, LASERVITREOLYSISVIDEO) $\rhd$ 1.0
depicts(FLOATERSCENE, LaserVitreolysis) $\rhd$ 1.0
StillRegion(ROI) $\rhd$ 1.0
$\mathrm{ROI} \sqsubseteq \forall R.L \rhd 1.0$

Listing 7.1 (Cont'd) Video event representation with the proposed hybrid DL-based formalism

7: $\mathrm{depicts}(\mathrm{ROI}, \mathrm{Floater}) \rhd 1.0$

8: $\mathrm{FSM} \equiv \diamond(00{:}00{:}48)(00{:}00{:}48 \; b \; \#).(F@00{:}00{:}48)$

9: $\mathrm{LHF} \equiv \diamond(00{:}00{:}50)(00{:}00{:}50 \; a \; \#).(L@00{:}00{:}50)$

where

ROI	the region of interest of the floater treatment scene, i.e., the depicted Weiss ring floater
L	the point of laser beam focus

Owing to the RCC8 support of the proposed formalism, it is straightforward to describe the non-tangential proper part inverse relation between the floater and the laser point (6). The temporal axioms enable the formal description of the floater treatment scene with the reference interval #, before which the floater stopped moving around (FSM) at time point 00:00:48 (8), as annotated by Allen's before relation b. Interval # was followed by the laser beam hitting the floater (LHF) at time point 00:00:50, and represented using Allen's after relation a (9).

Sophisticated statements are not only be explicitly stated as background knowledge, but can also be inferred (see Listing 7.2), such as the floater has been transformed by the laser to ionized gas (1) or the treatment scene is part of the floater video (2).

Listing 7.2 Examples for inference

1: $V\diamond(V, W)(V \; b \; W).(\mathrm{LHF}@V \sqcap \mathrm{FE}@W)$

2: $\mathrm{partOf}(\mathrm{VITREOLYSIS}, \mathrm{FLOATERVIDEO})$

where LHF represents the laser hitting the floater at time point V, FE represents the floater transformation at time point W, and b is Allen's before relation.

Depending on the nature of the treatment and the characteristics of the floater, the formal description of floater treatment videos can also include the following:

- the out-of-focus scenes, during which the practitioner ensures adequate distance from both the eye lens and the retina for safety;
- the location and shape changes of treatment-resistant floaters that need to be hit by the laser multiple times before they can be converted from solid material that cast shadows on the retina to ionized gas (plasma) that can be absorbed into the eye.

7.3 Exploiting Hidden 3D Information for Video Understanding

3D models play an important role in a wide range of applications, including engineering, medicine, 3D printing, scientific visualization, education, and entertainment. However, the automated processing of 3D models is rather limited due to the difficulty of capturing their semantics. For instance, surgeons can use accurate 3D printed human organs for preoperative diagnostic techniques to improve surgical decisions (as opposed to 2D images of X-ray, ultrasound, and MRI for surgical planning) [159], but the most appropriate 3D models cannot be retrieved efficiently from an organ repository by simple keywords or labels used in traditional information retrieval methods. It would be desirable to find the model of interest by its characteristics, which requires rich semantics that can be provided only by high-level structured descriptors. Moreover, it would also be desirable to accurately describe unique structural complexities of custom 3D models [160], which, when semantically enriched, could help retrieve models of successfully operated similar cases for surgical training.

3D model repositories usually provide textual information about technical characteristics in the form of descriptions and tags. Such data include complexity (low-poly vs. high-poly), animation-readiness (rigged or not), file format (e.g., .max, .3ds, .obj, .blend, .fbx, .ztl), classification category (e.g., body parts, organs, clinical procedures, medical devices), and licensing (e.g., Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International). However, textual descriptions do not provide machine-interpretable structured data about the visual appearance of the represented models, nor about the meaning of 3D scenes. Consequently, such information does not make 3D models searchable by 3D characteristics, such as size, shape, material, and shininess.

Low-level feature descriptors automatically extracted from 3D models, and the feature aggregates and statistics based on them, provide information that might be useful for classification, object matching, and object tracking, but they are usually inadequate for 3D scene understanding. 3D shape descriptors include view-based (e.g., compact multi-view descriptor, light-field descriptor), histogram-based (e.g., 3D shape spectrum descriptor, bag of features (BoF), generalized shape distributions), transform-based (e.g., spherical harmonics descriptor, PCA spherical harmonics transform, spherical trace transform descriptor), graph-based (e.g., skeletal graph descriptor, Reeb graph descriptor), and hybrid 3D descriptors. While low-level descriptors can be used for training machine learning systems, they are generally inefficient for scene interpretation and content-based retrieval, because they cannot describe the meaning of the visual content and appearance in a machine-interpretable way due to a lack of semantics. The notorious semantic gap, the discrepancy between the automatically extractable low-level features and the high-level semantics, which is extensively described in the literature for image semantics, and to a lesser extent for video semantics,[1] also poses a challenge for 3D model descriptions [161].

Photos and 2D videos of real-world objects often do not provide the information needed to fully understand their geometry and material-related features.[2] The precise representation of real-world objects requires shape measurements and spectrophotometric property acquisition, typically obtained using 3D laser scanners, RGB-D depth cameras, Kinect depth sensors, structured light devices, photogrammetry, and photomodeling [162]. While many only think of the dimension and shape of a model when it comes to 3D models, there are many other properties to be captured, for example, material, texture, transparency, reflectivity, dichroism, coating, and finish. Based on their background, expert knowledge, and experience, practitioners are capable of defining these rich semantics for 3D models in the form of high-level descriptors, many of which are correlated to modeling choices and parameters that have been standardized as terms, classes and properties, and parameter value ranges. The first prominent 3D annotation standard was the Virtual Reality Modeling Language (VRML, ISO/IEC 14772–1:1997), which is now succeeded by Extensible 3D (X3D, ISO/IEC 19775, 19776, and 19777). The X3D standard is supported natively or via plugins by both industry-leading proprietary and open source 3D computer graphics software, such as AutoDesk 3ds Max, AutoDesk Maya, AC3D, Modo, Blender, and Seamless3d.

The vocabulary of X3D, however, was originally released in semistructured XML Schema (XSD), which is machine-readable yet not machine-interpretable, making the standard inefficient for automation, data sharing, and data reuse. These limitations can be addressed using Semantic Web standards, such as RDF, RDFS, and OWL, so that the structured representation of the correspond-

---

[1] Sikos, L. F. (2016) RDF-powered semantic video annotation tools with concept mapping to Linked Data for next-generation video indexing. Multimedia Tools and Applications, 76(12):14437–14460. doi: 10.1007/s11042-016-3705-7

[2] Sikos, L. F. (2016) Rich semantics for interactive 3D models of cultural artifacts. In: Garoufallou, E., Subirats, I., Stellato, A., Greenberg, J. (eds.) Metadata and semantics research, pp. 169–180. 10th International Conference on Metadata and Semantics Research, Göttingen, Germany, 22–25 November 2016. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-49157-8_14

ing concepts becomes machine-interpretable by linking them to their formal definitions and other, related concepts from highly specialized datasets and the *Linked Open Data (LOD) Cloud*[3] [163], and mapping them to geometric primitives [164]. The formal structured knowledge representation of 3D models enables efficient annotation, segmentation, indexing, and retrieval. Ontology-based 3D model retrieval can be performed not only by textual descriptions, but also by 3D characteristics such as shape or material [165].

For this reason, there have been attempts to map the application-specific XML Schema (XSD) file of X3D to domain-specific machine-interpretable metadata terms defined in OWL, none of which was sound or complete so far. Previous OWL mappings of X3D have not been formally grounded and did not exploit the repertoire of mathematical constructors available in the implementation language, which is a known limiting factor for the reasoning potential of ontologies.[4]

Due to the limitations of unstructured thesauri and semistructured controlled vocabularies, our focus here is exclusively on those 3D ontologies that provide structured data, namely, RDFS and OWL 3D ontologies. Structured ontologies in the 3D modeling domain without X3D alignment include the *Geometrical Application Ontology* [166], the *Common Shape Ontology* [167], the *3D Graphics Ontology*, the *Constructive Solid Geometry Modeling Ontology*, the *3D Plant Modeling Ontology* [168], 3D interaction ontologies [169], the *Kinect Ontology*,[5] the *Ontology of 3D Visualization Techniques (O3VT)*,[6] and 3D and 4D enterprise modeling ontologies [170]. 3D domain ontologies without X3D alignment cover a narrow knowledge domain in fields such as medicine (e.g., *Spine Ontology* [171]), engineering (e.g., *Furniture Ontology*[7]), and human modeling (e.g., *Ontology for Virtual Humans* [172]. Since the 3D ontologies not based on X3D cannot represent arbitrary 3D structures and characteristics, the 3D ontologies of our interest are the X3D-based ontologies. Similar to the OWL mappings of other multimedia standards originally written in XSD used partly or

---

[3] http://lod-cloud.net

[4] Sikos, L. F. (2016) A novel approach to multimedia ontology engineering for automated reasoning over audiovisual LOD datasets. In: Nguyễn, N. T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) Intelligent information and database systems, pp. 3–12. 8th Asian Conference on Intelligent Information and Database Systems, Đà Nẵng, Vietnam, 14–16 March 2016. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-49381-6_1

[5] http://users.abo.fi/rowikstr/KinectOntology/KinectOntology.owl

[6] http://cui.unige.ch/isi/onto/o3dvt/O3DVT.owl

[7] http://cui.unige.ch/isi/onto/2010/12/furniture.owl

solely for representing 3D models and animations, such as MPEG-7[8,9,10] and CityGML[11], there have been multiple attempts to map the XML Schema file of X3D to OWL.

The first OWL mapping of X3D, *OntologyX3D*,[12] was created by mapping X3D node elements representing graphics and virtual reality concepts into OWL classes [173]. The classes of OntologyX3D aimed to group object geometry and appearance, scene navigation, lights, environmental effects, sound, sensors, events, and animation. The class hierarchy was built to correspond to VRML and X3D scene graphs. A set of semantic properties has also been declared in OntologyX3D, which was suitable for the basic description of 3D scenes.

The *3D-CO Ontology* was another OWL mapping of X3D [174]. It defined the universal class `Scene`, the equivalent of the root element of the X3D Schema, attached to CIDOC-CRM for representing cultural heritage objects.

Gilson and Silva mapped a small subset of the XML Schema file of X3D to RDFS as part of their SVG to X3D translation.[13] This mapping was built by loading the source and target ontology schemata into the Ontology MApping FRAmework Toolkit (MAFRA),[14] and creating semantic bridges between the two ontologies.

A more recent OWL mapping of the X3D XML Schema file was based on a discussion to semantically enrich CAD models [175], and has been generated using XSLT [176]. The transformation process indicated several consistency issues of the X3D standard. In contrast to previous attempts, this mapping focused not only on the classes, but also on the properties, resulting in a basic role hierarchy. While the resulting ontology overcame many issues of its predecessors and covered the most up-to-date version of X3D at the time, it had its own ontology engineering issues. For example, it defined property ranges with VRML and X3D datatypes, e.g., `SFBool`, `MFInt32`, rather than the globally deployed XML Schema datatypes, such as `xsd:boolean` and `xsd:integer`, which would have been a better choice to maximize interoperability. Moreover, the mapping made the false assump-

---

[8] http://rhizomik.net/ontologies/2005/03/Mpeg7-2001.owl

[9] http://multimedia.semanticweb.org/COMM/visual.owl

[10] http://mpeg7.org/mpeg7.ttl

[11] http://cui.unige.ch/isi/onto//citygml2.0.owl

[12] http://people.cs.umass.edu/~kalo/papers/graphicsOntologies/OntologyX3D.zip

[13] http://www.cs.swan.ac.uk/~csowen/SVGtoX3D/examples/X3D_OntologyRDFS.htm

[14] http://mafra-toolkit.sourceforge.net

tion that X3D nodes logically correspond to OWL classes, and incorrectly mapped `xsd:complexType` declarations, which should be able to hold values such as vectors, to `owl:Class`, rather than `owl:datatypeProperty` with a range declaration (`rdfs:range xsd:complexType`). The XLink cross references have been mapped to OWL object properties, and the X3D element attributes to string literals (defined as `owl:datatypeProperty` with a range declaration set to `xsd:string`). The ontology defined default property values as RDFS comments. Considering the high number of datatype properties in X3D, this approach lost the semantics of many X3D roles. Another issue of this mapping, similar to all the previous X3D mappings, is the inadequate description of the defined concepts and roles.

7.3.1 The 3D Modeling Ontology (3DMO)

To address the aforementioned issues, to cover all features of X3D, including the ones that are new to X3D 3.3, and to add X3DOM terms, a novel OWL 2 ontology, the *3D Modeling Ontology (3DMO)*,[15] has been developed with standards alignment and mathematical grounding using a description logic formalism (see A.2).[16] In addition, the X3D and X3DOM terms have been complemented by a large set of new concepts and roles excluded from X3D but used by 3D graphic designers, 3D animators, and CAD designers, covering the most common terminology of AutoDesk 3ds Max, Maya, and AutoCAD.

7.3.1.1 Modeling Challenges

Several stages of ontology engineering indicated modeling issues derived from the original XSD file of the X3D standard, which have been resolved as follows.

A permanent namespace URL is imperative for the viability of any ontology, therefore a new namespace has been introduced for the OWL 2 mapping of the X3D standard, which supports on-

---

[15] http://3dontology.org

[16] Sikos, L. F. (2016) Rich semantics for interactive 3D models of cultural artifacts. In: Garoufallou, E., Subirats, I., Stellato, A., Greenberg, J. (eds.) Metadata and semantics research, pp. 169–180. 10th International Conference on Metadata and Semantics Research, Göttingen, Germany, 22–25 November 2016. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-49157-8_14

tology versioning and takes into account the proposed (but not yet standardized) namespace structure. Using a dedicated URL and omitting the file extension and the `specifications` directory from this namespace structure yields to `http://3dontology.org/`, which points to the latest ontology version at any given time. This namespace can be abbreviated by the `t3dmo:` prefix for the concepts, roles, and individuals of the ontology. The web server hosting the ontology serves the ontology file for semantic agents and HTML5 to web browsers through content negotiation, as per the suggestion of the World Wide Web Consortium (W3C) [177].

There is no direct matching between the nodes, fields, and values of X3D and description logic concepts and roles (or OWL classes, object properties, and datatype properties). Therefore, transparent translation of the X3D XML Schema file to description logics or OWL was not an option. To make the task more challenging, the X3D standard does not define role hierarchy and complex relations. For this reason, the XSD file of X3D 3.3 proved to be inadequate for mapping the X3D vocabulary to OWL 2. To address these issues, a new concept hierarchy and a new role hierarchy have been manually created. Due to the different aims and designs, and age, of the preceding OWL mappings of X3D, no previously released mappings have been used in the 3D Modeling Ontology.

The naming conventions of X3D made property modeling a challenge, because X3D properties have a greatly varying scope, and the domain of some properties is a set of classes, rather than a single class. Those X3D terms that represent concepts and roles at the same time, such as `color` and `geoOrigin`, led to conceptual ambiguity issues. To resolve these issues, the role names have been extended to reflect specificity, thereby ensuring that they can be understood without contextual information, which is imperative in knowledge representation.

The consistency issues of X3D had to be addressed as well. Several concepts share homonymous roles in the standard, such as `color`, which can be defined for `BlendMode` and for `ColorRGBA`. In the first case, it represents the constant color used by the blend mode constant, in the second case it corresponds to a set of RGBA colors. This issue was resolved by term name extension and domain definitions. The multiple descriptions for a property defined for two different concepts with the same name are misleading in the standard. This was addressed by modifying, and sometimes extending, their descriptions, and declaring them using Dublin Core descriptions (`dc:description`). In the X3D standard, there are ambiguous properties that are defined for multiple concepts with a different meaning, are of a different type, or have a different range and domain. For example, the `bottom` object property of a cubemap defines the texture for the bottom of the cubemap. The `bottom` datatype property of a cylinder specifies whether the bottom cap of the cylinder is created. In an OWL ontology, a property has to be either an object property or a datatype property. Therefore, the corresponding atomic roles have also been amended.

To further increase confusion, there are property values in the X3D standard that correspond to multiple datatypes. For example, the `axisRotation` of `CylinderSensor` can be a vector or a floating-point number. In the first case, `axisRotation` represents the local sensor coordinate system. In the second, it specifies whether the lateral surface of the virtual cylinder or the end-cap disks of the virtual geometry sensor are used for manipulation, or constrains the rotation output of the cylinder sensor. For these properties, two options have been considered:

1) the extension of the property name to be more specific, or
2) the implementation of the less restrictive datatype of the two, which was applicable only to those datatypes that are supersets of the other. To add context to the corresponding properties, the descriptions have been updated accordingly.

The X3D specification features proprietary datatypes, many of which, such as `SFNode` and `SFColor`, are inherited from VRML. Wherever possible, all datatypes have been converted to standard XSD datatypes to maximize interoperability. For example, `xsd:boolean` has been used instead of `x3d:SFBoolean`, and the `x3d:SFVec3f` declarations have been replaced by `xsd:complexType`. The X3D standard declares URLs as strings, which have been declared in the new ontology using the more appropriate `xsd:anyURI` datatype instead. The majority of X3D properties are datatype properties, many of which have an array of permissible values (representing vectors or matrices) rather than just one value of a specific datatype. Such datatype properties have been defined as `xsd:complexType` rather than the more specific but single-value datatypes, such as `xsd:float`, `xsd:integer`, `xsd:string`, or `xsd:anyURI`.

The discrepancy between the X3D terminology and the 3D modeling terms used in AutoDesk 3D modeling software tools has also been considered, and concept relatedness defined across terminologies. For example, the image file paths of bump maps used in AutoDesk 3ds Max and AutoDesk Maya correspond to the attribute values of `url` attributes of multiple instances of the `ImageTexture` node within an instance of the `MultiTexture` node, which, however, cannot be inferred based on concept inclusion. Such semantic similarities have therefore been defined explicitly, which enables the implementation of the most specific concept in 3D scene representations, and also enables differentiating between similar concepts.

7.3.1.2 Conceptual Modeling with Best Practices

In contrast to the common practice of creating OWL ontologies using a graphical interface without formal grounding, the 3D Modeling Ontology has been developed manually and in accordance with ontology engineering best practices. Hence, the development of the ontology included the following steps: specification, knowledge acquisition, conceptualization, term enumeration, concept hierarchy building, the definition of roles and the role hierarchy, individual inclusion, evaluation, and documentation.[17]

The scope of the 3D Modeling Ontology was defined to cover the knowledge domain of 3D modeling with reference to the representation of arbitrary 3D models and 3D scenes. The purpose of the 3D Modeling Ontology is to provide a comprehensive coverage of 3D modeling concepts and roles with X3D alignment. Being an upper ontology, the vocabulary of the 3D Modeling Ontology can be combined with concepts of domain ontologies and LOD datasets as usual to further detail high-level semantics not applicable to arbitrary 3D scenes, such as SNOMED-CT terms[18] for medical 3D models, or Kerameikos terms[19] for cultural artifact models.

The in-depth analysis of the X3D and X3DOM documentation,[20] the X3D nodes, fields, and permissible property values in the XSD file of X3D 3.3 served as the basis for the conceptualization, term enumeration, and concept hierarchy. Based on the resulting description logic formalism, OWL classes and properties have been created. In contrast to previous OWL mappings of X3D, which used RDF/XML serialization, the 3D Modeling Ontology was written in Turtle, mainly because the RDF/XML serialization would have been too verbose for representing the large number of concepts and properties.

---

[17] Sikos, L. F. (2017) Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5

[18] http://purl.bioontology.org/ontology/SNOMEDCT

[19] http://kerameikos.org

[20] http://doc.x3dom.org

As it was infeasible to use automated ontology engineering techniques, such as XSLT transformation, natural language processing-based concept extraction, and statistical and machine learning techniques during ontology engineering, the components of the 3D Modeling Ontology have been individually assessed and manually coded. This approach resulted in code optimality and an easy-to-read, neat layout for future extensions, which is in sharp contrast to those ontologies that have been generated using the OWL API and contain a large number of whitespaces and multi-line comments about trivial ontology term IRIs, and order concepts and roles with a logic that makes ontology implementation difficult.

The TBox of the 3D Modeling Ontology is not solely based on the X3D vocabulary; new concepts (e.g., `t3dmo:3DModel`, `t3dmo:Dodecahedron`, `t3dmo:DesignStudio`) and new roles (e.g., `t3dmo:baseForm`, `t3dmo:animated`, `t3dmo:designedBy`, `t3dmo:hasVertices`) used by 3D graphic designers have also been added to maximize the implementation potential.

7.3.1.3 Formal Grounding

The $\mathcal{SROIQ}^{(\mathcal{D})}$ underpinning enables complex reasoning tasks for all the 3D applications that utilize the 3D Modeling Ontology. The main benefits of description logics are the higher efficiency in decision problems than first-order predicate logic and the expressivity higher than that of propositional logic. As such, description logics are ideal for modeling concepts, roles, individuals, and the relationships between them. For this reason, the 3D Modeling Ontology was formally grounded in the form of description logic axioms, which have then been translated to OWL 2 (see Table 7.1).

Table 7.1 Examples for description logic to OWL 2 translation

DL Axiom	OWL Implementation
$\text{Polyhedron} \sqsubseteq \text{Polytope}$	`t3dmo:Polyhedron rdfs:subClassOf t3dmo:Polytope .`
$\text{Box} \sqcap \text{Pyramid} \sqsubseteq \bot$	`t3dmo:Box owl:disjointWith t3dmo:Pyramid .`
$\exists \text{topRadius}.\top \sqsubseteq \text{Cone}$	`t3dmo:topRadius rdfs:domain t3dmo:Cone .`
$\text{createdBy} \equiv \text{creatorOf}^{-}$	`t3dmo:createdBy a owl:ObjectProperty ;` `owl:inverseOf t3dmo:creatorOf .`

Table 7.1 (Cont'd) Examples for description logic to OWL 2 translation

DL Axiom	OWL Implementation

$\top\sqsubseteq\forall{\geqslant}0\text{zeroone.float}$
$\top\sqsubseteq\forall{\leqslant}1\text{zeroone.float}$
	`t3dmo:zeroone rdfs:range [ a rdfs:Datatype ;`
	`owl:onDatatype xsd:float ; owl:withRestrictions (`
	`[xsd:minInclusive "0"]`
	`[xsd:maxInclusive "1"] ) ] .`

$\top\sqsubseteq\forall\text{transparency.zeroone}$ $\exists\text{transparency.}\top\sqsubseteq\text{Material}$	`t3dmo:transparency rdfs:range t3dmo:zeroone ;` `rdfs:domain t3dmo:Material .`

Since the DL expressivity of the 3D Modeling Ontology is $\mathcal{SROIQ}^{(\mathcal{D})}$, it exploits all mathematical constructors of the implementation language, OWL 2 DL. While this makes the combined reasoning complexity N2ExpTime-complete, the language of the ontology retains decidability and enables advanced reasoning tasks that have not been feasible with any of the previous 3D ontologies.

The concept terms that are suitable for representing 3D objects have been selected from the set of terms acquired from the X3D vocabulary, as well as the panels and settings of AutoDesk 3ds Max, AutoDesk Maya, AutoDesk AutoCAD, and their official documentation at the AutoDesk Knowledge Network.[21] Based on these terminologies, atomic concepts have been created with X3D-alignment, and enumerated for inclusion in the 3D Modeling Ontology. Synonymous concepts have been identified and declared as equivalent concepts. Similar and related concepts have also been defined.

After defining the concepts, they have been arranged in a taxonomic hierarchy with subsumption declarations of the form $C \sqsubseteq D$, in which, if concept D is more general than concept C, then every instance of C is also an instance of D.

The general structure of the ontology is based on the X3D and X3DOM node hierarchy described in the official specification. As mentioned before, however, not all X3D nodes can be transformed directly into a description logic concept or role. Moreover, X3DOM has a considerable divergence from the X3D standard. To eliminate the ambiguity derived from the X3D nodes defined as the logical equivalent of both a concept and a role, the 3D Modeling Ontology applies an extended name

---

[21] https://knowledge.autodesk.com

and different capitalization for such concepts and roles. This settled the question of determining when to add a new concept or represent the differences with role values instead. In other words, a new concept has been introduced in each case when specific concepts had properties the more general concepts did not, the property restrictions of a specific concept were different from those of the more general concept, or when the specific concepts have been associated with different relationships than the more general concept. Concepts that had one direct subconcept only have been examined to ensure a correct and comprehensive representation. Similarly, intermediate concepts have been considered for all the concepts with numerous subconcepts to avoid extremes, such as overly specific concepts, very low number of roles, a flat concept hierarchy, or the incorrect representation of the majority of concept differences as role values.

As the next step, the roles and the role hierarchy have been defined. The domain and range of the roles have been defined with the most general concept or concepts possible. The permissible values or value ranges have been defined for all datatype properties. The 3D Modeling Ontology features proprietary datatypes with XSD-alignment for its datatype properties. Although the X3D node hierarchy provided a good starting point for creating the concept hierarchy, the specifications upon which a role hierarchy could have been created are literally missing from the X3D standard. Due to the number of role overlaps and the different meanings and characteristics of homonymous atomic roles, however, the X3D roles cannot be categorized efficiently in a taxonomical tree structure anyway, and role scopes are indicated mainly by their domain declarations instead.

Deciding which object to define as a concept or an individual determined the lowest level of granularity for the ontology. The X3D vocabulary does not contain individuals, which, however, would be beneficial in 3D model annotations. Individual assertions not only collect information about individuals, but also serve as examples for a particular concept type, making the ontology implementation easier by clearly indicating the intended use and specificity of concepts. Therefore, the 3D Modeling Ontology defines an ABox with individuals, such as colors, materials, 3D computer graphics software, and animation studios.

The first version of the 3D Modeling Ontology features 526 classes, 780 properties, and 82 individuals in the form of 3,810 axioms, covering the entire vocabulary of X3D 3.3, and core concepts and roles of the 3D modeling industry.

7.3.1.4 Evaluation

The 3D Modeling Ontology has been evaluated according to the five ontology engineering principles of Gruber:

- Clarity: the intended semantics of the 3D Modeling Ontology terms have been defined with machine-interpretable constraints for software agents and concise descriptions with contextual information for human consumption. Entities and string literals have been checked using a U.S. English spellchecker. Although the atomic concepts and roles defined in the ontology characterize the 3D modeling domain, their definition is independent from the modeling context.
- Coherence: the integrity of the 3D Modeling Ontology has been checked throughout the ontology engineering process with industry-leading reasoners, including FaCT++ and HermiT. This guarantees that no RDF statement can be automatically inferred from the axioms of the ontology that would contradict any given definition. Since the `xsd:complexType` datatype, which is used extensively in the 3D Modeling Ontology, is not yet supported by FaCT++, the final integrity checking was performed using HermiT. The correctness of the ontology has been reviewed by the Web3D Consortium, the association that defined the X3D standard.
- Minimal encoding bias: in the 3D Modeling Ontology, the conceptualization of the 3D domain has been specified at the knowledge level independent from any symbol-level encoding. The ontology engineering has been conducted using open standards, such as RDF, RDFS, OWL, and Turtle, rather than proprietary specifications, serializations, or file formats.
- Minimal ontological commitment: the 3D Modeling Ontology has been designed to be as lightweight as possible through the inclusion of truly relevant concepts, the correct use of namespaces, the shorthand annotation of RDF triples of the same subject, and the use of the compact serialization Turtle. De facto standard ontology terms have been reused from mainstream ontologies, such as Creative Commons, Dublin Core, DBpedia, FOAF, and Schema.org, according to the Semantic Web best practices.

- Extendibility: new concepts, roles, and individuals can be easily added to the ontology without changing the core concept or role hierarchy. The 3D Modeling Ontology is aligned with XSD, X3D, and X3DOM, and can be easily interlinked with LOD datasets.

The description logic expressivity of the 3D Modeling Ontology, $\mathcal{SROIQ}^{(\mathcal{D})}$, is significantly higher than that of other 3D ontologies, which makes the 3D Modeling Ontology the most expressive ontology to date in the 3D modeling domain and suitable for reasoning over 3D models and scenes (see Table 7.2).

Table 7.2 Comparison of 3D ontologies

Ontology	Language	DL Expressivity	X3D Alignment
CityGML 2.0 Mapping	OWL	$\mathcal{ALEHI}^{(\mathcal{D})}$	–
Furniture Ontology	OWL 2	$\mathcal{ALCHQ}$	–
Gilson-Silva Mapping	RDFS	$\mathcal{AL}$	+
Kinect Ontology	OWL 2	$\mathcal{ALCRIF}^{(\mathcal{D})}$	–
O3DVT	OWL 2	$\mathcal{ALEHQ}$	–
OntologyX3D	OWL	$\mathcal{ALUIN}^{+(\mathcal{D})}$	+
Petit Mapping	OWL	$\mathcal{AL}^{(\mathcal{D})}$	+
3D Modeling Ontology	OWL 2	$\mathcal{SROIQ}^{(\mathcal{D})}$	+

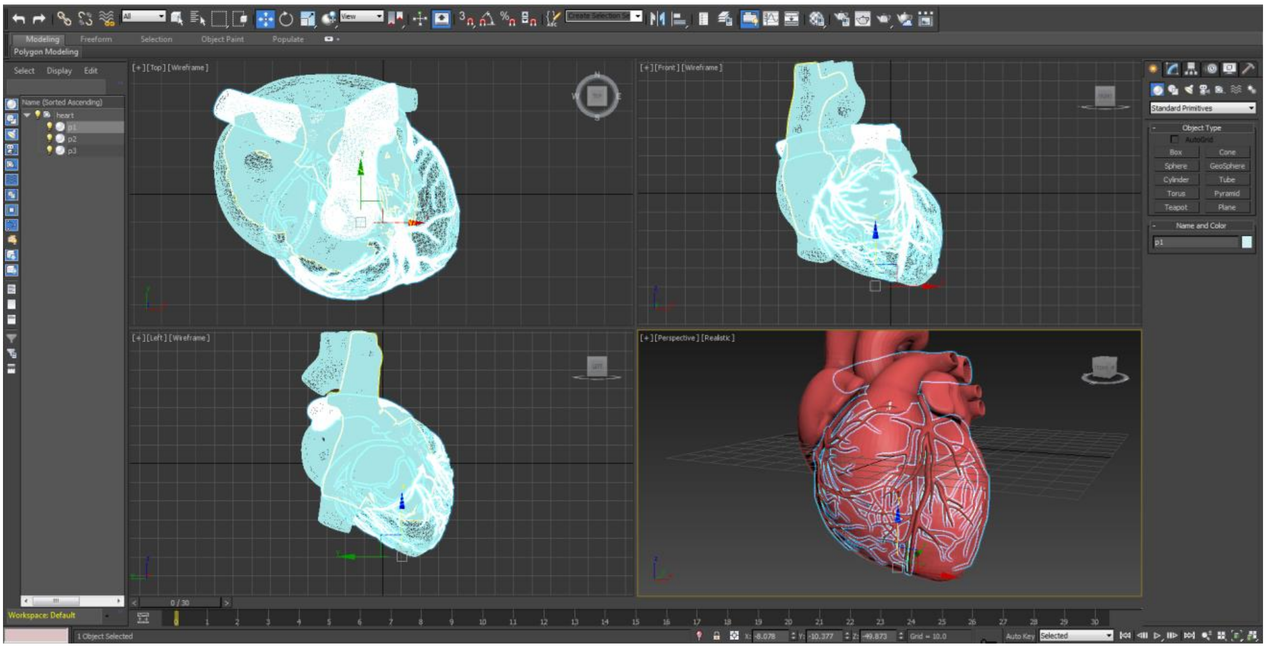
This comparison includes all the 3D ontologies that were available online at the time of writing.

7.3.1.5 Case Study 1: Keyphrase-Based vs. Ontology-Based 3D Model Retrieval

The first case study involved feature-based 3D model indexing and retrieval by formally describing 3D models of human anatomy, retrieving the ones with particular 3D characteristics using SPARQL, and comparing the results with traditional keyphrase-based search.

Although geometric and spectrophotometric properties manipulated in 3D modeling software tools can be directly described in any X3D-compliant knowledge representation, only the 3D Modeling Ontology defines machine-interpretable terms for these features, and additional information about the modeling software in which the model was created and the geometric primitives that make up the model.

To demonstrate how the 3D Modeling Ontology can be used for the structured representation of 3D models, assume a heart model created in AutoDesk 3ds Max, and describe it using terms from the 3D Modeling Ontology (see Fig. 7.1).



```
@prefix t3dmo: <http://3dontology.org/> .
@prefix snomedct: <http://purl.bioontology.org/ontology/SNOMEDCT/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://3dontology.org/3dmodels/heart/> a t3dmo:3DModel , snomedct:80891009 ;
t3dmo:createdIn t3dmo:AutoDesk3dsMax ;
t3dmo:baseForm t3dmo:prolateSpheroid ;
```

Figure 7.1 The 3D Modeling Ontology provides structured representation for 3D models with X3D-aligned descriptors for geometric primitives and 3D characteristics that constitute a model

```
t3dmo:hasFaces "177454"^^xsd:nonNegativeInteger ;
t3dmo:hasEdges "532362"^^xsd:nonNegativeInteger ;
t3dmo:hasVertices "92026"^^xsd:nonNegativeInteger ;
…
t3dmo:diffuseColor "0.745 0.090 0.090"^^xsd:complexType ;
t3dmo:specularColor "0.098 0.098 0.098"^^xsd:complexType ;
t3dmo:transparency "0.000"^^xsd:decimal .
```

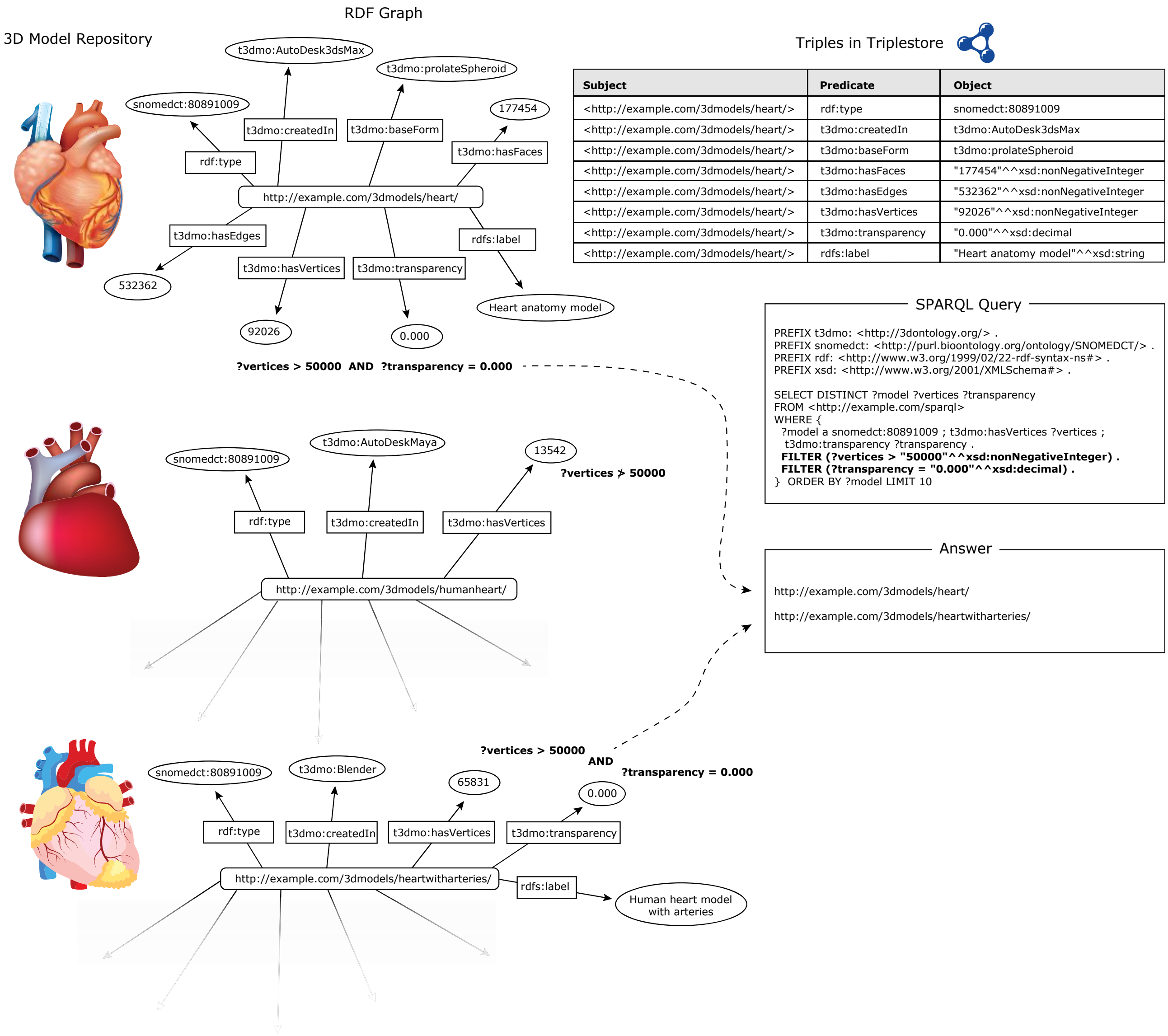
Figure 7.1 (Cont'd) The 3D Modeling Ontology provides structured representation for 3D models with X3D-aligned descriptors for geometric primitives and 3D characteristics that constitute a model

While 3ds Max scenes can be exported to X3D using the InstantExport plugin of Fraunhofer IDG,[22] it only supports semistructured formats, such as XML and XHTML, and leaves rich semantics unexploited (by typically utilizing less than 10 terms from the X3D vocabulary). X3D files generated automatically with InstantExport can be published on websites without proprietary plugins, but they are limited to some basic object features and the definition of the geometry with coordinate indices of `IndexedFaceSet` for the compound objects.

By providing the most comprehensive coverage of 3D modeling terms in OWL 2 with DL-based formal grounding, the 3D Modeling Ontology is the most suitable ontology to date in the 3D modeling domain for interlinking 3D models and model segments with Linked Data. Structured 3D model representations correspond to RDF graphs which, when interlinked with Linked Open Data resource identifiers, naturally merge to the Linked Open Data Cloud. The structured representations obtained this way can be queried and updated from diverse data sources manually and programmatically using SPARQL (e.g., [178]). For example, SPARQL makes it possible to find heart models in a 3D organ model repository that are opaque and have a high enough level of detail to 3D print for discussing the surgical decisions before an operation, such as the ones with more than 50,000 vertices, and order them alphabetically, as demonstrated in Fig. 7.2.

---

[22] ftp://ftp.igd.fraunhofer.de/outgoing/irbuild/InstantExport/



Subject	Predicate	Object
<http://example.com/3dmodels/heart/>	rdf:type	snomedct:80891009
<http://example.com/3dmodels/heart/>	t3dmo:createdIn	t3dmo:AutoDesk3dsMax
<http://example.com/3dmodels/heart/>	t3dmo:baseForm	t3dmo:prolateSpheroid
<http://example.com/3dmodels/heart/>	t3dmo:hasFaces	"177454"^^xsd:nonNegativeInteger
<http://example.com/3dmodels/heart/>	t3dmo:hasEdges	"532362"^^xsd:nonNegativeInteger
<http://example.com/3dmodels/heart/>	t3dmo:hasVertices	"92026"^^xsd:nonNegativeInteger
<http://example.com/3dmodels/heart/>	t3dmo:transparency	"0.000"^^xsd:decimal
<http://example.com/3dmodels/heart/>	rdfs:label	"Heart anatomy model"^^xsd:string

SPARQL Query

```
PREFIX t3dmo: <http://3dontology.org/> .
PREFIX snomedct: <http://purl.bioontology.org/ontology/SNOMEDCT/> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .

SELECT DISTINCT ?model ?vertices ?transparency
FROM <http://example.com/sparql>
WHERE {
  ?model a snomedct:80891009 ; t3dmo:hasVertices ?vertices ;
   t3dmo:transparency ?transparency .
  FILTER (?vertices > "50000"^^xsd:nonNegativeInteger) .
  FILTER (?transparency = "0.000"^^xsd:decimal) .
}  ORDER BY ?model LIMIT 10
```

Answer

```
http://example.com/3dmodels/heart/

http://example.com/3dmodels/heartwitharteries/
```

Figure 7.2 Using a SPARQL query to retrieve medically accurate heart models from a 3D model repository to be 3D printed for preoperative surgical discussions

As demonstrated above, SPARQL queries allow multiple parameters to be combined into a single query, as opposed to keyword-based traditional search, with which this is not feasible to the same extent.

In this case study, precision and recall are used in terms of a set of retrieved concepts and a set of relevant concepts. In this context, precision is the ratio of the number of correctly captured concepts and the sum of correctly captured concepts and incorrect concepts associated with the scene. Recall is the ratio of the number of correctly captured concepts and the sum of correctly captured concepts and depicted concepts that have not been captured. For example, considering 38 concepts of interest and 32 false positives, the precision of keyword-based search was 38 / (38 + 32) = 0.54. Table 7.3 summarizes the corresponding precision and recall values of traditional keyword-based search and SPARQL queries using terms of three previous 3D ontologies and the proposed 3D ontology for characteristics-based 3D model retrieval.

Table 7.3 Precision and recall with keywords, and terms from previous 3D ontologies and the 3D Modeling Ontology

	Precision	Recall
Keywords	0.54	0.40
OntologyX3D	0.75	0.72
Gilson-Silva Mapping	0.69	0.66
Mapping of Petit et al.	0.80	0.77
3DMO	0.91	0.87

The semantic descriptions utilizing the 3D Modeling Ontology can be stored in Linked Open Data datasets, graph databases, such as triplestores or quadstores, and lightweight semantics in the website markup as RDFa, HTML5 Microdata, or JSON-LD, similar to the terms of other OWL ontologies.

Those structured knowledge representations that implement the concepts and roles of the 3D Modeling Ontology are suitable for not only 3D model indexing and retrieval, but also for reasoning over the represented models, for example, to measure material thickness based on the declared coordinates, or to compare two 3D models by volume.

7.3.1.6 Case Study 2: Content-Based Video Retrieval using 3D Model Characteristics

The second case study involved content-based video retrieval with hidden 3D features. Accurate and detailed medical 3D animations and their spatiotemporal representation can visualize specific details, functions, and development of human organs and complex scientific concepts for medical training [179]. 3D surgery animations enable the exploration of surgical procedures and decision planning before operation. However, the retrieval of 3D animations demonstrating a particular disease, instrument, or procedure is not straightforward. By describing the 3D characteristics of the models animated in these videos using the 3D Modeling Ontology, together with spatiotemporal annotation of video segments and their regions of interest, the associated semantics can be efficiently captured and indexed.[23]

In this case study, cardiology animations have been used to compare the concept coverage of previous X3D-based ontologies to that of the 3D Modeling Ontology. The video structure has been described using the core reference ontology VidOnt, the video scenes have been annotated with timestamps using the SWRL Temporal Ontology, the regions of interest as moving regions with spatiotemporal segmentation using MPEG-7 terms and Media Fragment URIs (see Figure 7.3), and the 3D models depicted in the scenes with previous X3D-based ontologies and the 3D Modeling Ontology.

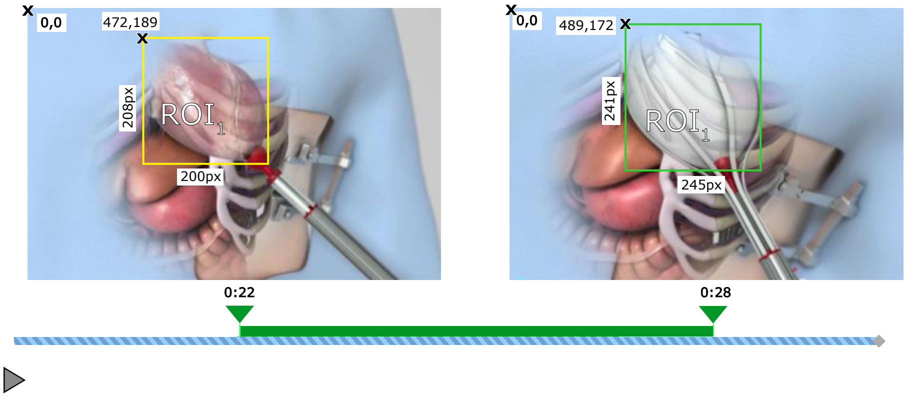


Figure 7.3 The spatiotemporal annotation of 3D models as moving regions of interest (ROIs) in medical animations can be associated with hidden 3D characteristics described using X3D terms. Animation by Mardil Medical[24]

Once annotated, complex reasoning tasks have been performed on the represented 3D models of the video scenes using the RDFS entailment rules [180], the Ter Horst reasoning rules [181], and the OWL reasoning rules [182], and by employing the hypertableau calculus [183].

The results have been mapped to Linked Data definitions, such as `dbpedia:Mitral_valve_replacement` and `dbpedia:Ventricle_(heart)`, which can be utilized in intelligent video recom-

---

[23] Sikos, L. F. (2017) 3D model indexing in videos for content-based retrieval via X3D-based semantic enrichment and automated reasoning. *Invited paper*. In: 22nd International Conference on 3D Web Technology. Brisbane, QLD, Australia, 5–7 June 2017. New York: ACM. doi: 10.1145/3055624.3075943

[24] https://vimeo.com/86452375

mendations for medical training and hypervideo applications that display rich semantics about the represented models during playback.[25]

The comparisons were done by enumerating 50 concepts (including body parts (e.g., rib, heart), procedures (e.g., injection, surgery), medical concepts (e.g., implant, investigational device, ventricular reshaping device), and high-level concepts (e.g., Tendyne Mitral Valve Implant, Mardil Ven-Touch), and 3D properties (e.g., number of vertices, edges, and faces, diffuse color, specular color, transparency, and shininess) of interest, which formed the set of real positives. Precision values were calculated by dividing the number of captured concepts and properties with the sum of 50 and the number of incorrectly identified concepts. Recall values were calculated as the ratio of true positives and the sum of true positives and the number of concepts that could not be captured using the ontology. Table 7.4 demonstrates the performance difference between the terms of OntologyX3D and the 3D Modeling Ontology when retrieving 3D models with predetermined 3D characteristics from animations about the Tendyne mitral valve implant[26] and the Mardil Ventouch ventricular reshaping device.[27]

Table 7.4 Performance comparison of OntologyX3D and 3DMO in 3D model retrieval from computer animations

Model	OntologyX3D		3DMO	
	Precision	Recall	Precision	Recall
Tendyne	0.75	0.70	0.96	0.92
Mardil Ventouch	0.68	0.63	0.90	0.82

Label- or tag-based retrieval of these models would rely on classifications, which would not be suitable for retrieving those heart models from a repository that are, for example, detailed enough to be 3D printed for a particular application.

The results indicate that the 3D Modeling Ontology provides a more comprehensive coverage of 3D modeling terms than its predecessors, such as OntologyX3D. Note, however, that the semantic

---

[25] Sikos, L. F. (2018) Ontology-based structured video annotation for content-based video retrieval via spatiotemporal reasoning. *Invited chapter*. In: Kwaśnicka, H., Jain, L. C. (eds.) Bridging the semantic gap in image and video analysis. Intelligent Systems Reference Library, vol. 145, pp. 97–122. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-73891-8_6

[26] A product of Tendyne Holdings

[27] A trademark of Mardil Medical

enrichment of videos with structured annotations about the represented 3D space and its features using terms from the 3D Modeling Ontology is best used in combination with domain ontologies and common sense knowledge bases, similar to every other upper ontology.[28] The 3D Modeling Ontology can also provide the formally described background knowledge to complement other content-based indexing approaches, such as the ones that utilize the 3D Extended Histogram of Oriented Gradients (3DHOG) Descriptor [184], the 3D Distance Field Descriptor (3D DFD), bag of visual words (BOVW), or 3D co-occurrence matrices [185].

7.3.2 Reasoning over 3D Information Represented in Video Scenes

The full potential of description logic-based ontologies cannot be exploited simply by asserting facts directly to a knowledge base or LOD dataset. Deriving new information via reasoning enables intelligent multimedia tasks, including automated 3D scene interpretation so that software agents can process the geometry and 3D features of the represented 3D space.[29] These intelligent tasks, such as measuring the thickness of the wall of an object or computing the volume of the represented object, would not be possible without rich semantics. The logical underpinning of the 3D Modeling Ontology ensures that software agents can interpret the 3D concepts and roles used in formal descriptions, such as the ones declared in the previous section.

In the context of computer animations, spatiotemporal reasoning can be performed over the semantically enriched 3D models. For example, a video event such as the first incision of a surgery can be formally described using temporal description logics, such as $\mathcal{TL}$-$\mathcal{F}$, or rules-based mechanisms, such as SWRL rules, and automatically recognized via reasoning. This can be combined with the structured annotation of the 3D models used in the scene to obtain a comprehensive coverage of concepts and interlink them with Linked Data definitions. This enables, among others, intelligent

---

[28] Sikos, L. F. (2017) Utilizing multimedia ontologies in video scene interpretation via information fusion and automated reasoning. In: 2017 Federated Conference on Computer Science and Information Systems, pp. 91–98. Prague, Czech Republic, 3–6 September 2017. New York: IEEE. doi: 10.15439/978-83-946253-7-5

[29] Sikos, L. F. (2017) A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain. *Invited paper*. International Journal of Metadata, Semantics and Ontologies, 12(1):59–70. doi: 10.1504/IJMSO.2017.10008658

video recommendation engines and hypervideo applications that utilize rich semantics about the represented models.

7.3.3 3D Content Retrieval from Video Scenes

The most common 3D model retrieval methods include shape-based (shape distribution [186], spherical harmonic function [187], frequency domain feature, statistical characteristics), topological graph-based, and global or local geometry-based methods [188]. Some of these methods are applicable to a particular shape model, such as mesh or volume model, while others to any 3D model. They are different in terms of efficiency, discriminative power, multi-scale support, partial matching support, robustness, and pose normalization.

The knowledge representation of 3D models with the 3D Modeling Ontology makes it possible to execute advanced queries manually or programmatically with SPARQL. For example, multiple parameters can be combined into a single query, such as the following: find 100 opaque 3D models with more than 50,000 vertices, and order them alphabetically by name (see Listing 7.3).

Listing 7.3 Advanced querying with SPARQL

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX t3dmo: <http://3dontology.org/> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .
SELECT DISTINCT ?model_name ?vertices ?transparency
FROM <http://example.com/sparql>
WHERE {
  ?model a t3dmo:3dModel ;
  rdfs:label ?model_name ;
  t3dmo:hasVertices ?vertices ;
  t3dmo:transparency ?transparency .
  FILTER (?vertices > "50000"^^xsd:nonNegativeInteger) .
  FILTER (?transparency = "0.000"^^xsd:decimal) .
}
ORDER BY ?model_name LIMIT 100
```

Note that this type of querying is not possible on traditional keyword-based 3D model repositories, and currently this is the only way to retrieve 3D models by their actual characteristics rather than by their textual descriptors that might be too general, misspelt, or biased by the opinion or preference of the creator.

The previous 3D model indexing method has been evaluated using a hundred randomly selected medical 3D models. The models have been indexed and retrieved using ontologies of implementations found in the literature, such as OntologyX3D, the Gilson-Silva mapping of the X3D vocabulary, and the Kinect Ontology. Those 3D ontologies that are known from the literature yet do not have an online ontology file anymore, such as the Geometrical Application Ontology and the Common Shape Ontology, have not been considered. The evaluation indicates a significant increase in terms of both precision and recall for 3D model representations when using the presented method with the 3D Modeling Ontology (see Table 7.5).

Table 7.5 Precision and recall with and without 3D Modeling Ontology terms

	Precision	Recall
OntologyX3D	0.79	0.51
Gilson-Silva mapping	0.68	0.62
Kinect Ontology	0.14	0.09
3DMO	0.83	0.86

The main reasons behind the favorable results can be summarized as follows. Earlier 3D ontologies, both the ones with and the ones without X3D alignment, have a number of ontology engineering issues. These include overly specific or generic concept definitions or important concepts missing from the concept hierarchy (e.g., the Kinect Ontology defines the specific Kinect3DModel, but no 3D models in general), suboptimal or missing domain and range declarations (e.g., OntologyX3D has a `vertices` datatype property, but its range is defined as `xsd:string`, `hasCoordinate` has the range `Coordinate`, which is defined as a class with no semantics), low expressivity (e.g., $\mathcal{AL}$ in the case of the Gilson-Silva X3D mapping, $\mathcal{ALUIN}^{+(\mathcal{D})}$ in the case of OntologyX3D, $\mathcal{ALCRIF}^{(\mathcal{D})}$ in the case of the Kinect Ontology), and lack of important 3D modeling terms used in the industry (many of which are defined in the 3D Modeling Ontology). Regarding the level of semantic enrichment achieved for 3D models in computer animations, to date only the 3D Modeling Ontology is able to incorporate 3D model indexing with spatiotemporal annotations in 3D scene interpretation.

7.3.4 Summary

In this chapter, two novel approaches have been demonstrated for the interpretation of videos: reasoning over temporal DL axioms, and exploiting hidden 3D information from moving regions of interest in videos.

Using the hybrid description logic-based formalism introduced in this thesis for the semantic annotation of videos is unique in a way that the representation does not require additional vocabularies and implementation standards, and it is currently the only formalism that can be used for this purpose at the abstraction level.

The presented approach for the semantic enrichment of 3D models in videos opens new directions in the feature-based retrieval of 3D models. The formal representation of, and reasoning over, 3D models of CGI and CAD animations enables efficient content-based retrieval. The 3D Modeling Ontology, a state-of-the-art X3D-based OWL 2 ontology, enables interlinking concepts of 3D models and their segments with the LOD Cloud, including the ones appearing in computer animations that can be annotated for spatiotemporal reasoning. As a result, 3D models can be indexed and retrieved based on their 3D characteristics, such as size, shape, and material, which provide semantics for the vertices, edges, and faces of 3D objects, such as cylinders, cones, spheres, and polygons. When used in multimedia indexing, videos that feature a particular 3D object can be retrieved efficiently. Empirical results indicate that this approach outperforms traditional keyword-based, metadata-based, and other ontology-based 3D model retrieval techniques.

Chapter 8
Summary and Conclusions

> "Necessity is the mother
> of all invention."
>
> — *Albert Einstein*

To bridge the semantic gap, and more specifically, to address the limitations of previous approaches to the automated interpretation of medical videos, novel ontologies and methodologies have been introduced, aligned with international standards, and complemented by best practices for ontology engineering. While the primary focus of these contributions is the medical domain, the implementation potential has been tested in a wide range of knowledge domains, such as engineering and cultural heritage, and application areas, such as content-based video retrieval and feature-based 3D model retrieval.

8.1 Original Contributions

The thorough literature review conducted at an initial phase of candidature indicated that the importance of video semantics is understated compared to image semantics, a far more researched area. After identifying serious yet common design issues of multimedia ontologies, best practices have been introduced and applied during the development, evaluation, and maintenance of novel ontologies. These feature a high DL expressivity, favorable reasoning complexity, and best-in-class reasoning power. The corresponding original contributions can be summarized as follows:

1. After identifying the lack of standardization, poor conceptualization, and low reasoning potential of multimedia ontologies,[1] a novel core reference ontology, the *Video Ontology (VidOnt)*, has been introduced for video representation and reasoning, acting as a mediator between de facto standard and standard upper and domain ontologies.[2]
    a. In contrast to previous multimedia ontologies, the proposed ontology has been formally grounded in $\mathcal{SROIQ}^{(\mathcal{D})}$, one of the most expressive decidable description logics. By exploiting all the available mathematical constructors of the description logic underlying the implementation language, OWL 2, the reasoning power of the ontology exceeds the reasoning potential of all previously released multimedia ontologies.[3]
    b. The proposed ontology contributes to narrowing the semantic gap between automatically extractable low-level audio and video features and high-level semantics.
    c. In parallel with MPEG-7-alignment, the ontology was designed with semantic concept mapping to Linked Data in mind, based on a comprehensive review of ontology-based video annotation software.[4] The applicability of the proposed ontology in ontology-based structured video annotation has been confirmed via content-based video retrieval implementations and spatiotemporal reasoning.[5]
2. The *MPEG-7 Ontology*, the first formally grounded and complete OWL 2 implementation of the ISO standard ISO/IEC 15938 (MPEG-7), has been created for the semantic enrichment of

---

[1] Sikos, L. F., Powers, D. M. W. (2015) Knowledge-driven video information retrieval with LOD: from semi-structured to structured video metadata. In: Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, pp. 35–37. Melbourne, VIC, Australia, 23 October 2015. New York: ACM. doi: 10.1145/2810133.2810141

[2] Sikos, L. F. (2018) VidOnt: a core reference ontology for reasoning over video scenes. *Invited paper*. Journal of Information and Telecommunication. doi: 10.1080/24751839.2018.1437696

[3] Sikos, L. F. (2016) A novel approach to multimedia ontology engineering for automated reasoning over audiovisual LOD datasets. In: Nguyễn, N. T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) Intelligent information and database systems, pp. 3–12. 8th Asian Conference on Intelligent Information and Database Systems, Đà Nẵng, Vietnam, 14–16 March 2016. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-49381-6_1

[4] Sikos, L. F. (2016) RDF-powered semantic video annotation tools with concept mapping to Linked Data for next-generation video indexing. Multimedia Tools and Applications, 76(12):14437–14460. doi: 10.1007/s11042-016-3705-7

[5] Sikos, L. F. (2018) Ontology-based structured video annotation for content-based video retrieval via spatiotemporal reasoning. *Invited chapter*. In: Kwaśnicka, H., Jain, L. C. (eds.) Bridging the semantic gap in image and video analysis. Intelligent Systems Reference Library, vol. 145, pp. 97–122. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-73891-8_6

multimedia resources, including the formal knowledge representation of videos and video scenes.

   a. The proposed ontology is the one and only MPEG-7 ontology to date grounded in a description logic formalism, yielding to a correct conceptual model and well-understood computational properties.
   b. In contrast to other ontologies that cover only the upper MDS, visual, or audio descriptors of MPEG-7, the proposed ontology uniquely covers the entire vocabulary of the MPEG-7 standard, and exposes all the MPEG-7 terms through a permanent URI.
   c. Previous MPEG-7 ontologies inherited issues from the standard because of the reliance of the mappings on the XSD file of MPEG-7, and many defined some terms of the specification rather than the corresponding concepts. The proposed ontology considers the semantics of the concepts defined in the standard, and correctly models the corresponding classes, properties, relations, and taxonomical structure.

3. The *3D Modeling Ontology (3DMO)*, a state-of-the-art ontology, has been proposed for ontology-based 3D scene and model indexing aligned with ISO/IEC 19775, 19776, and 19777.[6]
   a. The proposed ontology is the first and only 3D ontology formally grounded in $\mathcal{SROIQ}^{(\mathcal{D})}$, one of the most expressive decidable description logics.[7]
   b. The proposed ontology is the first ontology that covers the entire X3D 3.3 vocabulary, complemented by important terms from the 3D modeling industry.
   c. The proposed ontology has been complemented by a method for content-based video retrieval via exploiting hidden 3D features.[8]

---

[6] Sikos, L. F. (2016) Rich semantics for interactive 3D models of cultural artifacts. In: Garoufallou, E., Subirats, I., Stellato, A., Greenberg, J. (eds.) Metadata and semantics research, pp. 169–180. 10th International Conference on Metadata and Semantics Research, Göttingen, Germany, 22–25 November 2016. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-49157-8_14

[7] Sikos, L. F. (2017) A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain. *Invited paper*. International Journal of Metadata, Semantics and Ontologies, 12(1):59–70. doi: 10.1504/IJMSO.2017.10008658

[8] Sikos, L. F. (2017) 3D model indexing in videos for content-based retrieval via X3D-based semantic enrichment and automated reasoning. *Invited paper*. In: 22nd International Conference on 3D Web Technology. Brisbane, QLD, Australia, 5–7 June 2017. New York: ACM. doi: 10.1145/3055624.3075943

4. While rule-based formalisms are known to be efficient in describing video events, most of them are not decidable and have implementation issues. For these reasons, a very expressive yet decidable spatiotemporal formalism has been proposed for structured video annotation and content-based video retrieval. Since this formalism can answer mixed SPARQL queries over general, spatial, temporal, and fuzzy relation types, it is an advancement compared to former approaches, for which formalizing complex video events by combining primitive actions has been feasible only for well-defined knowledge domains with constrained actions and environment.[9]
    a. The proposed hybrid description logic employs general, spatial, temporal, and fuzzy axioms for video scene representation and automated reasoning-based scene interpretation, while achieving a favorable tradeoff between expressivity and reasoning complexity.
    b. The spatial part of the proposed formalism is RCC8-compliant, and supports role negation at the ABox level, making it possible to check whether the RCC relations hold or infer the disconnectivity of regions.
    c. The temporal part of the proposed formalism implements a fragment of the interval-based Halpern-Shoham logic, supports existential temporal quantifiers only, and universal temporal quantifiers are not permitted to retain decidability.
    d. The proposed formalism supports the fuzzification of axioms to be able to express the certainty or uncertainty of concept and event depiction.
5. Based on the comprehensive review of the state of the art, an approach has been proposed to determine the list of DL-based multimedia ontologies to be used for the annotation of video scenes while taking into account all major aspects of ontology implementation, including the intended semantics, the quality of conceptualization, specificity, DL expressivity, standards

---

[9] Sikos, L. F. (2017) Spatiotemporal reasoning for complex video event recognition in content-based video retrieval. In: Hassanien, A., Shaalan, K., Gaber, T., Tolba, M. (eds.) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017. Advances in intelligent systems and computing, vol. 639, pp. 704–713. 3rd International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 9–11 September 2017. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-64861-3_66

alignment, namespace and documentation stability, spatiotemporal annotation support, and annotation support for uncertainty.[10]

8.2 Publications of the Ph.D. Candidature

- Sikos, L. F., Powers, D. M. W. (2015) Knowledge-driven video information retrieval with LOD: from semi-structured to structured video metadata. In: Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, pp. 35–37. Melbourne, VIC, Australia, 23 October 2015. New York: ACM. doi: 10.1145/2810133.2810141
- Sikos, L. F. (2016) A novel approach to multimedia ontology engineering for automated reasoning over audiovisual LOD datasets. In: Nguyễn, N. T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) Intelligent information and database systems, pp. 3–12. 8th Asian Conference on Intelligent Information and Database Systems, Đà Nẵng, Vietnam, 14–16 March 2016. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-49381-6_1
- Sikos, L. F. (2016) RDF-powered semantic video annotation tools with concept mapping to Linked Data for next-generation video indexing.[11] Multimedia Tools and Applications, 76(12):14437–14460. doi: 10.1007/s11042-016-3705-7
- Sikos, L. F. (2016) Rich semantics for interactive 3D models of cultural artifacts. In: Garoufallou, E., Subirats, I., Stellato, A., Greenberg, J. (eds.) Metadata and semantics research, pp. 169–180. 10th International Conference on Metadata and Semantics Research, Göttingen, Germany, 22–25 November 2016. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-49157-8_14
- Sikos, L. F. (2017) Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5

---

[10] Sikos, L. F. (2017) Utilizing multimedia ontologies in video scene interpretation via information fusion and automated reasoning. In: 2017 Federated Conference on Computer Science and Information Systems, pp. 91–98. Prague, Czech Republic, 3–6 September 2017. New York: IEEE. doi: 10.15439/978-83-946253-7-5

[11] A substantial part of this publication has been revised, extended, and tailored to the primary knowledge domain of the thesis in chapter 5.

- Sikos, L. F. (2017) 3D model indexing in videos for content-based retrieval via X3D-based semantic enrichment and automated reasoning. *Invited paper*. In: 22nd International Conference on 3D Web Technology. Brisbane, QLD, Australia, 5–7 June 2017. New York: ACM. doi: 10.1145/3055624.3075943
- Sikos, L. F. (2017) A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain.[12] *Invited paper*. International Journal of Metadata, Semantics and Ontologies, 12(1):59–70. doi: 10.1504/IJMSO. 2017.10008658
- Sikos, L. F. (2017) Spatiotemporal reasoning for complex video event recognition in content-based video retrieval. In: Hassanien, A., Shaalan, K., Gaber, T., Tolba, M. (eds.) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017. Advances in intelligent systems and computing, vol. 639, pp. 704–713. 3rd International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 9–11 September 2017. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-64861-3_66
- Sikos, L. F. (2017) Utilizing multimedia ontologies in video scene interpretation via information fusion and automated reasoning. In: 2017 Federated Conference on Computer Science and Information Systems, pp. 91–98. Prague, Czech Republic, 3–6 September 2017. New York: IEEE. doi: 10.15439/978-83-946253-7-5
- Sikos, L. F. (2018) VidOnt: a core reference ontology for reasoning over video scenes. *Invited paper*. Journal of Information and Telecommunication. doi: 10.1080/24751839.2018. 1437696
- Sikos, L. F. (2018) Ontology-based structured video annotation for content-based video retrieval via spatiotemporal reasoning. *Invited chapter*. In: Kwaśnicka, H., Jain, L. C. (eds.) Bridging the semantic gap in image and video analysis. Intelligent Systems Reference Library, vol. 145, pp. 97–122. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-73891-8_6

---

[12] A substantial part of this publication has been incorporated in Section 7.3.1.

- Sikos, L. F. (2018) A spatiotemporal knowledge representation formalism for intelligent applications: from medical video analysis to action recognition. In: Hassanien, A. E. et al. (eds.) Machine learning paradigms: theory and applications. Studies in computational intelligence. Cham, Switzerland: Springer

Wherever results from these publications are incorporated in the text, a reference is given in the footnotes, as opposed to those from publications of the author published prior to the candidature, which are listed under References, such as these former publications that influenced this Ph.D. research:

- Sikos, L. F. (2011) Advanced (X)HTML5 metadata and semantics for Web 3.0 videos. DESIDOC Journal of Library and Information Technology, 31(4):247–252. doi: 10.14429/djlit.31.4.1105
- Sikos, L. F. (2014) Web standards: mastering HTML5, CSS3, and XML (Second edition). New York: Apress. ISBN-10: 1484208846, ISBN-13: 978-1-484208-84-7, doi: 10.1007/978-1-4842-0883-0
- Sikos, L. F. (2015) Mastering structured data on the Semantic Web: from HTML5 Microdata to Linked Open Data. New York: Apress. ISBN 978-1-4842-1050-5, doi: 10.1007/978-1-4842-1049-9

8.3 Further Results

The research output attracted attention from both academia and the industry, and resulted in two invited papers[13,14] and an invited conference talk.[15] Also, I was invited to be a journal reviewer (Com-

---

[13] Sikos, L. F. (2017) A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain. *Invited paper*. International Journal of Metadata, Semantics and Ontologies, 12(1):59–70. doi: 10.1504/IJMSO.2017.10008658

[14] Sikos, L. F. (2018) VidOnt: a core reference ontology for reasoning over video scenes. *Invited paper*. Journal of Information and Telecommunication. doi: 10.1080/24751839.2018.1437696

[15] Sikos, L. F. (2017) 3D model indexing in videos for content-based retrieval via X3D-based semantic enrichment and automated reasoning. Invited paper. In: 22nd International Conference on 3D Web Technology. Brisbane, QLD, Australia, 5–7 June 2017. New York: ACM. doi: 10.1145/3055624.3075943

puting and Informatics, Logic and Logical Philosophy), a program committee member of six conferences (ACM, IEEE, Springer), a session chair of two conferences (ACM, IEEE), the author of a book chapter,[16] and the author of a prestigious reference book chapter.[17]

Ongoing discussions with the Web3D Consortium, the developer of X3D, indicated mutual interest for a potential collaboration to develop the next version of the 3D Modeling Ontology based on the upcoming X3D 4.0, with improved conceptual modeling and role hierarchy, provenance data for X3D-X3DOM term distinction, and X3D versioning for each class and property.

Using my expertise in multimedia reasoning, I published a monograph, which covers knowledge engineering in the multimedia domain, spatiotemporal annotation and reasoning, and ontology engineering best practices.[18]

The ontologies introduced in this thesis feature permanent namespace URLs on purl.org and dedicated websites with implementation examples at http://3dontology.org, http://mpeg7.org, and http://videoontology.org.

8.4 Academic Community Involvement

Editorial Activities

- Volume Editor
  Intelligent Systems Reference Library (Springer)

Reviewer Activities

- Computing and Informatics, from 2016
- Logic and Logical Philosophy, from 2018

---

[16] Sikos, L. F. (2018) A spatiotemporal knowledge representation formalism for intelligent applications: from medical video analysis to action recognition. In: Hassanien, A. E. et al. (eds.) Machine learning paradigms: theory and applications. Studies in computational intelligence. Cham, Switzerland: Springer (to appear)

[17] Sikos, L. F. (2018) Ontology-based structured video annotation for content-based video retrieval via spatiotemporal reasoning. *Invited chapter*. In: Kwaśnicka, H., Jain, L. C. (eds.) Bridging the semantic gap in image and video analysis. Intelligent Systems Reference Library, vol. 145, pp. 97–122. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-73891-8_6

[18] Sikos, L. F. (2017) Description logics in multimedia reasoning. Cham, Switzerland: Springer. ISBN 978-3-319-54065-8, doi: 10.1007/978-3-319-54066-5

Conference Organization Activities

2017

- Session Chair
  22nd International Conference on 3D Web Technology (ACM)
- Program Committee Member
  3rd International Workshop on Artificial Intelligence in Machine Vision and Graphics (IEEE)
- Program Committee Member
  12th International Symposium on Advances in Artificial Intelligence and Applications (IEEE)
- Session Chair
  12th International Symposium on Advances in Artificial Intelligence and Applications (IEEE)

2018

- Program Committee Member
  3rd International Conference on Advanced Machine Learning Technologies and Applications (Springer)
- Program Committee Member
  23rd International Conference on 3D Web Technology (ACM)
- Program Committee Member
  4th International Workshop on Artificial Intelligence in Machine Vision and Graphics (IEEE)
- Program Committee Member
  13th International Symposium on Advances in Artificial Intelligence and Applications (IEEE)

Invited Talk

- Invited Speaker
  22nd International Conference on 3D Web Technology (ACM), Brisbane, Australia, 2017

Author Invitations

- Studies in Computational Intelligence (Springer), 2018
- Journal of Information and Telecommunication (Taylor & Francis), 2017
- International Journal of Metadata, Semantics and Ontologies (Inderscience), 2017
- Intelligent Systems Reference Library (Springer), 2016

Memberships and Affiliations

- Association for Automated Reasoning (AAR), 2017–present
- Institute of Electrical and Electronics Engineers (IEEE), 2017–present
- Association for Computing Machinery (ACM), 2017–present

8.5 Summary

This chapter detailed the research output of the candidature, including primary innovation and the corresponding practical implementations. It described the international standards covered by these contributions, and other research outputs, such as ontology websites that provide a standardized namespace and full documentation for the ontologies introduced during the Ph.D.

Index

Appendix

A.1 The MPEG-7 Ontology (MPEG7)

Type: upper ontology

Standard coverage: ISO/IEC 15938

Website: http://mpeg7.org

Namespace: `http://purl.org/ontology/mpeg7/`

Prefix: `mpeg7`

```
@prefix mpeg7: <http://purl.org/ontology/mpeg7/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@base <http://purl.org/ontology/mpeg7/> .

### Provenance ###

<http://purl.org/ontology/mpeg7/mpeg7.ttl> a owl:Ontology ; dc:creator
dbpedia:Leslie_Sikos ; dc:description "The MPEG-7 Ontology is an OWL 2 ontology
covering the vocabulary of the MPEG-7 international standard (ISO/IEC 15938)."@en ;
dc:title "MPEG-7 Ontology"@en .
```

```
### Concepts ###

mpeg7:AudioObject a owl:Class .
mpeg7:AudioVisualMaterial a owl:Class .
mpeg7:Author a owl:Class .
mpeg7:Box a owl:Class .
mpeg7:Camera a owl:Class .
mpeg7:Character a owl:Class .
mpeg7:Component a owl:Class .
mpeg7:Database a owl:Class .
mpeg7:Descriptor a owl:Class .
mpeg7:Episode a owl:Class .
mpeg7:Event a owl:Class .
mpeg7:Face a owl:Class .
mpeg7:Film a owl:Class .
  mpeg7:Rush a owl:Class ; owl:subClassOf mpeg7:Film .
mpeg7:Frame a owl:Class .
mpeg7:Graphic a owl:Class .
mpeg7:Image a owl:Class .
mpeg7:Instrument a owl:Class .
mpeg7:KeyFrame a owl:Class .
mpeg7:Locality a owl:Class .
mpeg7:Medium a owl:Class .
mpeg7:Mesh a owl:Class .
mpeg7:Mix a owl:Class .
mpeg7:Model a owl:Class .
mpeg7:MultimediaContent a owl:Class .
mpeg7:Object a owl:Class .
mpeg7:Owner a owl:Class .
mpeg7:User a owl:Class .
```

```
  mpeg7:Organization a owl:Class ; owl:subClassOf mpeg7:Owner , mpeg7:User .
  mpeg7:Person a owl:Class ; owl:subClassOf mpeg7:Owner , mpeg7:User .
mpeg7:Polygon a owl:Class .
mpeg7:Place a owl:Class .
mpeg7:Program a owl:Class .
  mpeg7:Promotion a owl:Class ; owl:subClassOf mpeg7:Program .
mpeg7:Recorder a owl:Class .
mpeg7:Scene a owl:Class .
mpeg7:Shot a owl:Class .
mpeg7:SoundEffect a owl:Class .
mpeg7:Source a owl:Class .
mpeg7:Speech a owl:Class .
mpeg7:Story a owl:Class .
mpeg7:Stream a owl:Class .
mpeg7:TechnicalStaff a owl:Class .
mpeg7:Terminal a owl:Class .
mpeg7:TrademarkImage a owl:Class .
mpeg7:TransitionEffect a owl:Class .
mpeg7:Video a owl:Class .
mpeg7:VisualObject a owl:Class .

### Roles ###


mpeg7:SpatialDecomposition a owl:ObjectProperty , owl:TransitiveProperty .
mpeg7:SpatioTemporalDecomposition a owl:ObjectProperty .
mpeg7:StillRegionSpatialDecomposition a owl:ObjectProperty .
mpeg7:TemporalDecomposition a owl:ObjectProperty .
mpeg7:SpatioTemporalDecomposition owl:propertyChainAxiom (mpeg7:TemporalDecomposition
mpeg7:SpatialDecomposition) .
mpeg7:abbrev a owl:DatatypeProperty ; rdfs:range xsd:string .
```

```
mpeg7:Abstract a rdf:Property ; rdfs:range mpeg7:TextAnnotationType .
mpeg7:accidental a rdf:Property ; rdfs:range mpeg7:degreeAccidentalType .
mpeg7:acousticScore a owl:DatatypeProperty ; rdfs:range mpeg7:nonNegativeReal .
mpeg7:Address a rdf:Property ; rdfs:range mpeg7:PlaceType .
mpeg7:AddressLine a rdf:Property ; rdfs:range mpeg7:TextualType .
mpeg7:AdministrativeUnit a rdf:Property .
mpeg7:Affiliation a rdf:Property .
mpeg7:Agent a rdf:Property ; rdfs:range mpeg7:AgentType .
mpeg7:aggregation a owl:DatatypeProperty .
mpeg7:altitude a owl:DatatypeProperty ; rdfs:range xsd:double .
mpeg7:AmountOfMotion a rdf:Property ; rdfs:range mpeg7:MixtureAmountOfMotionType ,
mpeg7:NonMixtureAmountOfMotionType .
mpeg7:aspectRatio a owl:DatatypeProperty ; rdfs:range mpeg7:nonNegativeReal .
mpeg7:AstronomicalBody a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:audio a rdf:Property .
mpeg7:AudioChannels a rdf:Property .
mpeg7:AudioCoding a rdf:Property .
mpeg7:AudioDefects a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:AudioVisual a rdf:Property ; rdfs:range mpeg7:AudioVisualSegmentType .
mpeg7:authority a owl:DatatypeProperty ; rdfs:range xsd:NMTOKEN .
mpeg7:Availability a rdf:Property ; rdfs:range mpeg7:AvailabilityType .
mpeg7:AvailabilityPeriod a rdf:Property .
mpeg7:Average a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned8 .
mpeg7:BackgroundNoiseLevel a rdf:Property ; rdfs:range mpeg7:BackgroundNoiseLevelType .
mpeg7:Balance a rdf:Property ; rdfs:range mpeg7:BalanceType .
mpeg7:Bandwidth a rdf:Property ; rdfs:range mpeg7:BandwidthType .
mpeg7:Beat a rdf:Property ; rdfs:range mpeg7:beatType .
mpeg7:best a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:BinCounts a rdf:Property .
mpeg7:BitRate a rdf:Property .
```

```
mpeg7:bitsPer a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:bitsPerBin a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned4 .
mpeg7:block a rdf:Property ; rdfs:range mpeg7:unsigned16 .
mpeg7:BoomDown a owl:DatatypeProperty .
mpeg7:BoomUp a owl:DatatypeProperty .
mpeg7:BPM a rdf:Property ; rdfs:range mpeg7:AudioBPMType .
mpeg7:BroadcastReady a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:BrowsingCategory a rdf:Property .
mpeg7:BuildingName a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:cameraFollows a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:CaptionLanguage a rdf:Property .
mpeg7:CbACCoeff14 a owl:DatatypeProperty .
mpeg7:CbACCoeff2 a owl:DatatypeProperty .
mpeg7:CbACCoeff20 a owl:DatatypeProperty .
mpeg7:CbACCoeff27 a owl:DatatypeProperty .
mpeg7:CbACCoeff5 a owl:DatatypeProperty .
mpeg7:CbACCoeff63 a owl:DatatypeProperty .
mpeg7:CbACCoeff9 a owl:DatatypeProperty .
mpeg7:CbDCCoeff a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned6 .
mpeg7:CentralFourierFeature a rdf:Property .
mpeg7:ChannelNo a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:channels a rdf:Property ; rdfs:range mpeg7:integerVector .
mpeg7:Citizenship a owl:DatatypeProperty ; rdfs:range mpeg7:countryCode .
mpeg7:City a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:Classification a rdf:Property ; rdfs:range mpeg7:ClassificationType .
mpeg7:closed a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:Coeff a owl:DatatypeProperty ; rdfs:range mpeg7:integerVector .
mpeg7:colorDomain a rdf:Property .
mpeg7:ColorLayout a rdf:Property ; rdfs:range mpeg7:ColorLayoutType .
mpeg7:colorQuant a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned3 .
```

```
mpeg7:ColorQuantization a rdf:Property ; rdfs:range mpeg7:ColorQuantizationType .
mpeg7:colorReferenceFlag a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:ColorSampling a rdf:Property ; rdfs:range mpeg7:ColorSamplingType .
mpeg7:ColorSpace a rdf:Property ; rdfs:range mpeg7:ColorSpaceType .
mpeg7:ColorStructure a rdf:Property ; rdfs:range mpeg7:ColorStructureType .
mpeg7:ColorTemperature a rdf:Property ; rdfs:range mpeg7:ColorTemperatureType .
mpeg7:ColorTransMat a rdf:Property .
mpeg7:ColorVariance a rdf:Property .
mpeg7:Comment a owl:DatatypeProperty ; rdfs:range mpeg7:TextAnnotationType .
mpeg7:ComponentMediaProfile a rdf:Property ; rdfs:range mpeg7:MediaProfileType .
mpeg7:CompositeFeature a rdf:Property .
mpeg7:confidence a owl:DatatypeProperty ; rdfs:range mpeg7:zeroToOneType .
mpeg7:ConfusionInfo a rdf:Property ; rdfs:range mpeg7:ConfusionCountType .
mpeg7:confusionInfoRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:Content a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:Contour a rdf:Property ; rdfs:range mpeg7:ContourShapeType , mpeg7:contourType .
mpeg7:CoordDef a rdf:Property .
mpeg7:CoordRef a rdf:Property .
mpeg7:Coords a rdf:Property ; rdfs:domain mpeg7:Polygon ; rdfs:range
mpeg7:IntegerMatrixType .
mpeg7:CopyrightString a rdf:Property ; rdfs:range mpeg7:TextualType .
mpeg7:Country a owl:DatatypeProperty ; rdfs:range mpeg7:countryCode .
mpeg7:Covariance a rdf:Property ; rdfs:range mpeg7:FloatMatrixType .
mpeg7:CrACCoeff14 a owl:DatatypeProperty .
mpeg7:CrACCoeff2 a owl:DatatypeProperty .
mpeg7:CrACCoeff20 a owl:DatatypeProperty .
mpeg7:CrACCoeff27 a owl:DatatypeProperty .
mpeg7:CrACCoeff5 a owl:DatatypeProperty .
mpeg7:CrACCoeff63 a owl:DatatypeProperty .
mpeg7:CrACCoeff9 a owl:DatatypeProperty .
```

```
mpeg7:CrDCCoeff a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned6 .
mpeg7:Creation a rdf:Property ; rdfs:range mpeg7:CreationType .
mpeg7:CreationCoordinates a rdf:Property .
mpeg7:CreationLocation a rdf:Property ; rdfs:range mpeg7:PlaceType .
mpeg7:CreationTime a rdf:Property ; rdfs:range mpeg7:timePointType .
mpeg7:CreationTool a rdf:Property ; rdfs:range mpeg7:CreationToolType .
mpeg7:Creator a rdf:Property ; rdfs:range mpeg7:CreatorType .
mpeg7:CrossChannelCorrelation a rdf:Property ; rdfs:range
mpeg7:CrossChannelCorrelationType .
mpeg7:Date a owl:DatatypeProperty ; rdfs:range mpeg7:TimeType .
mpeg7:dateFrom a rdf:Property ; rdfs:range mpeg7:timePointType .
mpeg7:dateTo a rdf:Property ; rdfs:range mpeg7:timePointType .
mpeg7:datum a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:DcOffset a rdf:Property ; rdfs:range mpeg7:DcOffsetType .
mpeg7:defaultLattice a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:defaultSpeakerInfoRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:Definition a rdf:Property ; rdfs:range mpeg7:TextualType .
mpeg7:Deletion a rdf:Property ; rdfs:range mpeg7:integerVector .
mpeg7:Denominator a rdf:Property .
mpeg7:Depth a rdf:Property ; rdfs:range mpeg7:TemporalInterpolationType .
mpeg7:Description a rdf:Property ; rdfs:range mpeg7:CompleteDescriptionType .
mpeg7:DescriptionMetadata a rdf:Property ; rdfs:range mpeg7:DescriptionMetadataType .
mpeg7:descriptionMetadataRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:DescriptionProfile a rdf:Property ; rdfs:range mpeg7:DescriptionProfileType .
mpeg7:descriptorMask a owl:DatatypeProperty .
mpeg7:DetectionProcess a owl:DatatypeProperty .
mpeg7:dim a rdf:Property ; rdfs:range mpeg7:listOfPositiveIntegerForDim . # ISO/IEC
15938-2
mpeg7:directed a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:Direction a owl:DatatypeProperty .
```

```
mpeg7:display a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:Dissemination a rdf:Property ; rdfs:range mpeg7:DisseminationType .
mpeg7:DollyBackward a owl:DatatypeProperty .
mpeg7:DollyForward a owl:DatatypeProperty .
mpeg7:DominantColor a rdf:Property ; rdfs:range mpeg7:DominantColorType .
mpeg7:DominantDirection a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned3 .
mpeg7:Duration a owl:DatatypeProperty ; rdfs:range mpeg7:durationType .
mpeg7:DynamicShapeVariation a rdf:Property .
mpeg7:Edge a rdf:Property ; rdfs:range mpeg7:EdgeHistogramType .
mpeg7:ElectronicAddress a rdf:Property ; rdfs:range mpeg7:ElectronicAddressType .
mpeg7:ellipseFlag a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:Email a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:encoding a rdf:Property .
mpeg7:Energy a owl:DatatypeProperty ; rdfs:range mpeg7:textureListType .
mpeg7:EnergyDeviation a owl:DatatypeProperty ; rdfs:range mpeg7:textureListType .
mpeg7:EntityIdentifier a rdf:Property ; rdfs:range mpeg7:UniqueIDType .
mpeg7:ErrorClass a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:ErrorEvent a rdf:Property ; rdfs:range mpeg7:ErrorEventType .
mpeg7:ErrorEventList a rdf:Property .
mpeg7:FamilyName a rdf:Property ; rdfs:range mpeg7:NameComponentType .
mpeg7:Fax a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:Feature a rdf:Property .
mpeg7:Field a rdf:Property .
mpeg7:FigureTrajectory a rdf:Property ; rdfs:range mpeg7:FigureTrajectoryType .
mpeg7:FileFormat a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:FileSize a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:First a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:Fixed a owl:DatatypeProperty .
mpeg7:Flatness a rdf:Property ; rdfs:range mpeg7:AudioSpectrumFlatnessType .
mpeg7:FocusOfExpansion a rdf:Property ; rdfs:range mpeg7:FocusOfExpansionType .
```

```
mpeg7:Form a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:Format a rdf:Property .
mpeg7:FourierFeature a rdf:Property .
mpeg7:FractionalPresence a rdf:Property ; rdfs:range mpeg7:FractionalPresenceType .
mpeg7:FreeTextAnnotation a rdf:Property ; rdfs:range mpeg7:TextualType .
mpeg7:front a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:Genre a rdf:Property .
mpeg7:GeographicPosition a rdf:Property .
mpeg7:GivenName a rdf:Property ; rdfs:range mpeg7:NameComponentType .
mpeg7:GlobalCurvature a owl:DatatypeProperty ; rdfs:range mpeg7:curvatureType .
mpeg7:HarmonicRatio a rdf:Property ; rdfs:range mpeg7:AudioLLDScalarType .
mpeg7:HarmonicSpectralCentroid a rdf:Property ; rdfs:range
mpeg7:HarmonicSpectralCentroidType .
mpeg7:HarmonicSpectralDeviation a rdf:Property ; rdfs:range
mpeg7:HarmonicSpectralDeviationType .
mpeg7:HarmonicSpectralSpread a rdf:Property ; rdfs:range
mpeg7:HarmonicSpectralSpreadType .
mpeg7:HarmonicSpectralVariation a rdf:Property ; rdfs:range
mpeg7:HarmonicSpectralVariationType .
mpeg7:Header a rdf:Property ; rdfs:range mpeg7:HeaderType .
mpeg7:height a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:hiEdge a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:HighestPeakY a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned7 .
mpeg7:hiLimit a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:HomogeneousPattern a rdf:Property ; rdfs:range mpeg7:HomogeneousTextureType .
mpeg7:hopSize a rdf:Property ; rdfs:range mpeg7:mediaDurationType .
mpeg7:HorizontalPosition a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:How a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:href a rdf:Property ; rdfs:range mpeg7:termReferenceType .
mpeg7:id a owl:DatatypeProperty ; rdfs:range mpeg7:ID .
```

```
mpeg7:IlluminationCompensatedColor a rdf:Property ; rdfs:range
mpeg7:IlluminationInvariantColorType .
mpeg7:Index a rdf:Property .
mpeg7:IndexEntry a rdf:Property ; rdfs:range mpeg7:SpokenContentIndexEntryType .
mpeg7:initial a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:InitialRegion a rdf:Property ; rdfs:range mpeg7:RegionLocatorType .
mpeg7:Insertion a rdf:Property ; rdfs:range mpeg7:integerVector .
mpeg7:InstanceIdentifier a rdf:Property ; rdfs:range mpeg7:UniqueIDType .
mpeg7:InstanceRef a rdf:Property ; rdfs:range mpeg7:ReferenceType .
mpeg7:IntegratedCoordinateSystem a rdf:Property .
mpeg7:Intensity a owl:DatatypeProperty .
mpeg7:InternalCoordinates a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:InterpolationFunctions a rdf:Property .
mpeg7:Interval a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned32 .
mpeg7:IsOriginalMono a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:key a rdf:Property .
mpeg7:KeyNote a rdf:Property .
mpeg7:KeyTimePoint a rdf:Property .
mpeg7:KeyValue a rdf:Property .
mpeg7:Keyword a rdf:Property .
mpeg7:KeywordAnnotation a rdf:Property ; rdfs:range mpeg7:KeywordAnnotationType .
mpeg7:Kind a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:Language a rdf:Property ; rdfs:range mpeg7:ExtendedLanguageType .
mpeg7:Last a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:LastUpdate a rdf:Property ; rdfs:range mpeg7:timePointType .
mpeg7:latitude a rdf:Property .
mpeg7:lattice a rdf:Property ; rdfs:range xsd:anyURI .
mpeg7:lfe a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:linguisticUnit a rdf:Property .
mpeg7:LinkingName a rdf:Property ; rdfs:range mpeg7:NameComponentType .
```

```
mpeg7:LocalCoordinateSystem a rdf:Property .
mpeg7:Location a rdf:Property ; rdfs:range mpeg7:PlaceType .
mpeg7:loEdge a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:LogAttackTime a rdf:Property ; rdfs:range mpeg7:LogAttackTimeType .
mpeg7:loLimit a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:longitude a rdf:Property .
mpeg7:Lyric a owl:DatatypeProperty ; rdfs:range mpeg7:TextualType .
mpeg7:MagnitudeOfART a rdf:Property .
mpeg7:MaterialType a mpeg7:TermUseType .
mpeg7:Max a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:maximum a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:maxRange a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:MaxSqDist a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:Mean a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:MediaDuration a rdf:Property ; rdfs:range mpeg7:MediaIncrDurationType .
mpeg7:MediaFormat a rdf:Property ; rdfs:range mpeg7:MediaFormatType .
mpeg7:MediaIdentification a rdf:Property ; rdfs:range mpeg7:MediaIdentificationType .
mpeg7:MediaIncrDuration a rdf:Property ; rdfs:range mpeg7:MediaIncrDurationType .
mpeg7:MediaInstance a rdf:Property ; rdfs:range mpeg7:MediaInstanceType .
mpeg7:MediaLocator a rdf:Property ; rdfs:range mpeg7:MediaLocatorType .
mpeg7:MediaProfile a rdf:Property ; rdfs:range mpeg7:MediaProfileType .
mpeg7:MediaQuality a rdf:Property ; rdfs:range mpeg7:MediaQualityType .
mpeg7:MediaRelIncrTimePoint a rdf:Property ;
 rdfs:range mpeg7:MediaRelIncrTimePointType .
mpeg7:MediaTime a rdf:Property ; rdfs:range mpeg7:MediaTimeType .
mpeg7:mediaTimeBase a rdf:Property ; rdfs:range mpeg7:xPathRefType .
mpeg7:mediaTimeUnit a rdf:Property ; rdfs:range mpeg7:mediaDurationType .
mpeg7:MediaUri a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:MelodyContour a rdf:Property ; rdfs:range mpeg7:MelodyContourType .
mpeg7:MelodySequence a rdf:Property ; rdfs:range mpeg7:MelodySequenceType .
```

```
mpeg7:Member a rdf:Property ; rdfs:range mpeg7:PersonType .
mpeg7:Meter a rdf:Property ; rdfs:range mpeg7:MeterType .
mpeg7:Min a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:minDuration a rdf:Property ; rdfs:range mpeg7:mediaDurationType .
mpeg7:minDurationRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:minimum a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:minRange a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:mode a rdf:Property ; rdfs:range mpeg7:termReferenceType .
mpeg7:MotionActivity a rdf:Property ; rdfs:range mpeg7:MotionActivityType .
mpeg7:motionModel a owl:DatatypeProperty .
mpeg7:MotionParams a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:MotionTrajectory a rdf:Property ; rdfs:range mpeg7:MotionTrajectoryType .
mpeg7:Name a rdf:Property ; rdfs:range mpeg7:PersonNameType , mpeg7:TextualType .
mpeg7:NameTerm a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:Nationality a owl:DatatypeProperty ; rdfs:range mpeg7:countryCode .
mpeg7:node a rdf:Property . # ; rdfs:range mpeg7:unsigned16 .
mpeg7:nodeOffset a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned16 .
mpeg7:NoteArray a rdf:Property .
mpeg7:NoteRelDuration a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:num a rdf:Property ; rdfs:range mpeg7:unsigned16 .
mpeg7:Numeration a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:Numerator a rdf:Property .
mpeg7:NumOfBins a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned12 .
mpeg7:numOfBitplanesDiscarded a owl:DatatypeProperty .
mpeg7:numOfCoeff a owl:DatatypeProperty .
mpeg7:numOfDimensions a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:numOfElements a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:numOfLongRuns a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned5 .
mpeg7:numOfMediumRuns a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned5 .
mpeg7:numOfOriginalEntries a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
```

```
mpeg7:numOfPartX a rdf:Property ; rdfs:range mpeg7:unsigned8 .
mpeg7:numOfPartY a rdf:Property ; rdfs:range mpeg7:unsigned8 .
mpeg7:numOfShortRuns a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned6 .
mpeg7:octaveResolution a rdf:Property .
mpeg7:offset a rdf:Property ; rdfs:range mpeg7:mediaDurationType .
mpeg7:organization a owl:DatatypeProperty ; rdfs:range xsd:NMTOKEN .
mpeg7:PanLeft a owl:DatatypeProperty .
mpeg7:PanRight a owl:DatatypeProperty .
mpeg7:Parameters a owl:DatatypeProperty .
mpeg7:ParameterTrajectory a rdf:Property ; rdfs:range mpeg7:ParameterTrajectoryType .
mpeg7:ParametricMotion a rdf:Property ; rdfs:range mpeg7:ParametricMotionType .
mpeg7:Params a rdf:Property ; rdfs:range mpeg7:TemporalInterpolationType .
mpeg7:Peak a rdf:Property .
mpeg7:peakX a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned6 .
mpeg7:peakY a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned3 .
mpeg7:Percentage a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned5 .
mpeg7:PerceptibleDefects a rdf:Property .
mpeg7:Period a rdf:Property .
mpeg7:PersonDescription a rdf:Property ; rdfs:range mpeg7:TextualType .
mpeg7:PersonGroup a rdf:Property ; rdfs:range mpeg7:PersonGroupType .
mpeg7:PersonType a rdf:Property .
mpeg7:phone a rdf:Property ; rdfs:range mpeg7:PhoneLexiconIndexType .
mpeg7:PhoneIndex a rdf:Property .
mpeg7:PhoneIndexEntry a rdf:Property .
mpeg7:PhoneLexicon a rdf:Property ; rdfs:range mpeg7:PhoneLexiconType .
mpeg7:phoneLexiconRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:PhoneLink a rdf:Property .
mpeg7:phoneticAlphabet a rdf:Property ; rdfs:range mpeg7:phoneticAlphabetType .
mpeg7:phoneticTranscription a rdf:Property .
mpeg7:PitchNote a rdf:Property .
```

```
mpeg7:Pixel a rdf:Property .
mpeg7:PlaceDescription a rdf:Property ; rdfs:range mpeg7:TextualType .
mpeg7:PlanarSurfaces a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned12 .
mpeg7:Point a rdf:Property ; rdfs:range mpeg7:GeographicPointType .
mpeg7:positionalOrder a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:PostalAddress a rdf:Property .
mpeg7:PostalTown a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:PostingIdentifier a owl:DatatypeProperty ; rdfs:range mpeg7:TextualType .
mpeg7:preferred a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:primary a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:PrivateIdentifier a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:probability a owl:DatatypeProperty ; rdfs:range mpeg7:zeroToOneType .
mpeg7:profileAndLevelIndication a rdf:Property .
mpeg7:PrototypeCurvature a owl:DatatypeProperty ; rdfs:range mpeg7:curvatureType .
mpeg7:provenance a rdf:Property .
mpeg7:PublicIdentifier a rdf:Property ; rdfs:range mpeg7:UniqueIDType .
mpeg7:Purpose a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:QualityRating a rdf:Property .
mpeg7:Random a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:rate a owl:DatatypeProperty ; rdfs:range mpeg7:nonNegativeReal .
mpeg7:RatingScheme a rdf:Property .
mpeg7:RatingSource a rdf:Property ; rdfs:range mpeg7:AgentType .
mpeg7:RatingValue a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:ratio a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:Raw a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:rear a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:ref a rdf:Property .
mpeg7:Region a rdf:Property ; rdfs:range mpeg7:regionCode .
mpeg7:Regularity a owl:DatatypeProperty .
mpeg7:RelatedMaterial a rdf:Property ; rdfs:range mpeg7:RelatedMaterialType .
```

```
mpeg7:RelativeDelay a rdf:Property ; rdfs:range mpeg7:RelativeDelayType .
mpeg7:Relevance a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned4 .
mpeg7:Repr a rdf:Property .
mpeg7:representation a rdf:Property .
mpeg7:RepresentativeFeature a rdf:Property ; rdfs:range mpeg7:GofGopFeatureType .
mpeg7:resolution a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:Rights a rdf:Property ; rdfs:range mpeg7:RightsType .
mpeg7:Role a rdf:Property ; rdfs:range mpeg7:TermUseType ,
 mpeg7:ControlledTermUseType .
mpeg7:RollAnticlockwise a owl:DatatypeProperty .
mpeg7:RollClockwise a owl:DatatypeProperty .
mpeg7:RoomName a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:RoomNumber a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:Salutation a rdf:Property ; rdfs:range mpeg7:NameComponentType .
mpeg7:Sample a rdf:Property .
mpeg7:ScalableColor a owl:DatatypeProperty ; rdfs:range mpeg7:ScalableColorType .
mpeg7:Scalar a rdf:Property ; rdfs:range xsd:float .
mpeg7:Scale a owl:DatatypeProperty ; rdfs:range mpeg7:scaleType .
mpeg7:Scaling a rdf:Property .
mpeg7:Segment a rdf:Property ; rdfs:range mpeg7:CameraMotionSegmentType .
mpeg7:SeriesOfScalar a rdf:Property .
mpeg7:SeriesOfVector a rdf:Property .
mpeg7:Setting a rdf:Property .
mpeg7:ShapeMask a rdf:Property ; rdfs:range mpeg7:RegionShapeType .
mpeg7:ShapeVariation a rdf:Property ; rdfs:range mpeg7:ShapeVariationType .
mpeg7:side a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:SignLanguage a rdf:Property .
mpeg7:SingularSurfaces a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned12 .
mpeg7:SpatialCoherency a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned5 .
mpeg7:SpatialDistributionParams a rdf:Property .
```

```
mpeg7:SpatialLocalizationParams a rdf:Property .
mpeg7:spatialRef a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:SpeakerInfo a rdf:Property ; rdfs:range mpeg7:SpeakerInfoType .
mpeg7:speakerInfoRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:SpectralCentroid a rdf:Property ; rdfs:range mpeg7:SpectralCentroidType .
mpeg7:Spectrum a rdf:Property .
mpeg7:SpokenLanguage a owl:DatatypeProperty ; rdfs:range xsd:language .
mpeg7:StandardDeviation a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned8 .
mpeg7:StartingFrequency a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:StartingNote a rdf:Property .
mpeg7:StartingPitch a rdf:Property .
mpeg7:StateProvinceCounty a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:StaticShapeVariation a rdf:Property .
mpeg7:StatisticalVariation a rdf:Property .
mpeg7:Status a owl:DatatypeProperty .
mpeg7:StreamID a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:StreetName a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:StreetNumber a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:strength a owl:DatatypeProperty ; rdfs:range mpeg7:zeroToOneType .
mpeg7:structure a rdf:Property .
mpeg7:StructuredAnnotation a rdf:Property ; rdfs:range mpeg7:StructuredAnnotationType .
mpeg7:StructuredInternalCoordinates a rdf:Property .
mpeg7:StructuredPostalAddress a rdf:Property .
mpeg7:style a rdf:Property .
mpeg7:Subject a rdf:Property ; rdfs:range mpeg7:TextAnnotationType .
mpeg7:SubRangeIndex a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned6 .
mpeg7:SubregionCompositeFeature a rdf:Property .
mpeg7:Substitution a rdf:Property ; rdfs:range mpeg7:IntegerMatrixType .
mpeg7:Summarization a rdf:Property ; rdfs:range mpeg7:SummarizationType .
mpeg7:supplemental a owl:DatatypeProperty ; rdfs:range xsd:boolean .
```

```
mpeg7:target a rdf:Property .
mpeg7:Telephone a rdf:Property .
mpeg7:TemporalCentroid a rdf:Property ; rdfs:range mpeg7:TemporalCentroidType .
mpeg7:temporalOrder a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:TemporalParams a rdf:Property .
mpeg7:TemporalTransition a rdf:Property ; rdfs:range mpeg7:VisualTimeSeriesType .
mpeg7:termID a owl:DatatypeProperty ; rdfs:range xsd:NMTOKEN .
mpeg7:TextureBrowsing a rdf:Property ; rdfs:range mpeg7:TextureBrowsingType .
mpeg7:TiltDown a owl:DatatypeProperty .
mpeg7:TiltUp a owl:DatatypeProperty .
mpeg7:TimeIncr a rdf:Property ; rdfs:range mpeg7:mediaDurationType ,
mpeg7:MediaIncrDurationType .
mpeg7:timeOffset a rdf:Property ; rdfs:range mpeg7:unsigned16 .
mpeg7:TimePoint a rdf:Property ; rdfs:range mpeg7:timePointType .
mpeg7:TimeStamp a rdf:Property ; rdfs:range mpeg7:MediaTimeType .
mpeg7:Title a rdf:Property ; rdfs:range mpeg7:NameComponentType ,
mpeg7:RelatedMaterialType .
mpeg7:TitleAudio a rdf:Property ; rdfs:range mpeg7:TemporalSegmentLocatorType .
mpeg7:TitleImage a rdf:Property ; rdfs:range mpeg7:ImageLocatorType .
mpeg7:TitleMedia a rdf:Property ; rdfs:range mpeg7:TitleMediaType .
mpeg7:TitleVideo a rdf:Property ; rdfs:range mpeg7:TemporalSegmentLocatorType .
mpeg7:Token a rdf:Property .
mpeg7:Tool a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:totalNumOfSamples a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:track a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:TrackLeft a owl:DatatypeProperty .
mpeg7:TrackRight a owl:DatatypeProperty .
mpeg7:translation a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:TransmissionTechnology a rdf:Property ; rdfs:range
mpeg7:TransmissionTechnologyType .
```

```
mpeg7:type a owl:DatatypeProperty ; rdfs:range xsd:NMTOKEN .
mpeg7:Unit a owl:DatatypeProperty .
mpeg7:units a owl:DatatypeProperty .
mpeg7:unlocatedRegion a owl:DatatypeProperty ; rdfs:domain mpeg7:Box , mpeg7:Polygon ;
rdfs:range xsd:boolean .
mpeg7:UpperLimitOfHarmonicity a rdf:Property ; rdfs:range mpeg7:AudioLLDScalarType .
mpeg7:Url a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:UsedTool a rdf:Property ; rdfs:range mpeg7:CreationToolType .
mpeg7:value a owl:DatatypeProperty ; rdfs:range xsd:string .
mpeg7:Values a rdf:Property .
mpeg7:variable a owl:DatatypeProperty ; rdfs:range xsd:boolean .
mpeg7:Variance a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:VarianceScalewise a rdf:Property ; rdfs:range mpeg7:FloatMatrixType .
mpeg7:VarianceSummed a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:Vector a owl:DatatypeProperty ; rdfs:range mpeg7:floatVector .
mpeg7:vectorSize a rdf:Property ; rdfs:range xsd:positiveInteger .
mpeg7:Vertex a rdf:Property ; rdfs:range mpeg7:TemporalInterpolationType .
mpeg7:VerticalPosition a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:VisualCoding a rdf:Property .
mpeg7:VisualDefects a rdf:Property ; rdfs:range mpeg7:ControlledTermUseType .
mpeg7:Weight a rdf:Property ; rdfs:range mpeg7:floatVector .
mpeg7:WhatAction a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:WhatObject a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:When a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:Where a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:Who a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:WholeInterval a rdf:Property .
mpeg7:Why a rdf:Property ; rdfs:range mpeg7:TermUseType .
mpeg7:width a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger .
mpeg7:word a rdf:Property ; rdfs:range mpeg7:WordLexiconIndexType .
```

```
mpeg7:WordIndex a rdf:Property .
mpeg7:WordIndexEntry a rdf:Property .
mpeg7:WordLexicon a rdf:Property ; rdfs:range mpeg7:WordLexiconType .
mpeg7:wordLexiconRef a owl:DatatypeProperty ; rdfs:range xsd:anyURI .
mpeg7:WordLink a rdf:Property .
mpeg7:worst a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:x a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned8 .
mpeg7:xOrigin a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:xpath a rdf:Property ; rdfs:range mpeg7:xPathRefType .
mpeg7:xRepr a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned8 .
mpeg7:xSrcSize a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:y a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned8 .
mpeg7:YACCoeff14 a owl:DatatypeProperty .
mpeg7:YACCoeff2 a owl:DatatypeProperty .
mpeg7:YACCoeff20 a owl:DatatypeProperty .
mpeg7:YACCoeff27 a owl:DatatypeProperty .
mpeg7:YACCoeff5 a owl:DatatypeProperty .
mpeg7:YACCoeff63 a owl:DatatypeProperty .
mpeg7:YACCoeff9 a owl:DatatypeProperty .
mpeg7:YDCCoeff a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned6 .
mpeg7:yOrigin a owl:DatatypeProperty ; rdfs:range xsd:float .
mpeg7:yRepr a owl:DatatypeProperty ; rdfs:range mpeg7:unsigned8 .
mpeg7:ySrcSize a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger .
mpeg7:ZoomIn a owl:DatatypeProperty .
mpeg7:ZoomOut a owl:DatatypeProperty .

### Datatypes ###

mpeg7:AdvancedFaceRecognitionType a rdfs:Datatype .
mpeg7:AffectiveType a rdfs:Datatype .
```

```
mpeg7:AgentObjectType a rdfs:Datatype .
mpeg7:AgentType a rdfs:Datatype .
mpeg7:AudioBPMType a rdfs:Datatype .
mpeg7:AudioDSType a rdfs:Datatype .
mpeg7:AudioDType a rdfs:Datatype .
mpeg7:AudioFundamentalFrequencyType a rdfs:Datatype .
mpeg7:AudioHarmonicityType a rdfs:Datatype .
mpeg7:AudioLLDScalarType a rdfs:Datatype . # ISO/IEC 15938-4 Audio
mpeg7:AudioLLDVectorType a rdfs:Datatype .
mpeg7:AudioPowerType a rdfs:Datatype .
mpeg7:AudioSegmentMediaSourceDecompositionType a rdfs:Datatype .
mpeg7:AudioSegmentTemporalDecompositionType a rdfs:Datatype .
mpeg7:AudioSegmentType a rdfs:Datatype .
mpeg7:AudioSignalQualityType a rdfs:Datatype .
mpeg7:AudioSignatureType a rdfs:Datatype .
mpeg7:AudioSpectrumBasisType a rdfs:Datatype .
mpeg7:AudioSpectrumCentroidType a rdfs:Datatype .
mpeg7:AudioSpectrumEnvelopeType a rdfs:Datatype .
mpeg7:AudioSpectrumFlatnessType a rdfs:Datatype .
mpeg7:AudioSpectrumProjectionType a rdfs:Datatype .
mpeg7:AudioSpectrumSpreadType a rdfs:Datatype .
mpeg7:AudioSummaryComponentType a rdfs:Datatype .
mpeg7:AudioTempoType a rdfs:Datatype .
mpeg7:AudioType a rdfs:Datatype .
mpeg7:AudioVisualSegmentMediaSourceDecompositionType a rdfs:Datatype .
mpeg7:AudioVisualSegmentTemporalDecompositionType a rdfs:Datatype .
mpeg7:AudioVisualSegmentType a rdfs:Datatype .
mpeg7:AudioVisualType a rdfs:Datatype .
mpeg7:AudioVisualType a rdfs:Datatype .
mpeg7:AudioWaveformType a rdfs:Datatype .
```

```
mpeg7:auxiliaryLanguageType a rdfs:Datatype .
mpeg7:AvailabilityType a rdfs:Datatype .
mpeg7:BackgroundNoiseLevelType a rdfs:Datatype .
mpeg7:BalanceType a rdfs:Datatype .
mpeg7:BandwidthType a rdfs:Datatype .
mpeg7:basicDurationType a rdfs:Datatype . # ISO/IEC 15938-2
mpeg7:basicTimePointType a rdfs:Datatype . # ISO/IEC 15938-2
mpeg7:beatType a rdfs:Datatype .
mpeg7:BoxListType a rdfs:Datatype .
mpeg7:BrowsingPreferencesType a rdfs:Datatype .
mpeg7:CameraMotionSegmentType a rdfs:Datatype .
mpeg7:CameraMotionType a rdfs:Datatype .
mpeg7:characterSetCode a rdfs:Datatype .
mpeg7:ClassificationPreferencesType a rdfs:Datatype .
mpeg7:ClassificationSchemeAliasType a rdfs:Datatype .
mpeg7:ClassificationSchemeBaseType a rdfs:Datatype .
mpeg7:ClassificationSchemeDescriptionType a rdfs:Datatype .
mpeg7:ClassificationSchemeType a rdfs:Datatype .
mpeg7:ClassificationType a rdfs:Datatype .
mpeg7:CollectionType a rdfs:Datatype .
mpeg7:ColorLayoutType a rdfs:Datatype .
mpeg7:ColorQuantizationType a rdfs:Datatype .
mpeg7:ColorSamplingType a rdfs:Datatype .
mpeg7:ColorSpaceType a rdfs:Datatype .
mpeg7:ColorStructureType a rdfs:Datatype .
mpeg7:ColorTemperatureType a rdfs:Datatype .
mpeg7:CompleteDescriptionType a rdfs:Datatype .
mpeg7:ConceptType a rdfs:Datatype .
mpeg7:ConfusionCountType a rdfs:Datatype .
mpeg7:ContentAbstractionType a rdfs:Datatype .
```

```
mpeg7:ContentCollectionType a rdfs:Datatype .
mpeg7:ContentDescriptionType a rdfs:Datatype .
mpeg7:ContentEntityType a rdfs:Datatype .
mpeg7:ContentManagementType a rdfs:Datatype .
mpeg7:ContourShapeType a rdfs:Datatype .
mpeg7:contourType a rdfs:Datatype .
mpeg7:ControlledTermUseType a rdfs:Datatype .
mpeg7:countryCode a rdfs:Datatype .
mpeg7:CreationDescriptionType a rdfs:Datatype .
mpeg7:CreationInformationType a rdfs:Datatype .
mpeg7:CreationPreferencesType a rdfs:Datatype .
mpeg7:CreationToolType a rdfs:Datatype .
mpeg7:CreationType a rdfs:Datatype .
mpeg7:CreatorType a rdfs:Datatype .
mpeg7:CrossChannelCorrelationType a rdfs:Datatype .
mpeg7:currencyCode a rdfs:Datatype .
mpeg7:curvatureType a rdfs:Datatype .
mpeg7:DcOffsetType a rdfs:Datatype .
mpeg7:degreeAccidentalType a rdfs:Datatype .
mpeg7:degreeNoteType a rdfs:Datatype .
mpeg7:dependencyOperatorType a rdfs:Datatype .
mpeg7:DependencyStructurePhraseType a rdfs:Datatype .
mpeg7:DependencyStructureType a rdfs:Datatype .
mpeg7:DescriptionMetadataType a rdfs:Datatype .
mpeg7:DescriptionProfileType a rdfs:Datatype .
mpeg7:DisseminationType a rdfs:Datatype .
mpeg7:DominantColorType a rdfs:Datatype .
mpeg7:DoubleDiagonalMatrixType a rdfs:Datatype .
mpeg7:DoubleMatrixType a rdfs:Datatype .
mpeg7:doubleVector a rdfs:Datatype .
```

```
mpeg7:DSType a rdfs:Datatype .
mpeg7:DType a rdfs:Datatype .
mpeg7:durationType a rdfs:Datatype .
mpeg7:EdgeHistogramType a rdfs:Datatype .
mpeg7:ElectronicAddressType a rdfs:Datatype .
mpeg7:ErrorEventType a rdfs:Datatype .
mpeg7:EventType a rdfs:Datatype .
mpeg7:ExtendedLanguageType a rdfs:Datatype .
mpeg7:ExtentType a rdfs:Datatype .
mpeg7:FaceRecognitionType a rdfs:Datatype .
mpeg7:FigureTrajectoryType a rdfs:Datatype .
mpeg7:FilteringAndSearchPreferencesType a rdfs:Datatype .
mpeg7:FinancialType a rdfs:Datatype .
mpeg7:FloatDiagonalMatrixType a rdfs:Datatype .
mpeg7:FloatMatrixType a rdfs:Datatype .
mpeg7:floatVector a rdfs:Datatype .
mpeg7:FocusOfExpansionType a rdfs:Datatype .
mpeg7:FractionalPresenceType a rdfs:Datatype .
mpeg7:GeographicPointType a rdfs:Datatype .
mpeg7:GoFGoPColorType a rdfs:Datatype .
mpeg7:GofGopFeatureType a rdfs:Datatype . # ISO/IEC 15938-4 Visual AMD/1
mpeg7:GraphType a rdfs:Datatype .
mpeg7:GridLayoutType a rdfs:Datatype .
mpeg7:HarmonicInstrumentTimbreType a rdfs:Datatype .
mpeg7:HarmonicSpectralCentroidType a rdfs:Datatype .
mpeg7:HarmonicSpectralDeviationType a rdfs:Datatype .
mpeg7:HarmonicSpectralSpreadType a rdfs:Datatype .
mpeg7:HarmonicSpectralVariationType a rdfs:Datatype .
mpeg7:HeaderType a rdfs:Datatype .
mpeg7:HierarchicalSummaryType a rdfs:Datatype .
```

```
mpeg7:HomogeneousTextureType a rdfs:Datatype .
mpeg7:IlluminationInvariantColorType a rdfs:Datatype .
mpeg7:ImageLocatorType a rdfs:Datatype .
mpeg7:ImageTextType a rdfs:Datatype .
mpeg7:ImageType a rdfs:Datatype .
mpeg7:IncrDurationType a rdfs:Datatype .
mpeg7:InlineMediaType a rdfs:Datatype .
mpeg7:InlineTermDefinitionType a rdfs:Datatype .
mpeg7:InstrumentTimbreType a rdfs:Datatype .
mpeg7:IntegerDiagonalMatrixType a rdfs:Datatype .
mpeg7:IntegerMatrixType a rdfs:Datatype .
mpeg7:integerVector a rdfs:Datatype .
mpeg7:IrregularVisualTimeSeriesType a rdfs:Datatype .
mpeg7:KeyType a rdfs:Datatype .
mpeg7:KeywordAnnotationType a rdfs:Datatype .
mpeg7:LexiconType a rdfs:Datatype .
mpeg7:listOfPositiveIntegerForDim a rdfs:Datatype ; rdfs:range xsd:positiveInteger . # ISO/IEC 15938-2
mpeg7:LogAttackTimeType a rdfs:Datatype .
mpeg7:MaskType a rdfs:Datatype .
mpeg7:MediaAgentType a rdfs:Datatype .
mpeg7:mediaDurationType a rdfs:Datatype .
mpeg7:MediaFormatType a rdfs:Datatype .
mpeg7:MediaIdentificationType a rdfs:Datatype .
mpeg7:MediaIncrDurationType a rdfs:Datatype .
mpeg7:MediaInformationType a rdfs:Datatype .
mpeg7:MediaInstanceType a rdfs:Datatype .
mpeg7:MediaLocatorType a rdfs:Datatype .
mpeg7:MediaProfileType a rdfs:Datatype .
mpeg7:MediaQualityType a rdfs:Datatype .
```

```
mpeg7:MediaRelIncrTimePointType a rdfs:Datatype .
mpeg7:MediaRelTimePointType a rdfs:Datatype .
mpeg7:MediaReviewType a rdfs:Datatype .
mpeg7:MediaSourceSegmentDecompositionType a rdfs:Datatype .
mpeg7:mediaTimeOffsetType a rdfs:Datatype .
mpeg7:mediaTimePointType a rdfs:Datatype .
mpeg7:MediaTimeType a rdfs:Datatype .
mpeg7:MediaTranscodingHintsType a rdfs:Datatype .
mpeg7:MelodyContourType a rdfs:Datatype .
mpeg7:MelodySequenceType a rdfs:Datatype .
mpeg7:MelodyType a rdfs:Datatype .
mpeg7:MeterType a rdfs:Datatype .
mpeg7:mimeType a rdfs:Datatype .
mpeg7:MinusOneToOneMatrixType a rdfs:Datatype .
mpeg7:minusOneToOneType a rdfs:Datatype .
mpeg7:minusOneToOneVector a rdfs:Datatype .
mpeg7:MixtureAmountOfMotionType a rdfs:Datatype .
mpeg7:MixtureCameraMotionSegmentType a rdfs:Datatype .
mpeg7:MotionActivityType a rdfs:Datatype .
mpeg7:MotionTrajectoryType a rdfs:Datatype .
mpeg7:MovingRegionFeatureType a rdfs:Datatype .
mpeg7:MovingRegionSpatioTemporalDecompositionType a rdfs:Datatype .
mpeg7:MovingRegionType a rdfs:Datatype .
mpeg7:Mpeg7BaseType a rdfs:Datatype . # ISO/IEC 15938-5 MDS
mpeg7:Mpeg7Type a rdfs:Datatype . # (COR/1)
mpeg7:MultimediaCollectionType a rdfs:Datatype .
mpeg7:MultimediaContentType a rdfs:Datatype .
mpeg7:NameComponentType a rdfs:Datatype .
mpeg7:NonDependencyStructurePhraseType a rdfs:Datatype .
mpeg7:NonMixtureAmountOfMotionType a rdfs:Datatype .
```

```
mpeg7:NonMixtureCameraMotionSegmentType a rdfs:Datatype .
mpeg7:nonNegativeReal a rdfs:Datatype .
mpeg7:ObjectType a rdfs:Datatype .
mpeg7:OrderingKeyType a rdfs:Datatype .
mpeg7:OrganizationType a rdfs:Datatype .
mpeg7:ParameterTrajectoryType a rdfs:Datatype .
mpeg7:ParametricMotionType a rdfs:Datatype .
mpeg7:ParentalGuidanceType a rdfs:Datatype .
mpeg7:PercussiveInstrumentTimbreType a rdfs:Datatype .
mpeg7:PersonGroupType a rdfs:Datatype .
mpeg7:PersonNameType a rdfs:Datatype . # AMD/2
mpeg7:PersonType a rdfs:Datatype .
mpeg7:PhoneLexiconIndexType a rdfs:Datatype .
mpeg7:PhoneLexiconType a rdfs:Datatype .
mpeg7:phoneticAlphabetType a rdfs:Datatype .
mpeg7:PhoneType a rdfs:Datatype .
mpeg7:PlaceType a rdfs:Datatype .
mpeg7:PointOfViewType a rdfs:Datatype .
mpeg7:PositionType a rdfs:Datatype .
mpeg7:PreferenceConditionType a rdfs:Datatype .
mpeg7:preferenceValueType a rdfs:Datatype .
mpeg7:ProbabilityMatrixType a rdfs:Datatype .
mpeg7:probabilityVector a rdfs:Datatype .
mpeg7:RatingType a rdfs:Datatype .
mpeg7:ReferenceType a rdfs:Datatype .
mpeg7:regionCode a rdfs:Datatype .
mpeg7:RegionLocatorType a rdfs:Datatype .
mpeg7:RegionShapeType a rdfs:Datatype .
mpeg7:RegularVisualTimeSeriesType a rdfs:Datatype .
mpeg7:RelatedMaterialType a rdfs:Datatype .
```

```
mpeg7:RelationType a rdfs:Datatype . # AMD/1
mpeg7:RelativeDelayType a rdfs:Datatype .
mpeg7:RelIncrTimePointType a rdfs:Datatype .
mpeg7:RelTimePointType a rdfs:Datatype .
mpeg7:RightsType a rdfs:Datatype .
mpeg7:ScalableColorType a rdfs:Datatype .
mpeg7:ScalableSeriesType a rdfs:Datatype .
mpeg7:scaleType a rdfs:Datatype .
mpeg7:SegmentDecompositionType a rdfs:Datatype .
mpeg7:SegmentType a rdfs:Datatype .
mpeg7:SemanticBagType a rdfs:Datatype .
mpeg7:SemanticBaseType a rdfs:Datatype .
mpeg7:SemanticPlaceType a rdfs:Datatype .
mpeg7:SemanticStateType a rdfs:Datatype .
mpeg7:SemanticTimeType a rdfs:Datatype .
mpeg7:SemanticType a rdfs:Datatype .
mpeg7:SequentialSummaryType a rdfs:Datatype .
mpeg7:SeriesOfScalarBinaryType a rdfs:Datatype .
mpeg7:SeriesOfScalarType a rdfs:Datatype .
mpeg7:SeriesOfVectorBinaryType a rdfs:Datatype .
mpeg7:SeriesOfVectorType a rdfs:Datatype .
mpeg7:Shape3DType a rdfs:Datatype .
mpeg7:ShapeVariationType a rdfs:Datatype .
mpeg7:SilenceHeaderType a rdfs:Datatype .
mpeg7:SilenceType a rdfs:Datatype .
mpeg7:SourcePreferencesType a rdfs:Datatype .
mpeg7:Spatial2DCoordinateSystemType a rdfs:Datatype .
mpeg7:SpatialSegmentDecompositionType a rdfs:Datatype .
mpeg7:SpatioTemporalLocatorType a rdfs:Datatype .
mpeg7:SpatioTemporalSegmentDecompositionType a rdfs:Datatype .
```

```
mpeg7:SpeakerInfoType a rdfs:Datatype .

mpeg7:SpectralCentroidType a rdfs:Datatype .

mpeg7:SpokenContentHeaderType a rdfs:Datatype .

mpeg7:SpokenContentIndexEntryType a rdfs:Datatype .

mpeg7:SpokenContentLatticeType a rdfs:Datatype .

mpeg7:SpokenContentLinkType a rdfs:Datatype .

mpeg7:StillRegionFeatureType a rdfs:Datatype .

mpeg7:StillRegionSpatialDecompositionType a rdfs:Datatype .

mpeg7:StillRegionType a rdfs:Datatype .

mpeg7:StructuredAnnotationType a rdfs:Datatype .

mpeg7:StructuredCollectionType a rdfs:Datatype .

mpeg7:SummarizationType a rdfs:Datatype .

mpeg7:summaryComponentType a rdfs:Datatype .

mpeg7:SummaryDescriptionType a rdfs:Datatype .

mpeg7:SummaryPreferencesType a rdfs:Datatype .

mpeg7:SummarySegmentGroupType a rdfs:Datatype .

mpeg7:SummarySegmentType a rdfs:Datatype .

mpeg7:SummaryType a rdfs:Datatype .

mpeg7:TemporalCentroidType a rdfs:Datatype .

mpeg7:TemporalInterpolationType a rdfs:Datatype .

mpeg7:TemporalMaskType a rdfs:Datatype .

mpeg7:TemporalSegmentDecompositionType a rdfs:Datatype .

mpeg7:TemporalSegmentLocatorType a rdfs:Datatype .

mpeg7:termAliasReferenceType a rdfs:Datatype .

mpeg7:TermDefinitionBaseType a rdfs:Datatype .

mpeg7:TermDefinitionType a rdfs:Datatype .

mpeg7:termReferenceListType a rdfs:Datatype .

mpeg7:termReferenceType a rdfs:Datatype .

mpeg7:termRelationQualifierType a rdfs:Datatype .

mpeg7:termURIReferenceType a rdfs:Datatype .
```

```
mpeg7:TermUseType a rdfs:Datatype .

mpeg7:TextAnnotationType a rdfs:Datatype .

mpeg7:TextualBaseType a rdfs:Datatype .

mpeg7:TextualType a rdfs:Datatype .

mpeg7:TextureBrowsingType a rdfs:Datatype .

mpeg7:textureListType a rdfs:Datatype .

mpeg7:timeOffsetType a rdfs:Datatype .

mpeg7:timePointType a rdfs:Datatype .

mpeg7:TimeType a rdfs:Datatype .

mpeg7:TitleMediaType a rdfs:Datatype .

mpeg7:TitleType a rdfs:Datatype .

mpeg7:TransmissionTechnologyType a rdfs:Datatype .

mpeg7:UniqueIDType a rdfs:Datatype .

mpeg7:unsigned1 a rdfs:Datatype .

mpeg7:unsigned10 a rdfs:Datatype .

mpeg7:unsigned11 a rdfs:Datatype .

mpeg7:unsigned12 a rdfs:Datatype .

mpeg7:unsigned13 a rdfs:Datatype .

mpeg7:unsigned14 a rdfs:Datatype .

mpeg7:unsigned15 a rdfs:Datatype .

mpeg7:unsigned16 a rdfs:Datatype .

mpeg7:unsigned17 a rdfs:Datatype .

mpeg7:unsigned18 a rdfs:Datatype .

mpeg7:unsigned19 a rdfs:Datatype .

mpeg7:unsigned2 a rdfs:Datatype .

mpeg7:unsigned20 a rdfs:Datatype .

mpeg7:unsigned21 a rdfs:Datatype .

mpeg7:unsigned22 a rdfs:Datatype .

mpeg7:unsigned23 a rdfs:Datatype .

mpeg7:unsigned24 a rdfs:Datatype .
```

```
mpeg7:unsigned25 a rdfs:Datatype .
mpeg7:unsigned26 a rdfs:Datatype .
mpeg7:unsigned27 a rdfs:Datatype .
mpeg7:unsigned28 a rdfs:Datatype .
mpeg7:unsigned29 a rdfs:Datatype .
mpeg7:unsigned3 a rdfs:Datatype .
mpeg7:unsigned30 a rdfs:Datatype .
mpeg7:unsigned31 a rdfs:Datatype .
mpeg7:unsigned32 a rdfs:Datatype .
mpeg7:unsigned4 a rdfs:Datatype .
mpeg7:unsigned5 a rdfs:Datatype .
mpeg7:unsigned6 a rdfs:Datatype .
mpeg7:unsigned7 a rdfs:Datatype .
mpeg7:unsigned8 a rdfs:Datatype .
mpeg7:unsigned9 a rdfs:Datatype .
mpeg7:UsageHistoryType a rdfs:Datatype .
mpeg7:UsageInformationType a rdfs:Datatype .
mpeg7:UsageRecordType a rdfs:Datatype .
mpeg7:UserActionHistoryType a rdfs:Datatype .
mpeg7:UserActionListType a rdfs:Datatype .
mpeg7:UserActionType a rdfs:Datatype .
mpeg7:userChoiceType a rdfs:Datatype .
mpeg7:UserDescriptionType a rdfs:Datatype .
mpeg7:UserIdentifierType a rdfs:Datatype .
mpeg7:UserPreferencesType a rdfs:Datatype .
mpeg7:VideoSegmentFeatureType a rdfs:Datatype .
mpeg7:VideoSegmentMediaSourceDecompositionType a rdfs:Datatype .
mpeg7:VideoSegmentSpatioTemporalDecompositionType a rdfs:Datatype .
mpeg7:VideoSegmentTemporalDecompositionType a rdfs:Datatype .
mpeg7:VideoSegmentType a rdfs:Datatype .
```

```
mpeg7:VideoTextType a rdfs:Datatype .
mpeg7:VideoType a rdfs:Datatype .
mpeg7:VisualDSType a rdfs:Datatype .
mpeg7:VisualDType a rdfs:Datatype .
mpeg7:VisualSummaryComponentType a rdfs:Datatype .
mpeg7:VisualTimeSeriesType a rdfs:Datatype .
mpeg7:WordFormType a rdfs:Datatype .
mpeg7:WordLexiconIndexType a rdfs:Datatype .
mpeg7:WordLexiconType a rdfs:Datatype . # Audio AMD/1
mpeg7:WordType a rdfs:Datatype .
mpeg7:xPathAbsoluteSelectorType a rdfs:Datatype .
mpeg7:xPathFieldType a rdfs:Datatype .
mpeg7:xPathRefType a rdfs:Datatype .
mpeg7:xPathSelectorType a rdfs:Datatype .
mpeg7:xPathType a rdfs:Datatype .
mpeg7:zeroToOneType a rdfs:Datatype .
```

A.2 The 3D Modeling Ontology (3DMO)

Type: upper ontology

Standard coverage: ISO/IEC 19775,
ISO/IEC 19776, ISO/IEC 19777

Website: http://3dontology.org

Namespace: `http://purl.org/ontology/x3d/`

Prefix: `t3dmo`

```
@prefix : <http://purl.org/ontology/x3d/> .
@prefix cc: <http://creativecommons.org/ns#> .
@prefix cito: <http://purl.org/spar/cito/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vidont: <http://vidont.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

### Provenance ###

<http://purl.org/ontology/x3d/3d.ttl> a owl:Ontology ; foaf:maker dbpedia:Leslie_Sikos
; cito:isReviewedBy dbpedia:Web3D_Consortium ; dc:description "3D Modeling Ontology
(3DMO) v1.1. A comprehensive 3D ontology with X3D 3.3 and X3DOM alignment."@en ;
dc:title "The 3D Modeling Ontology"@en .
```

```
### Concepts ###

:X3DField a owl:Class .
  :X3DArrayField a owl:Class ; rdfs:subClassOf :X3DField .
  :Uniform a owl:Class ; rdfs:subClassOf :X3DField .
:X3DNode a owl:Class ; dc:description "This abstract node type is the base type for all
nodes in the X3D system." .
  :GeoOrigin a owl:Class ; rdfs:subClassOf :X3DNode .
  :Param a owl:Class ; rdfs:subClassOf :X3DNode .
  :RigidBody a owl:Class ; rdfs:subClassOf :X3DNode .
  :ShaderPart a owl:Class ; rdfs:subClassOf :X3DNode .
  :TextureProperties a owl:Class ; rdfs:subClassOf :X3DNode .
    :X3DAppearanceChildNode a owl:Class ; rdfs:subClassOf :X3DNode .
    :BlendMode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
    :ColorMaskMode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
    :DepthMode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
    :LineProperties a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
    :X3DMaterialNode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
      :Material a owl:Class ; rdfs:subClassOf :X3DMaterialNode .
      :TwoSidedMaterial  a owl:Class ; rdfs:subClassOf :X3DMaterialNode .
    :X3DShaderNode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
      :CommonSurfaceShader a owl:Class ; rdfs:subClassOf :X3DShaderNode .
      :ComposedShader a owl:Class ; rdfs:subClassOf :X3DShaderNode .
    :X3DTextureNode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
      :MultiTexture a owl:Class ; rdfs:subClassOf :X3DTextureNode .
      :PixelTexture a owl:Class ; rdfs:subClassOf :X3DTextureNode .
      :RenderedTexture a owl:Class ; rdfs:subClassOf :X3DTextureNode .
        :RefinementTexture a owl:Class ; rdfs:subClassOf :RenderedTexture .
      :SurfaceShaderTexture a owl:Class ; rdfs:subClassOf :X3DTextureNode .
      :Texture a owl:Class ; rdfs:subClassOf :X3DTextureNode .
```

```
        :ImageTexture a owl:Class ; rdfs:subClassOf :Texture .
        :ImageTextureAtlas a owl:Class ; rdfs:subClassOf :Texture .
        :MovieTexture a owl:Class ; rdfs:subClassOf :Texture .
      :X3DEnvironmentTextureNode a owl:Class ; rdfs:subClassOf :X3DTextureNode .
        :ComposedCubeMapTexture a owl:Class ; rdfs:subClassOf
:X3DEnvironmentTextureNode .
        :GeneratedCubeMapTexture a owl:Class ; rdfs:subClassOf
:X3DEnvironmentTextureNode .
      :X3DTexture3DNode a owl:Class ; rdfs:subClassOf :X3DTextureNode .
    :X3DTextureTransformNode a owl:Class ; rdfs:subClassOf :X3DAppearanceChildNode .
      :TextureTransform a owl:Class ; rdfs:subClassOf :X3DTextureTransformNode .
      :TextureTransform3D a owl:Class ; rdfs:subClassOf :X3DTextureTransformNode .
      :TextureTransformMatrix3D a owl:Class ;
        rdfs:subClassOf :X3DTextureTransformNode .
    :X3DAppearanceNode a owl:Class ; rdfs:subClassOf :X3DNode .
    :Appearance a owl:Class ; rdfs:subClassOf :X3DAppearanceNode .
    :X3DChildNode a owl:Class ; rdfs:subClassOf :X3DNode .
    :ClipPlane a owl:Class ; rdfs:subClassOf :X3DChildNode .
    :CollisionCollection a owl:Class ; rdfs:subClassOf :X3DChildNode .
    :RigidBodyCollection a owl:Class ; rdfs:subClassOf :X3DChildNode .
    :X3DBindableNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
      :X3DBackgroundNode a owl:Class ; rdfs:subClassOf :X3DBindableNode .
        :Background a owl:Class ; rdfs:subClassOf :X3DBackgroundNode .
      :X3DEnvironmentNode a owl:Class ; rdfs:subClassOf :X3DBindableNode .
      :X3DFogNode a owl:Class ; rdfs:subClassOf :X3DBindableNode .
        :Fog a owl:Class ; rdfs:subClassOf :X3DFogNode .
      :X3DNavigationInfoNode a owl:Class ; rdfs:subClassOf :X3DBindableNode .
        :NavigationInfo a owl:Class ; rdfs:subClassOf :X3DNavigationInfoNode .
      :X3DViewpointNode a owl:Class ; rdfs:subClassOf :X3DBindableNode .
        :GeoViewpoint a owl:Class ; rdfs:subClassOf :X3DViewpointNode .
```

```
    :OrthoViewpoint a owl:Class ; rdfs:subClassOf :X3DViewpointNode .
    :Viewfrustum a owl:Class ; rdfs:subClassOf :X3DViewpointNode .
    :Viewpoint  a owl:Class ; rdfs:subClassOf :X3DViewpointNode .
:X3DBoundedObject a owl:Class ; rdfs:subClassOf :X3DChildNode .
  :X3DGroupingNode a owl:Class ; rdfs:subClassOf :X3DBoundedObject .
    :Anchor a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Billboard a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Block a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :CADAssembly a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :CADFace a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :CADLayer a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Collision a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :GeoTransform a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Group a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :HAnimSegment a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Inline a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
      :MultiPart a owl:Class ; rdfs:subClassOf :Inline .
    :RemoteSelectionGroup a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Scene a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :StaticGroup a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :Switch a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
    :X3DLODNode a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
      :DynamicLOD a owl:Class ; rdfs:subClassOf :X3DLODNode .
      :GeoLOD a owl:Class ; rdfs:subClassOf :X3DLODNode .
      :LOD a owl:Class ; rdfs:subClassOf :X3DLODNode .
    :X3DTransformNode a owl:Class ; rdfs:subClassOf :X3DGroupingNode .
      :GeoLocation a owl:Class ; rdfs:subClassOf :X3DTransformNode .
      :MatrixTransform a owl:Class ; rdfs:subClassOf :X3DTransformNode .
      :Transform a owl:Class ; rdfs:subClassOf :X3DTransformNode .
        :CADPart a owl:Class ; rdfs:subClassOf :Transform .
```

```
        :HAnimHumanoid a owl:Class ; rdfs:subClassOf :Transform .
        :HAnimJoint a owl:Class ; rdfs:subClassOf :Transform .
        :HAnimSite a owl:Class ; rdfs:subClassOf :Transform .
  :X3DNBodyCollidableNode a owl:Class ; rdfs:subClassOf :X3DBoundedObject .
    :CollidableShape a owl:Class ; rdfs:subClassOf :X3DNBodyCollidableNode .
  :X3DShapeNode a owl:Class ; rdfs:subClassOf :X3DBoundedObject .
    :Shape a owl:Class ; rdfs:subClassOf :X3DShapeNode .
    :X3DVolumeDataNode a owl:Class ; rdfs:subClassOf :X3DShapeNode .
      :IsoSurfaceVolumeData a owl:Class ; rdfs:subClassOf :X3DVolumeDataNode .
      :SegmentedVolumeData a owl:Class ; rdfs:subClassOf :X3DVolumeDataNode .
      :VolumeData a owl:Class ; rdfs:subClassOf :X3DVolumeDataNode .
:X3DFollowerNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
  :X3DChaserNode a owl:Class ; rdfs:subClassOf :X3DFollowerNode .
    :ColorChaser a owl:Class ; rdfs:subClassOf :X3DChaserNode .
    :OrientationChaser a owl:Class ; rdfs:subClassOf :X3DChaserNode .
    :PositionChaser a owl:Class ; rdfs:subClassOf :X3DChaserNode .
    :PositionChaser2D a owl:Class ; rdfs:subClassOf :X3DChaserNode .
    :ScalarChaser  a owl:Class ; rdfs:subClassOf :X3DChaserNode .
  :X3DDamperNode a owl:Class ; rdfs:subClassOf :X3DFollowerNode .
    :ColorDamper a owl:Class ; rdfs:subClassOf :X3DDamperNode .
    :CoordinateDamper a owl:Class ; rdfs:subClassOf :X3DDamperNode .
    :OrientationDamper a owl:Class ; rdfs:subClassOf :X3DDamperNode .
    :PositionDamper a owl:Class ; rdfs:subClassOf :X3DDamperNode .
    :PositionDamper2D a owl:Class ; rdfs:subClassOf :X3DDamperNode .
    :ScalarDamper a owl:Class ; rdfs:subClassOf :X3DDamperNode .
    :TexCoordDamper2D a owl:Class ; rdfs:subClassOf :X3DDamperNode .
:X3DInfoNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
  :GeoMetadata a owl:Class ; rdfs:subClassOf :X3DInfoNode .
  :WorldInfo a owl:Class ; rdfs:subClassOf :X3DInfoNode .
:X3DInterpolatorNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
```

```
    :ColorInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :CoordinateInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :GeoPositionInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :NormalInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :OrientationInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :PositionInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :ScalarInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
    :SplinePositionInterpolator a owl:Class ; rdfs:subClassOf :X3DInterpolatorNode .
  :X3DLightNode a owl:Class ; rdfs:subClassOf :X3DChildNode ;
    owl:disjointUnionOf ( :DirectionalLight :PointLight :SpotLight ) .
    :DirectionalLight a owl:Class ; rdfs:subClassOf :X3DLightNode .
    :PointLight a owl:Class ; rdfs:subClassOf :X3DLightNode .
    :SpotLight a owl:Class ; rdfs:subClassOf :X3DLightNode .
  :X3DSensorNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
    :CollisionSensor a owl:Class ; rdfs:subClassOf :X3DSensorNode .
    :TimeSensor a owl:Class ; rdfs:subClassOf :X3DSensorNode .
    :X3DPointingDeviceSensorNode a owl:Class ; rdfs:subClassOf :X3DSensorNode .
      :X3DDragSensorNode a owl:Class ; rdfs:subClassOf :X3DPointingDeviceSensorNode .
        :CylinderSensor a owl:Class ; rdfs:subClassOf :X3DDragSensorNode .
        :PlaneSensor a owl:Class ; rdfs:subClassOf :X3DDragSensorNode .
        :SphereSensor a owl:Class ; rdfs:subClassOf :X3DDragSensorNode .
        :TouchSensor a owl:Class ; rdfs:subClassOf :X3DDragSensorNode .
      :X3DTouchSensorNode a owl:Class ;
        rdfs:subClassOf :X3DPointingDeviceSensorNode .
  :X3DSoundNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
    :Sound a owl:Class ; rdfs:subClassOf :X3DSoundNode .
  :X3DTimeDependentNode a owl:Class ; rdfs:subClassOf :X3DChildNode .
    :X3DSoundSourceNode a owl:Class ; rdfs:subClassOf :X3DTimeDependentNode .
      :AudioClip a owl:Class ; rdfs:subClassOf :X3DSoundSourceNode .
  :X3DFontStyleNode a owl:Class ; rdfs:subClassOf :X3DNode .
```

```
:FontStyle a owl:Class ; rdfs:subClassOf :X3DFontStyleNode .
:X3DGeometricPropertyNode a owl:Class ; rdfs:subClassOf :X3DNode .
:HAnimDisplacer a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
:Normal a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
:PopGeometryLevel a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
:X3DColorNode a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
  :Color a owl:Class ; rdfs:subClassOf :X3DColorNode .
  :ColorRGBA a owl:Class ; rdfs:subClassOf :X3DColorNode .
:X3DCoordinateNode a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
  :Coordinate a owl:Class ; rdfs:subClassOf :X3DCoordinateNode .
  :GeoCoordinate a owl:Class ; rdfs:subClassOf :X3DCoordinateNode .
:X3DTextureCoordinateNode a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
  :MultiTextureCoordinate a owl:Class ; rdfs:subClassOf :X3DTextureCoordinateNode .
  :TextureCoordinate a owl:Class ; rdfs:subClassOf :X3DTextureCoordinateNode .
  :TextureCoordinate3D a owl:Class ; rdfs:subClassOf :X3DTextureCoordinateNode .
  :TextureCoordinateGenerator a owl:Class ;
   rdfs:subClassOf :X3DTextureCoordinateNode .
:X3DVertexAttributeNode a owl:Class ; rdfs:subClassOf :X3DGeometricPropertyNode .
  :FloatVertexAttribute a owl:Class ; rdfs:subClassOf :X3DVertexAttributeNode .
:X3DGeometryNode a owl:Class ; rdfs:subClassOf :X3DNode .
:ElevationGrid a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
:Extrusion a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
:GeoElevationGrid a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
:IndexedLineSet a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
:LineSet a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
:Mesh a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
:PointSet a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
  :ParticleSet a owl:Class ; rdfs:subClassOf :PointSet .
:SolidOfRevolution a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
  :IndexedFaceSet a owl:Class ; rdfs:subClassOf :SolidOfRevolution .
```

```
      :IndexedQuadSet a owl:Class ; rdfs:subClassOf :SolidOfRevolution .
      :IndexedTriangleSet a owl:Class ; rdfs:subClassOf :SolidOfRevolution .
      :IndexedTriangleStripSet a owl:Class ; rdfs:subClassOf :SolidOfRevolution .
      :QuadSet a owl:Class ; rdfs:subClassOf :SolidOfRevolution .
      :TriangleSet a owl:Class ; rdfs:subClassOf :SolidOfRevolution .
    :Text a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
    :X3DComposedGeometryNode a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
    :X3DPlanarGeometryNode a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
      :Arc2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :ArcClose2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :Circle2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :Disk2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :Polyline2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :Polypoint2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :Rectangle2D a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
      :TriangleSet2D  a owl:Class ; rdfs:subClassOf :X3DPlanarGeometryNode .
    :X3DSpatialGeometryNode  a owl:Class ; rdfs:subClassOf :X3DGeometryNode .
      :Box a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode ; owl:disjointWith
:Cone , :Cylinder , :Dish , :ExternalGeometry , :Nozzle , :Plane , :Pyramid ,
:RectangularTorus , :SlopedCylinder , :Snout , :Sphere , :SphereSegment , :Torus .
      :Cone a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Cylinder a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Dish a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :ExternalGeometry a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Nozzle a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Plane a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Pyramid a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :RectangularTorus a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :SlopedCylinder a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Snout a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
```

```
      :Sphere a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :SphereSegment a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :Torus a owl:Class ; rdfs:subClassOf :X3DSpatialGeometryNode .
      :X3DBinaryContainerGeometryNode a owl:Class ;
        rdfs:subClassOf :X3DSpatialGeometryNode .
        :BinaryGeometry a owl:Class ; rdfs:subClassOf :X3DBinaryContainerGeometryNode .
        :ImageGeometry a owl:Class ; rdfs:subClassOf :X3DBinaryContainerGeometryNode .
        :PopGeometry a owl:Class ; rdfs:subClassOf :X3DBinaryContainerGeometryNode .
    :X3DMetadataObject a owl:Class ; rdfs:subClassOf :X3DNode .
    :MetadataBoolean a owl:Class ; rdfs:subClassOf :X3DMetadataObject .
    :MetadataDouble a owl:Class ; rdfs:subClassOf :X3DMetadataObject .
    :MetadataFloat a owl:Class ; rdfs:subClassOf :X3DMetadataObject .
    :MetadataInteger a owl:Class ; rdfs:subClassOf :X3DMetadataObject .
    :MetadataSet a owl:Class ; rdfs:subClassOf :X3DMetadataObject .
    :MetadataString a owl:Class ; rdfs:subClassOf :X3DMetadataObject .
    :X3DRigidJointNode a owl:Class ; rdfs:subClassOf :X3DNode .
    :BallJoint a owl:Class ; rdfs:subClassOf :X3DRigidJointNode .
    :DoubleAxisHingeJoint a owl:Class ; rdfs:subClassOf :X3DRigidJointNode .
    :MotorJoint a owl:Class ; rdfs:subClassOf :X3DRigidJointNode .
    :SingleAxisHingeJoint a owl:Class ; rdfs:subClassOf :X3DRigidJointNode .
    :SliderJoint a owl:Class ; rdfs:subClassOf :X3DRigidJointNode .
    :UniversalJoint a owl:Class ; rdfs:subClassOf :X3DRigidJointNode .
    :X3DVolumeRenderStyleNode a owl:Class ; rdfs:subClassOf :X3DNode .
    :MPRVolumeStyle a owl:Class ; rdfs:subClassOf :X3DVolumeRenderStyleNode .
    :ProjectionVolumeStyle a owl:Class ; rdfs:subClassOf :X3DVolumeRenderStyleNode .
    :RadarVolumeStyle a owl:Class ; rdfs:subClassOf :X3DVolumeRenderStyleNode .
    :StippleVolumeStyle a owl:Class ; rdfs:subClassOf :X3DVolumeRenderStyleNode .
    :X3DComposableVolumeRenderStyleNode a owl:Class ; rdfs:subClassOf
:X3DVolumeRenderStyleNode .
```

```
      :BlendedVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :BoundaryEnhancementVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :CartoonVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :ComposedVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :EdgeEnhancementVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :OpacityMapVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :ShadedVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :SilhouetteEnhancementVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .
      :ToneMappedVolumeStyle a owl:Class ;
        rdfs:subClassOf :X3DComposableVolumeRenderStyleNode .

### Additional Concepts ###

:3DObject a owl:Class .
  :Primitive3DObject a owl:Class ; rdfs:subClassOf :3DObject .
    :Spheroid a owl:Class ; rdfs:subClassOf :Primitive3DObject .
      :OblateSpheroid a owl:Class ; rdfs:subClassOf :Spheroid .
      :ProlateSpheroid a owl:Class ; rdfs:subClassOf :Spheroid .
  :Complex3DObject a owl:Class ; rdfs:subClassOf :3DObject .
:Polytope a owl:Class .
  :Polygon a owl:Class ; rdfs:subClassOf :Polytope .
  :Polyhedron a owl:Class ; rdfs:subClassOf :Polytope .
```

```
    :Tetrahedron a owl:Class ; rdfs:subClassOf :Polyhedron .
    :Dodecahedron a owl:Class ; rdfs:subClassOf :Polyhedron .
    :Icosidodecahedron a owl:Class ; rdfs:subClassOf :Polyhedron .
    :Cubicuboctahedron a owl:Class ; rdfs:subClassOf :Polyhedron .
    :Triacontahedron a owl:Class ; rdfs:subClassOf :Polyhedron .

### AutoDesk 3ds Max ###

:3dsMaxObject a owl:Class ; rdfs:subClassOf :3DObject ; rdfs:domain :3dsMaxModel .
  :GeometryObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
    :GeometricPrimitive a owl:Class ; rdfs:subClassOf :GeometryObject .
      :StandardPrimitive a owl:Class ; rdfs:subClassOf :GeometricPrimitive .
        :Box a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Cone a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Sphere a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :GeoSphere a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Cylinder a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Tube a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Torus a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Pyramid a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :TeaPot a owl:Class ; rdfs:subClassOf :StandardPrimitive .
        :Plane a owl:Class ; rdfs:subClassOf :StandardPrimitive .
      :ExtendedPrimitive a owl:Class ; rdfs:subClassOf :GeometricPrimitive .
        :Hedra a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
        :TorusKnot a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
        :ChamferBox a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
        :ChampherCylinder a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
        :OilTank a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
        :Capsule a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
        :Spindle a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
```

```
      :LExt a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
      :Gengon a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
      :CExt a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
      :RingWave a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
      :Hose a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
      :Prism a owl:Class ; rdfs:subClassOf :ExtendedPrimitive .
  :AECObject a owl:Class ; rdfs:subClassOf :GeometryObject .
    :AECDesignElement a owl:Class ; rdfs:subClassOf :AECObject .
      :Door a owl:Class ; rdfs:subClassOf :AECDesignElement .
        :PivotDoor a owl:Class ; rdfs:subClassOf :Door .
        :BiFoldDoor a owl:Class ; rdfs:subClassOf :Door .
        :SlidingDoor a owl:Class ; rdfs:subClassOf :Door .
      :Window a owl:Class ; rdfs:subClassOf :AECDesignElement .
        :AwningWindow a owl:Class ; rdfs:subClassOf :Window .
        :FixedWindow a owl:Class ; rdfs:subClassOf :Window .
        :ProjectedWindow a owl:Class ; rdfs:subClassOf :Window .
        :Casement a owl:Class ; rdfs:subClassOf :Window .
        :PivotedWindow a owl:Class ; rdfs:subClassOf :Window .
        :SlidingWindow a owl:Class ; rdfs:subClassOf :Window .
      :Stairs a owl:Class ; rdfs:subClassOf :AECDesignElement .
        :StraightStair a owl:Class ; rdfs:subClassOf :Stairs .
        :UTypeStair a owl:Class ; rdfs:subClassOf :Stairs .
        :LTypeStair a owl:Class ; rdfs:subClassOf :Stairs .
        :SpiralStair a owl:Class ; rdfs:subClassOf :Stairs .
    :AECExtendedObject a owl:Class ; rdfs:subClassOf :AECObject .
      :Foliage a owl:Class ; rdfs:subClassOf :AECExtendedObject .
      :Railing a owl:Class ; rdfs:subClassOf :AECExtendedObject .
      :Wall a owl:Class ; rdfs:subClassOf :AECExtendedObject .
  :CompoundObject a owl:Class ; rdfs:subClassOf :GeometryObject .
    :Morph a owl:Class ; rdfs:domain :CompoundObject .
```

```
  :Scatter a owl:Class ; rdfs:domain :CompoundObject .
  :Conform a owl:Class ; rdfs:domain :CompoundObject .
  :Connect a owl:Class ; rdfs:domain :CompoundObject .
  :BlobMesh a owl:Class ; rdfs:domain :CompoundObject .
  :ShapeMerge a owl:Class ; rdfs:domain :CompoundObject .
  :Boolean a owl:Class ; rdfs:domain :CompoundObject .
  :Terrain a owl:Class ; rdfs:domain :CompoundObject .
  :Loft a owl:Class ; rdfs:domain :CompoundObject .
  :Mesher a owl:Class ; rdfs:domain :CompoundObject .
  :ProBoolean a owl:Class ; rdfs:domain :CompoundObject .
  :ProCutter a owl:Class ; rdfs:domain :CompoundObject .
:ParticleSystem a owl:Class ; rdfs:subClassOf :GeometryObject .
  :PFSource a owl:Class ; rdfs:subClassOf :ParticleSystem .
  :Snow a owl:Class ; rdfs:subClassOf :ParticleSystem .
  :Blizzard a owl:Class ; rdfs:subClassOf :ParticleSystem .
  :PCloud a owl:Class ; rdfs:subClassOf :ParticleSystem .
  :Spray a owl:Class ; rdfs:subClassOf :ParticleSystem .
  :SuperSpray a owl:Class ; rdfs:subClassOf :ParticleSystem .
  :PArrayt a owl:Class ; rdfs:subClassOf :ParticleSystem .
:PatchGrid a owl:Class ; rdfs:subClassOf :GeometryObject .
  :TriPatch a owl:Class ; rdfs:subClassOf :PatchGrid .
  :QuadPatch a owl:Class ; rdfs:subClassOf :PatchGrid .
:BodyObject a owl:Class ; rdfs:subClassOf :GeometryObject .
  :BodyUtility a owl:Class ; rdfs:subClassOf :BodyObject .
  :JoinBodies a owl:Class ; rdfs:subClassOf :BodyObject .
  :BodyCutter a owl:Class ; rdfs:subClassOf :BodyObject .
:NURBSSurface a owl:Class ; rdfs:subClassOf :GeometryObject .
  :PointSurf a owl:Class ; rdfs:subClassOf :NURBSSurface .
  :CVSurf a owl:Class ; rdfs:subClassOf :NURBSSurface .
:MentalRayObject a owl:Class ; rdfs:subClassOf :GeometryObject .
```

```
      :MRProxy a owl:Class ; rdfs:subClassOf :MentalRayObject .
    :PointCloud a owl:Class ; rdfs:subClassOf :GeometryObject .
    :DynamicsObject a owl:Class ; rdfs:subClassOf :GeometryObject .
      :Spring a owl:Class ; rdfs:subClassOf :DynamicsObject .
      :Damper a owl:Class ; rdfs:subClassOf :DynamicsObject .
    :AlembicObject  a owl:Class ; rdfs:subClassOf :GeometryObject .
:ShapeObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
  :Spline a owl:Class ; rdfs:subClassOf :ShapeObject .
    :Line a owl:Class ; rdfs:subClassOf :Spline .
    :Circle a owl:Class ; rdfs:subClassOf :Spline .
    :Arc a owl:Class ; rdfs:subClassOf :Spline .
    :NGon a owl:Class ; rdfs:subClassOf :Spline .
    :Text a owl:Class ; rdfs:subClassOf :Spline .
    :Egg a owl:Class ; rdfs:subClassOf :Spline .
    :Rectangle a owl:Class ; rdfs:subClassOf :Spline .
    :Ellipse a owl:Class ; rdfs:subClassOf :Spline .
    :Donut a owl:Class ; rdfs:subClassOf :Spline .
    :Star a owl:Class ; rdfs:subClassOf :Spline .
    :Helix a owl:Class ; rdfs:subClassOf :Spline .
    :Section a owl:Class ; rdfs:subClassOf :Spline .
  :NURBSCurve a owl:Class ; rdfs:subClassOf :ShapeObject ;
    dc:description "A non-uniform rational B-spline (NURBS) curve." .
    :PointCurve a owl:Class ; rdfs:subClassOf :NURBSCurve .
    :CVCurve a owl:Class ; rdfs:subClassOf :NURBSCurve .
  :ExtendedSpline a owl:Class ; rdfs:subClassOf :ShapeObject .
    :WRectangle a owl:Class ; rdfs:subClassOf :ExtendedSpline .
    :Angle a owl:Class ; rdfs:subClassOf :ExtendedSpline .
    :WideFlange a owl:Class ; rdfs:subClassOf :ExtendedSpline .
    :Channel a owl:Class ; rdfs:subClassOf :ExtendedSpline .
    :Tee a owl:Class ; rdfs:subClassOf :ExtendedSpline .
```

```
:LightObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
  :StandardLightObject a owl:Class ; rdfs:subClassOf :LightObject .
    :TargetSpotlightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :TargetDirectlightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :OmnilightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :MRAreaOmnilightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :FreeSpotlightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :FreeDirectlightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :SkylighObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
    :MRAreaSpotlightObject a owl:Class ; rdfs:subClassOf :StandardLightObject .
  :PhotometricLightObject a owl:Class ; rdfs:subClassOf :LightObject .
    :TargetLightObject a owl:Class ; rdfs:subClassOf :PhotometricLightObject .
    :MRSkyPortalLightObject a owl:Class ; rdfs:subClassOf :PhotometricLightObject .
    :FreeLightObject a owl:Class ; rdfs:subClassOf :PhotometricLightObject .
:CameraObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
  :PhysicalCameraObject a owl:Class ; rdfs:subClassOf :CameraObject .
  :TargetCameraObject a owl:Class ; rdfs:subClassOf :CameraObject .
  :FreeCameraObject a owl:Class ; rdfs:subClassOf :CameraObject .
:HelperObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
  :StandardHelperObject a owl:Class ; rdfs:subClassOf :HelperObject .
    :DummyObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :CrowdObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :ExposeTmObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :PointObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :IRaySectionObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :ProtractorObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :ContainerObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :DelegateObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :GridObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
    :LightMeterObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
```

```
  :TapeObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
  :CompassObject a owl:Class ; rdfs:subClassOf :StandardHelperObject .
:AtmosphericApparatus a owl:Class ; rdfs:subClassOf :HelperObject .
  :BoxGizmo a owl:Class ; rdfs:subClassOf :AtmosphericApparatus .
  :SphereGizmo a owl:Class ; rdfs:subClassOf :AtmosphericApparatus .
  :CylGizmo a owl:Class ; rdfs:subClassOf :AtmosphericApparatus .
:CameraMatch a owl:Class ; rdfs:subClassOf :HelperObject .
  :CamPointObject a owl:Class ; rdfs:subClassOf :CameraMatch .
:AssemblyHead a owl:Class ; rdfs:subClassOf :HelperObject .
  :Luminaire a owl:Class ; rdfs:subClassOf :AssemblyHead .
:Manipulator a owl:Class ; rdfs:subClassOf :HelperObject .
  :ConeAngleManipulator a owl:Class ; rdfs:subClassOf :Manipulator .
  :PlaneAngleManipulator a owl:Class ; rdfs:subClassOf :Manipulator .
  :Slider  a owl:Class ; rdfs:subClassOf :Manipulator .
:ParticleFlowObject a owl:Class ; rdfs:subClassOf :HelperObject .
  :FindTarget a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :InitialState a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :ParticlePaint a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :MPSolvent a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :BirthGrid a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :MPWorld a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :DataIcon a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :TestIcon a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :SpeedByIcon a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :GroupSelect a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :BirthTexture a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :MPBuoyancy a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :BirthStream a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :BlurWind a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
  :RandomWalk a owl:Class ; rdfs:subClassOf :ParticleFlowObject .
```

```
  :MassFXObject a owl:Class ; rdfs:subClassOf :HelperObject .
    :UConstraintObject a owl:Class ; rdfs:subClassOf :MassFXObject .
  :CATObject a owl:Class ; rdfs:subClassOf :HelperObject .
    :CATMuscle a owl:Class ; rdfs:subClassOf :CATObject .
    :MuscleStrand a owl:Class ; rdfs:subClassOf :CATObject .
    :CATParent a owl:Class ; rdfs:subClassOf :CATObject .
  :VRML97HelperObject a owl:Class ; rdfs:subClassOf :HelperObject .
    :AnchorObject a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :ProxSensor a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :NavInfo a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :Fog a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :SoundObject a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :LODObject a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :TouchSensor a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :TimeSensor a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :BackgroundObject a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :AudioClip a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :Billboard a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
    :InlineObject a owl:Class ; rdfs:subClassOf :VRML97HelperObject .
:SpaceWarpObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
  :ForceObject a owl:Class ; rdfs:subClassOf :SpaceWarpObject .
    :PushObject a owl:Class ; rdfs:subClassOf :ForceObject .
    :Vortex a owl:Class ; rdfs:subClassOf :ForceObject .
    :PBomb a owl:Class ; rdfs:subClassOf :ForceObject .
    :Gravity a owl:Class ; rdfs:subClassOf :ForceObject .
    :Displace a owl:Class ; rdfs:subClassOf :ForceObject .
    :Motor a owl:Class ; rdfs:subClassOf :ForceObject .
    :Drag a owl:Class ; rdfs:subClassOf :ForceObject .
    :PathFollow a owl:Class ; rdfs:subClassOf :ForceObject .
    :Wind a owl:Class ; rdfs:subClassOf :ForceObject .
```

```
  :DeflectorObject a owl:Class ; rdfs:subClassOf :SpaceWarpObject .
    :POmniFlect a owl:Class ; rdfs:subClassOf :DeflectorObject .
    :UOmniFlect a owl:Class ; rdfs:subClassOf :DeflectorObject .
    :SDeflector a owl:Class ; rdfs:subClassOf :DeflectorObject .
    :SOmniFlect a owl:Class ; rdfs:subClassOf :DeflectorObject .
    :UDeflector a owl:Class ; rdfs:subClassOf :DeflectorObject .
  :DeformableObject a owl:Class ; rdfs:subClassOf :SpaceWarpObject .
    :FFDBox a owl:Class ; rdfs:subClassOf :DeformableObject .
    :FFDCyl a owl:Class ; rdfs:subClassOf :DeformableObject .
    :Wave a owl:Class ; rdfs:subClassOf :DeformableObject .
    :Ripple a owl:Class ; rdfs:subClassOf :DeformableObject .
    :Displace a owl:Class ; rdfs:subClassOf :DeformableObject .
    :ConformDeformableObject a owl:Class ; rdfs:subClassOf :DeformableObject .
    :Bomb a owl:Class ; rdfs:subClassOf :DeformableObject .
  :ModifierBasedObject a owl:Class ; rdfs:subClassOf :SpaceWarpObject .
    :BendObject a owl:Class ; rdfs:subClassOf :ModifierBasedObject .
    :Twist a owl:Class ; rdfs:subClassOf :ModifierBasedObject .
    :Taper a owl:Class ; rdfs:subClassOf :ModifierBasedObject .
    :Skew a owl:Class ; rdfs:subClassOf :ModifierBasedObject .
    :Noise a owl:Class ; rdfs:subClassOf :ModifierBasedObject .
    :Stretch a owl:Class ; rdfs:subClassOf :ModifierBasedObject .
  :ParticleObject a owl:Class ; rdfs:subClassOf :SpaceWarpObject .
    :VectorFieldObject a owl:Class ; rdfs:subClassOf :ParticleObject .
:SystemObject a owl:Class ; rdfs:subClassOf :3dsMaxObject .
  :BoneSystem a owl:Class ; rdfs:subClassOf :SystemObject .
  :RingArray a owl:Class ; rdfs:subClassOf :SystemObject .
  :BipedObject a owl:Class ; rdfs:subClassOf :SystemObject .
  :SunlightObject a owl:Class ; rdfs:subClassOf :SystemObject .
  :DaylightObject a owl:Class ; rdfs:subClassOf :SystemObject .
```

```
### AutoDesk Maya ###

:MayaObject a owl:Class ; rdfs:subClassOf :3DObject ; rdfs:domain :MayaModel .
  :nHair a owl:Class ; rdfs:subClassOf :MayaObject .
  :nCloth a owl:Class ; rdfs:subClassOf :MayaObject .
  :nParticle a owl:Class ; rdfs:subClassOf :MayaObject .
  :Fur a owl:Class ; rdfs:subClassOf :MayaObject .
  :nCloth a owl:Class ; rdfs:subClassOf :MayaObject .
  :MayaSurface a owl:Class .
    :PolygonGeometry a owl:Class ; rdfs:subClassOf :MayaSurface .
    :Subdivison a owl:Class ; rdfs:subClassOf :MayaSurface .

###  3ds Max/Maya ###

:NonRiggedModel a owl:Class ; rdfs:domain :3DModel .
:RiggedModel a owl:Class ; rdfs:domain :3DModel , :3DAnimation .
  :JointChain a owl:Class ; rdfs:subClassOf :RiggedModel .
    :Joint a owl:Class ; rdfs:subClassOf :JointChain .
  :Skeleton a owl:Class ; rdfs:subClassOf :RiggedModel .
:Node a owl:Class .
:MayaMuscleSystem a owl:Class ; rdfs:domain :MayaModel .
  :Capsule a owl:Class ; rdfs:subClassOf :MayaMuscleSystem .
  :Bone a owl:Class ; rdfs:subClassOf :MayaMuscleSystem .
  :Muscle a owl:Class ; rdfs:subClassOf :MayaMuscleSystem .
:Map a owl:Class .
  :NormalMap a owl:Class ; rdfs:subClassOf :Map .
    :ObjectSpaceMap a owl:Class ; rdfs:subClassOf :NormalMap .
    :TangentSpaceMap a owl:Class ; rdfs:subClassOf :NormalMap .
  :BumpMap a owl:Class ; skos:broader :MultiTexture ; rdfs:subClassOf :Map .
  :DisplacementMap a owl:Class ; skos:related :BumpMap ; rdfs:subClassOf :Map .
```

```
### AutoDesk AutoCAD ###

:AutoCADObject a owl:Class ; rdfs:subClassOf :3DObject ; rdfs:domain :AutoCADModel .
  :AutoCADMesh a owl:Class ; rdfs:subClassOf :AutoCADObject .
    :SmoothObject a owl:Class ; rdfs:subClassOf :AutoCADMesh .
  :AutoCADSolid a owl:Class ; rdfs:subClassOf :AutoCADObject .
  :AutoCADSurface a owl:Class ; rdfs:subClassOf :AutoCADObject .

### Additional Concepts with Reused Concepts ###

dc:Agent a owl:Class .
  vidont:Avatar a owl:Class ; rdfs:subClassOf dc:Agent ;
   dc:description "An animated character in a simulated environment (CGI)." .
  schema:copyrightHolder a owl:Class ; rdfs:subClassOf dc:Agent .
    foaf:Organization a owl:Class ; rdfs:subClassOf schema:copyrightHolder .
      :DesignStudio a owl:Class ; rdfs:subClassOf foaf:Organization .
      :animationStudio a owl:Class ; rdfs:subClassOf foaf:Organization .
      vidont:filmStudio a owl:Class ; rdfs:subClassOf foaf:Organization .
      schema:productionCompany a owl:Class ; rdfs:subClassOf foaf:Organization .
    foaf:Person a owl:Class ; rdfs:subClassOf schema:copyrightHolder .
      :3DGraphicDesigner a owl:Class .
schema:CreativeWork a owl:Class .
  :3DComputerGraphicsSoftware a owl:Class ; rdfs:subClassOf schema:CreativeWork .
  :3DModel a owl:Class ; rdfs:subClassOf schema:CreativeWork .
    :3dsMaxModel a owl:Class ; rdfs:subClassOf :3DModel .
    :BlenderModel a owl:Class ; rdfs:subClassOf :3DModel .
    :MayaModel a owl:Class ; rdfs:subClassOf :3DModel .
    :AC3DModel a owl:Class ; rdfs:subClassOf :3DModel .
    :ModoModel a owl:Class ; rdfs:subClassOf :3DModel .
```

```
    :Seamless3dModel a owl:Class ; rdfs:subClassOf :3DModel .
    :X3DModel a owl:Class ; rdfs:subClassOf :3DModel .
  :AutoCADModel a owl:Class .
  schema:Review a owl:Class ; rdfs:subClassOf schema:CreativeWork ;
    dc:description "A review of an item, e.g., of a movie."@en .
  vidont:Video a owl:Class ; rdfs:subClassOf schema:CreativeWork .
schema:Movie a owl:Class ; rdfs:subClassOf :Video ; owl:disjointWith
schema:MusicVideoObject , schema:TVSeason , schema:trailer .
  :animatedMovie a owl:Class ; rdfs:subClassOf schema:Movie ; dc:description
"Differentiates films from their releases. An animated movie can be a cartoon, a
cartoon with motion picture, a computer animaton, or live action." .
    :computerAnimation a owl:Class ; rdfs:subClassOf :animatedMovie .
      :3DAnimation a owl:Class ; rdfs:subClassOf :computerAnimation .

### Roles ###

:alphaTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
 dc:description "Texture containing alpha component." .
:ambientTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing ambient component." .
:attrib a owl:ObjectProperty ; rdfs:domain :ElevationGrid , :X3DComposedGeometryNode ,
:IndexedTriangleStripSet , :LineSet , :QuadSet .
:back a owl:ObjectProperty ; rdfs:domain :ComposedCubeMapTexture ; dc:description
"Texture for the back of the cubemap." .
:background a owl:ObjectProperty ; rdfs:domain :RenderedTexture .
:bodies a owl:ObjectProperty ; rdfs:domain :RigidBodyCollection .
:body1 a owl:ObjectProperty ; rdfs:domain :X3DRigidJointNode ;
  dc:description "The first body to be joint by the node." .
:body2 a owl:ObjectProperty ; rdfs:domain :X3DRigidJointNode ;
  dc:description "The second rigid body to be joint by the node." .
```

```
:bottomTexture a owl:ObjectProperty ; rdfs:domain :ComposedCubeMapTexture ;
dc:description "Texture for the bottom of the cubemap." .
:cappingColor a owl:ObjectProperty ; rdfs:domain :ClipPlane ;
 dc:description "Defines the color of the capping." .
:children a owl:ObjectProperty , owl:TransitiveProperty ; rdfs:domain :X3DLODNode ,
:X3DTransformNode , :X3DGroupingNode , :Inline  ;
  dc:description "Grouping nodes have a field that contains a list of children nodes" .
:collidables a owl:ObjectProperty ; rdfs:domain :CollisionCollection ; dc:description
"The collidables field can be managed as a single entity for resolution of inter-object
collisions with other groups of collidable objects." .
:collider a owl:ObjectProperty ; rdfs:domain :CollisionSensor ; dc:description
"Specifies the nodes and spaces that are to be included in collision detection
computations." .
:colorMaskMode a owl:ObjectProperty ; dc:description "Holds a ColorMaskMode node." .
:data a owl:ObjectProperty ; rdfs:domain :GeoMetadata ; dc:description "The data field
is used to list all of the other nodes in a scene by DEF name that reference the data
described in the GeoMetadata node. The nodes in the data field are not rendered, so
DEF/USE can be used in order to first describe them and then to use them in the scene
graph." .
:depicts a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:depthMode a owl:ObjectProperty ; dc:description "Holds the depthMode node." .
:depthTexture a owl:ObjectProperty ; rdfs:domain :RadarVolumeStyle .
:diffuseDisplacementTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing diffuse displacement component." .
:diffuseTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing diffuse component." .
:displacementTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing displacement component." .
:displacers a owl:ObjectProperty ; rdfs:domain :HAnimJoint , :HAnimSegment .
```

```
:emissiveTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing emissive component." .
:environmentTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Cube texture containing the environment for perfect specular reflection
and transmission." .
:excludeNodes a owl:ObjectProperty ; rdfs:domain :RenderedTexture .
:fields a owl:ObjectProperty ; rdfs:domain :ComposedShader ;
 dc:description "Contains all fields of shader parts." .
:fog a owl:ObjectProperty ; rdfs:domain :RenderedTexture .
:FontStyle a owl:ObjectProperty , owl:TransitiveProperty ; rdfs:domain :Text ;
dc:description "Sets the text style." .
:front a owl:ObjectProperty ; rdfs:domain :ComposedCubeMapTexture ;
 dc:description "Texture for the front of the cubemap." .
:geometry a owl:ObjectProperty ; rdfs:domain :X3DVolumeDataNode , :RigidBody ,
:X3DShapeNode ; dc:description "Holds the geometry node." .
:gradients a owl:ObjectProperty ; rdfs:domain :IsoSurfaceVolumeData ;
 dc:description "Allows to provide the normals of the volume data." .
:joints a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid , :RigidBodyCollection .
:left a owl:ObjectProperty ; rdfs:domain :ComposedCubeMapTexture ; dc:description
"Texture for the left side of the cubemap." .
:levels a owl:ObjectProperty ; rdfs:domain :PopGeometry .
:massDensityModel a owl:ObjectProperty ; rdfs:domain :RigidBody .
:material a owl:ObjectProperty ; rdfs:domain :ShadedVolumeStyle ; dc:description
"Material." .
:multiDiffuseAlphaTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Multi diffuse alpha texture." .
:multiEmmisiveAmbientIntensityTexture a owl:ObjectProperty ; rdfs:domain
:CommonSurfaceShader ; dc:description "Multi emissive ambientIntensity texture." .
:multiSpecularShininessTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader
; dc:description "Multi specular shininess texture." .
```

```
:multiVisibilityTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Multi visibility texture." .
:normalTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
 dc:description "Texture containing normal component for normal mapping." .
:parts a owl:ObjectProperty ; rdfs:domain :ComposedShader ;
 dc:description "List of shader parts." .
:privateChild1Node a owl:ObjectProperty ; rdfs:domain :GeoLOD .
:privateChild2Node a owl:ObjectProperty ; rdfs:domain :GeoLOD .
:privateChild3Node a owl:ObjectProperty ; rdfs:domain :GeoLOD .
:privateChild4Node a owl:ObjectProperty ; rdfs:domain :GeoLOD .
:proxy a owl:ObjectProperty ; rdfs:domain :X3DGroupingNode ;
 dc:description "Alternate object to be checked for collision, in place of the children
of the node." .
:reflectionTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing reflection component." .
:renderStyle a owl:ObjectProperty ; rdfs:domain :ComposedVolumeStyle ,
:IsoSurfaceVolumeData , :SegmentedVolumeData , :VolumeData .
:right a owl:ObjectProperty ; rdfs:domain :ComposedCubeMapTexture ;
 dc:description "Texture for the right side of the cubemap." .
:root a owl:ObjectProperty ; rdfs:domain :DynamicLOD .
:rootNode a owl:ObjectProperty ; rdfs:domain :GeoLOD ; dc:description "The rootUrl and
rootNode fields provide two different ways to specify the geometry of the root tile.
The rootUrl field lets you specify a URL for a file that contains the geometry." .
:segmentIdentifiers a owl:ObjectProperty ; rdfs:domain :SegmentedVolumeData .
:segments a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:shaders a owl:ObjectProperty ; dc:description "Shaders." .
:shape a owl:ObjectProperty ; rdfs:domain :CADFace ; dc:description "The geometry and
appearance for the face or an LOD node containing different detail levels of the
shape." .
```

```
:shininessTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing shininess component." .
:sites a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:skeleton a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:skin a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:skinCoord a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:skinNormal a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:source a owl:ObjectProperty ; rdfs:domain :Sound .
:specularTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing specular component." .
:surfaceNormals a owl:ObjectProperty ; rdfs:domain :X3DComposableVolumeRenderStyleNode
, :OpacityMapVolumeStyle ; dc:description "Allows to provide the normals of the volume
data." .
:texture a owl:ObjectProperty ; dc:description "Texture." .
:textureProp a owl:ObjectProperty ; rdfs:domain :X3DTexture3DNode ;
dc:description "Sets a TextureProperty node." .
:textureTransform a owl:ObjectProperty ; dc:description "Texture transform." .
:top a owl:ObjectProperty ; rdfs:domain :ComposedCubeMapTexture , :Cone ,
:SlopedCylinder , :Snout ; dc:description "Texture for the top of the cubemap. For
cones, it specifies whether the top cap of the cone is created." .
:transferFunction a owl:ObjectProperty ; rdfs:domain :OpacityMapVolumeStyle ,
:RadarVolumeStyle ; dc:description "A texture that is going to be used to map each
voxel value to a specific color output." .
:transmissionTexture a owl:ObjectProperty ; rdfs:domain :CommonSurfaceShader ;
dc:description "Texture containing transmission component." .
:viewpoints a owl:ObjectProperty ; rdfs:domain :HAnimHumanoid .
:voxels a owl:ObjectProperty ; rdfs:domain :BlendedVolumeStyle ,
 :IsoSurfaceVolumeData , :SegmentedVolumeData ; dc:description "An ImageTextureAtlas
  node containing the volume data to be blended." .
```

```
:weightTransferFunction1 a owl:ObjectProperty ; rdfs:domain :BlendedVolumeStyle ;
dc:description "A 2D texture that maps each opacity value to a weight value, that will
be used on the parent volume data." .
:weightTransferFunction2 a owl:ObjectProperty ; rdfs:domain :BlendedVolumeStyle ;
dc:description "A 2D texture that maps each opacity value to a weight value, that will
be used on the volume data to be blended." .

:accumFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RadarVolumeStyle ; dc:description "Specifies a multiplier for the color and intensity
accumulated as the ray traverses the volume data in case of alpha composting based
volume rendering" .
:allowViewpointInside a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
 rdfs:domain :X3DVolumeDataNode ;
 dc:description "Allow to locate the viewpoint inside the volume." .
:alphaClipThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ;
 dc:description "Specify the threshold for the alpha clipping." .
:alphaFactor a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of alphaTexture is multiplied by this
value. If no texture is set, the value is used directly." .
:alphaFunc a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List
; rdf:first "none" ; rdf:rest [ a rdf:List ; rdf:first "never" ; rdf:rest [ a rdf:List
; rdf:first "less" ; rdf:rest [ a rdf:List ; rdf:first "equal" ; rdf:rest [ a rdf:List
; rdf:first "lequal" ; rdf:rest [ a rdf:List ; rdf:first "greater" ; rdf:rest [ a
rdf:List ; rdf:first "notequal" ; rdf:rest [ a rdf:List ; rdf:first "gequal" ; rdf:rest
[ a rdf:List ; rdf:rest [ a rdf:List ; rdf:first "always" ; rdf:rest rdf:nil ] ] ] ] ]
] ] ] ] ] ] ; rdfs:domain :BlendMode .
:alphaFuncValue a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :BlendMode
; dc:description "Defines how fragments which do not fulfill a certain condition are
handled." .
```

```
:alphaTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:CommonSurfaceShader ; dc:description "ChannelMask for alphaTexture in the form of a
glsl swizzle (e.g., 'rgb', 'a')." .
:alphaTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for alphaTexture." .
:alphaTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "The texture unit." .
:ambientFactor a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "The value of ambientTexture is
multiplied by this value. If no texture is set, the value is used directly." .
:ambientIntensity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:X3DLightNode , :X3DMaterialNode ; dc:description "Specifies the intensity of the
ambient emission from the light." .
:ambientTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:CommonSurfaceShader ; dc:description "ChannelMask for ambientTexture in the form of a
glsl swizzle (e.g., 'rgb', 'a')." .
:ambientTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for ambientTexture." .
:ambientTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "The texture unit." .
:anchorPoint a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
rdfs:domain :SingleAxisHingeJoint , :DoubleAxisHingeJoint , :UniversalJoint ;
dc:description "The common anchor point." .
:angle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :RectangularTorus ,
:SolidOfRevolution , :Torus ; dc:description "Defines the length of the torus as angle.
2 * π is a full torus (RectangularTorus) or the subtended angle through which the 2D
loop is swept around the x axis (SolidOfRevolution)." .
:angularDampingFactor a owl:DatatypeProperty ; rdfs:range xsd:float ;
```

```
 rdfs:domain :RigidBody ; dc:description "Angular damping factor." .
:angularVelocity a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
 rdfs:domain :RigidBody ; dc:description "Provides a constant velocity value to the
object every frame." .
:anisotropicDegree a owl:DatatypeProperty ; rdfs:range xsd:float ;
 rdfs:domain :TextureProperties ; dc:description "Describes the minimum degree of
 anisotropy to account for in texture filtering." .
:applied a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
rdfs:domain :LineProperties .
:attenuation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:PointLight , :SpotLight ; dc:description "PointLight/spotlight node's illumination
falls off with distance as specified by three attenuation coefficients." .
:attributeStride a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Stride of all (interleaved) attributes, given in
 bytes." .
:autoDamp a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :RigidBody ;
dc:description "The application of damping is controlled through the use of the
autoDamp field. When the value is FALSE, no damping is applied. When the value is TRUE,
rotational and translational damping is calculated and applied." .
:autoDisable a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :RigidBody ,
:RigidBodyCollection ; dc:description "By default, this automatic disabling is turned
off. It may be enabled by setting the autoDisable field to TRUE." .
:autoOffset a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DDragSensorNode ; dc:description "Determines whether offset values from previous
drag gestures are remembered / accumulated." .
:autoRefinement a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
 rdfs:domain :RefinementTexture ; dc:description "Defines whether texture refinement
 should be managed by another component." .
:avatarSize a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:NavigationInfo ; dc:description "avatarSize triplet values are: (a) collision distance
```

```
between user and geometry (near culling plane of the view frustrum) (b) viewer height
above terrain (c) tallest height viewer can WALK over. Hint: keep
(avatarSize.CollisionDistance / visibilityLimit) less then; 10,000 to avoid aliasing
artifacts (i.e. polygon 'tearing'). Interchange profile hint: this field may be
ignored." .
:axis a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:SingleAxisHingeJoint , :SliderJoint ; dc:description "The axis of the hinge is defined
to be along the unit vector described in the axis field and centered on the anchorPoint
described in world coordinates." .
:axis1 a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:DoubleAxisHingeJoint , :UniversalJoint ; dc:description "Specified relative to the
first body." .
:axis1Angle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :MotorJoint ;
dc:description "The three axis angle fields provide angles (in angle base units) for
this frame for the corresponding motor axis when in user-calculated mode." .
:axis1Torque a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "Torque on axis 1." .
:axis2 a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:DoubleAxisHingeJoint , :UniversalJoint ; dc:description "Specified relative to the
second body." .
:axis2Angle a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "The three axis angle fields provide angles (in angle base units) for
this frame for the corresponding motor axis when in user-calculated mode." .
:axis2Torque a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "Torque on axis 2." .
:axis3Angle a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "The three axis angle fields provide angles (in angle base units) for
this frame for the corresponding motor axis when in user-calculated mode." .
:axis3Torque a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "Torque on axis 3." .
```

```
:axisOfRotation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; dc:description
"The axis of rotation, whose direction is relative to local coordinate system. Axis 0 0
0 faces the viewer." .
:axisRotation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CylinderSensor , :PlaneSensor .
:backAmbientIntensity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:TwoSidedMaterial ; dc:description "Defines the ambient intensity for the back side." .
:backDiffuseColor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:TwoSidedMaterial ; dc:description "Defines the diffuse color for the back side." .
:backEmissiveColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:TwoSidedMaterial ; dc:description "Defines the emissive color for the back side." .
:backShininess a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:TwoSidedMaterial ; dc:description "Defines the shininess for the back side." .
:backSpecularColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:TwoSidedMaterial ; dc:description "Defines the specular color for the back side." .
:backTransparency a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:TwoSidedMaterial ; dc:description "Defines the transparency for the back side." .
:backUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
 rdfs:domain :Background .
:bbMaxModF a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
 rdfs:domain :PopGeometry ; dc:description "The maximum coordinates of the bounding
 box, in a normalized range between [0,1], and given modulo maxBBSize." .
:bbMin a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :PopGeometry ;
dc:description "Minimum coordinates of the bounding box, in object coordinates." .
:bbMinModF a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:PopGeometry ; dc:description "The minimum coordinates of the bounding box, in a
normalized range between [0,1], and given modulo maxBBSize." .
:bbShiftVec a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
rdfs:domain :PopGeometry ; dc:description "Field for internal use." .
```

```
:bboxCenter a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DLODNode , :X3DTransformNode , :X3DGroupingNode , :X3DVolumeDataNode , :Inline ,
:X3DShapeNode ; dc:description "Center of the bounding box." .
:bboxSize a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :X3DLODNode
, :X3DTransformNode , :X3DGroupingNode , :X3DVolumeDataNode , :X3DShapeNode , :Inline ;
dc:description "Size of the bounding box." .
:beamWidth a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SpotLight ;
dc:description "Specifies an inner solid angle in which the light source emits light at
uniform full intensity." .
:beginCap a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Extrusion ;
dc:description "Specifies whether the beginCap should exist." .
:bind a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DBackgroundNode , :X3DEnvironmentNode , :Fog , :X3DViewpointNode ,
:X3DNavigationInfoNode ; dc:description "Pushes/pops the node on/from the top of the
bindable stack." .
:binormal a owl:DatatypeProperty ; rdfs:range xsd:string ;
 rdfs:domain :BinaryGeometry .
:binormalTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel that
contains the tangents in v." .
:binormalType a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:BinaryGeometry ; dc:description "Specifies the byte format of the binormals." .
:borderColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:TextureProperties ; dc:description "Describes the color to use for border pixels." .
:borderWidth a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:TextureProperties ; dc:description "Describes the number of pixels to use for a
texture border." .
:bottom a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Cone , :Cylinder
, :SlopedCylinder , :Dish , :Snout ; dc:description "For cones, the bottom field
specifies whether the bottom cap of the cone is created. For cylinders, the bottom
```

```
field specifies whether the bottom cap of the cylinder is created. For dishes, it
specifies whether the bottom cap of the dish is created." .
:bottomUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:Background .
:bounce a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [
xsd:maxInclusive "1"^^xsd:float ] ) ] ; rdfs:domain :CollisionCollection ;
dc:description "Indicates how bouncy the surface contact is. 0 indicates no bounce at
all, while 1 indicates maximum bounce." .
:boundaryModeR a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Describes the way R texture coordinate boundaries
are handled." .
:boundaryModeS a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Describes the way S texture coordinate boundaries
are handled." .
:boundaryModeT a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Describes the way T texture coordinate boundaries
are handled." .
:boundaryOpacity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:BoundaryEnhancementVolumeStyle ; dc:description "The boundaryOpacity field specifies
the amount of boundary enhancement to use." .
:brightness a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RadarVolumeStyle ; dc:description "Specifies a multiplier for the RGB values of the
output color." .
:bvhType a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List ;
rdf:first "'jsBIH" ; rdf:rest [ a rdf:List ; rdf:first "BIH" ; rdf:rest [ a rdf:List ;
rdf:first "OCTREE" ;  rdf:rest rdf:nil ] ] ] ] ; rdfs:domain :StaticGroup ;
dc:description "Defines the type of bvh to use." .
:cappingStrength a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
 :ClipPlane ; dc:description "Defines the strength of the capping." .
```

```
:caps a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :RectangularTorus ,
:SolidOfRevolution , :Torus ; dc:description "Specifies whether the side caps exist." .
:ccw a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DPlanarGeometryNode , :X3DSpatialGeometryNode , :GeoElevationGrid ,
:X3DBinaryContainerGeometryNode , :X3DComposedGeometryNode , :X3DGeometryNode ,
:IndexedQuadSet , :LineSet , :Nozzle , :PointSet , :QuadSet ; dc:description "The
ordering of the vertex coordinates of the geometry with respect to user-given or
automatically generated normal vectors used in the lighting model equations." .
:center a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Transform ,
:X3DLODNode , :Plane , :TextureTransform , :TextureTransform3D ; dc:description
"Specifies a translation offset from the origin of the local coordinate system (0,0,0).
For LOD nodes, the center field is a translation offset in the local coordinate system
that specifies the centre of the LOD node for distance calculations." .
:centerOfMass a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:HAnimSegment , :RigidBody .
:centerOfRotation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:GeoViewpoint , :OrthoViewpoint , :Viewpoint ; dc:description "Specifies a center about
which to rotate the user's eyepoint when in EXAMINE mode." .
:child1Url a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :GeoLOD ;
dc:description "When the viewer enters the specified range, this geometry is replaced
with the contents of the four children files defined by child1Url through child4Url.
The four children files are loaded into memory only when the user is within the
specified range. Similarly, these are unloaded from memory when the user leaves this
range." .
:child2Url a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :GeoLOD ;
dc:description "When the viewer enters the specified range, this geometry is replaced
with the contents of the four children files defined by child1Url through child4Url.
The four children files are loaded into memory only when the user is within the
specified range. Similarly, these are unloaded from memory when the user leaves this
range." .
```

```
:child3Url a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :GeoLOD ;
dc:description "When the viewer enters the specified range, this geometry is replaced
with the contents of the four children files defined by child1Url through child4Url.
The four children files are loaded into memory only when the user is within the
specified range. Similarly, these are unloaded from memory when the user leaves this
range." .
:child4Url a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :GeoLOD ;
dc:description "When the viewer enters the specified range, this geometry is replaced
with the contents of the four children files defined by child1Url through child4Url.
The four children files are loaded into memory only when the user is within the
specified range. Similarly, these are unloaded from memory when the user leaves this
range." .
:closed  a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:SplinePositionInterpolator ; dc:description "Specifies whether the interpolator should
provide a closed loop, with continuous velocity vectors as the interpolator transitions
from the last key to the first key." .
:closureType a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a
rdf:List ; rdf:first "PIE" ; rdf:rest [ a rdf:List ; rdf:first "CHORD" ; rdf:rest
rdf:nil ] ] ] ; rdfs:domain :ArcClose2D ; dc:description "The end points of the arc
specified are connected as defined by the closureType field." .
:collideTime a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Collision ;
dc:description "The time of collision." .
:colorIndex a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:IndexedFaceSet , :IndexedLineSet ; dc:description "The index data for the color
 data." .
:colorOffset a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
 :PopGeometry ; dc:description "Offset, given in bytes, for the color attribute inside
 the interleaved attribute array." .
:colorPerVertex a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ElevationGrid , :IndexedFaceSet , :IndexedLineSet , :X3DComposedGeometryNode ;
```

```
dc:description "Determines whether colours specified in the color field are applied to
each vertex or each quadrilateral of the ElevationGrid node. For IndexedFaceSet, if
colorPerVertex is FALSE, colours are applied to each face. If colorPerVertex is true,
colours are applied to each vertex." .
:colorPrecision a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Precision, given in bytes, for the components of the
color attribute." .
:colorSteps a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CartoonVolumeStyle ; dc:description "Specifies how many distinct colors are taken from
the interpolated colors and used to render the object." .
:colorTransparency a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:BlendMode ; dc:description "The constant alpha used by blend modes constant." .
:colorType a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :BinaryGeometry
, :PopGeometry ; dc:description "Specifies the byte format of the colors." .
:colorURL a owl:DatatypeProperty ; rdfs:range xsd:anyURI ; rdfs:domain :BinaryGeometry
; dc:description "The URL to the binary file that contains the colors." .
:colorvector a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
 rdfs:domain :BlendMode , :X3DLightNode , :ColorRGBA , :X3DFogNode .
:compressed a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:BinaryGeometry ; dc:description "Flag that specifies whether the binary files are GZip
compressed." .
:constantForceMix a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBodyCollection ; dc:description "Applies damping to the calculations by violating
the normal constraints by applying a small, constant force to those calculations." .
:contactSurfaceThickness a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBodyCollection ; dc:description "Represents how far bodies may interpenetrate
after a collision. This allows simulation of softer bodies that may deform somewhat
during collision. The default value is zero." .
```

```
:contourStepSize a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:IsoSurfaceVolumeData ; dc:description "Specifies an step size to render isosurfaces
that are multiples of an initial isovalue." .
:convex a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Extrusion ,
:IndexedFaceSet ; dc:description "Indicates whether all polygons in the shape are
convex." .
:coolColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; dc:description
"Specifies the color to be used for surfaces facing away of the light direction." .
:coord a owl:DatatypeProperty ; rdfs:range xsd:anyURI ; rdfs:domain :BinaryGeometry ,
:HAnimSegment , :ImageGeometry , :LineSet , :PointSet ; dc:description "The URL to the
binary file that contains the mesh coordinates." .
:coordIndex a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
 rdfs:domain :HAnimDisplacer , :IndexedLineSet .
:coordType a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :BinaryGeometry
, :PopGeometry ; dc:description "Specifies the byte format of the coordinates." .
:creaseAngle a owl:DatatypeProperty ; rdfs:range xsd:double ; rdfs:domain
:ElevationGrid , :Extrusion , :GeoElevationGrid , :IndexedFaceSet , :SolidOfRevolution
; dc:description "Affects how default normals are generated." .
:crossOrigin a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DBackgroundNode , :X3DEnvironmentTextureNode , :X3DTexture3DNode ,
:GeneratedCubeMapTexture , :X3DTextureNode ; dc:description "Cross Origin Mode." .
:crossSection a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:Extrusion , :SolidOfRevolution ; dc:description "Describes the cross section as a 2D
piecewise linear curve (series of connected vertices)." .
:cutOffAngle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SpotLight ;
dc:description "Specifies the outer bound of the solid angle." .
:cycleInterval a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :TimeSensor ;
dc:description "The cycle of a TimeSensor node lasts for cycleInterval seconds." .
```

```
:cycleTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :TimeSensor ;
dc:description "Can be used for synchronization purposes such as sound with animation."
.
:dbottom a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Snout ;
dc:description "Defines the diameter of the bottom surface." .
:debug a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :StaticGroup ;
dc:description "Enables debugging." .
:depthFunc a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List
; rdf:first "NONE" ; rdf:rest [ a rdf:List ; rdf:first "NEVER" ; rdf:rest [ a rdf:List
; rdf:first "LESS" ; rdf:rest [ a rdf:List ; rdf:first "EQUAL" ; rdf:rest [ a rdf:List
; rdf:first "LEQUAL" ; rdf:rest [ a rdf:List ; rdf:first "GREATER" ; rdf:rest [ a
rdf:List ; rdf:first "NOTEQUAL" ; rdf:rest [ a rdf:List ; rdf:first "GEQUAL" ; rdf:rest
[ a rdf:List ; rdf:first "ALWAYS" ; rdf:rest rdf:nil ] ] ] ] ] ] ] ] ] ] ; rdfs:domain
:DepthMode ; dc:description "The depth function to use." .
:depthMap a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
 rdfs:domain :RenderedTexture ; dc:description "Determines if textures shall be treated
 as depth map." .
:description a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :Anchor ,
:X3DFogNode , :GeoViewpoint .
:desiredAngularVelocity1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "Desired angular velocity for the first axis." .
:desiredAngularVelocity2 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "Desired angular velocity for the second
 axis." .
:destFactor a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a
rdf:List ; rdf:first "none" ; rdf:rest [ a rdf:List ; rdf:first "zero" ; rdf:rest [ a
rdf:List ; rdf:first "one" ; rdf:rest [ a rdf:List ; rdf:first "dst_color" ; rdf:rest
[ a rdf:List ; rdf:first "src_color" ; rdf:rest [ a rdf:List ; rdf:first
"one_minus_dst_color" ; rdf:rest [ a rdf:List ; rdf:first "one_minus_src_color" ;
rdf:rest [ a rdf:List ; rdf:first "src_alpha, one_minus_src_alpha" ; rdf:rest [ a
```

```
rdf:List ; rdf:first "dst_alpha" ; rdf:rest [ a rdf:List ; rdf:first
"one_minus_dst_alpha" ; rdf:rest [ a rdf:List ; rdf:first "src_alpha_saturate" ;
rdf:rest [ a rdf:List ; rdf:first "constant_color" ; rdf:rest [ a rdf:List ; rdf:first
"one_minus_constant_color" ; rdf:rest [ a rdf:List ; rdf:first "constant_alpha" ;
rdf:rest [ a rdf:List ; rdf:first "one_minus_constant_alpha" ; rdf:rest rdf:nil ] ] ] ]
] ] ] ] ] ] ] ] ] ] ] ] ; rdfs:domain :BlendMode ; dc:description "The frame buffer
pixel is scaled according to the method defined by the destination factor." .
:destination a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:ColorChaser , :CoordinateDamper , :X3DEnvironmentNode , :OrientationChaser ,
:OrientationDamper , :PositionChaser , :PositionChaser2D , :PositionDamper ,
:PositionDamper2D , :ScalarChaser , :ScalarDamper , :TexCoordDamper2D ; dc:description
"The target color/orientation value." .
:diameter a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Dish ;
dc:description "Specifies the diameter of the base." .
:diffuseColor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:X3DMaterialNode ; dc:description "Reflects all X3D light sources depending on the
angle of the surface with respect to the light source." .
:diffuseFactor a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of diffuseTexture is multiplied by
this value. If no texture is set, the value is used directly." .
:diffuseTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:CommonSurfaceShader ; dc:description "ChannelMask for diffuseTexture in the form of a
glsl swizzle (e.g., 'rgb', 'a')." .
:diffuseTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for diffuseTexture." .
:diffuseTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
```

```
:dimensions a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DVolumeDataNode , :RenderedTexture ; dc:description "Specifies the size of of the
bounding box for the volume data." .
:direction a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:DirectionalLight , :SpotLight ; dc:description "Specifies the direction vector of the
illumination emanating from the light source in the local coordinate system." .
:disableAngularSpeed a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBody , :RigidBodyCollection ; dc:description "Defines conditions for when the
body ceases to considered as part of the rigid body calculations and should be
considered as at rest." .
:disableLinearSpeed a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBody , :RigidBodyCollection ; dc:description "Defines conditions for when the
body ceases to considered as part of the rigid body calculations and should be
considered as at rest." .
:disableTime a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :RigidBody ,
:RigidBodyCollection ; dc:description "Defines conditions for when the body ceases to
considered as part of the rigid body calculations and should be considered as at
rest." .
:displacementAxis a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:CommonSurfaceShader ; dc:description "Axis along which the vertices are
displacement" .
:displacementFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:CommonSurfaceShader ; dc:description "Factor for the displacement." .
:displacementTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for displacementTexture." .
:displacementTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:displacements a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:HAnimDisplacer .
```

```
:doPickPass a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Scene ;
dc:description "Flag to enable/disable pick pass." .
:drawOrder a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List
; rdf:first "Any" ; rdf:rest [ a rdf:List ; rdf:first "BackToFront" ; rdf:rest [ a
rdf:List ; rdf:first "FrontToBack" ; rdf:rest rdf:nil ] ] ] ] ; rdfs:domain
:ParticleSet ; dc:description "Defines the drawing order for the particles." .
:dtop a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Snout ;
dc:description "Defines the diameter of the top surface." .
:duration a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :X3DChaserNode ,
:ScalarChaser ; dc:description "Duration of the transition." .
:edgeColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:EdgeEnhancementVolumeStyle ; dc:description "Specifies the color to be used for the
edge enhacement." .
:elapsedTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :TimeSensor ;
dc:description "Delivers the current elapsed time since the TimeSensor was activated
and running, cumulative in seconds and not counting any time while in a paused
state." .
:elevationScaling a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:GeoViewpoint ; dc:description "Enable/disable elevation scaled speed for automatic
adjustment of user movement as recommended in spec." .
:emissiveColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DMaterialNode ; dc:description "Models 'glowing' objects." .
:emissiveFactor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of emissiveTexture is multiplied by
this value. If no texture is set, the value is used directly." .
:emissiveTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "ChannelMask for emissiveTexture in
the form of a glsl swizzle (e.g., 'rgb', 'a')." .
```

```
:emissiveTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for emissiveTexture." .
:emissiveTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:enableARC a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ElevationGrid
; dc:description "Adjusts rendering parameters." .
:enableCulling a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:RemoteSelectionGroup ; dc:description "Defines whether culling is used. If culling is
disabled the RemoteSelectionGroup works like a normal group." .
:enableDepthTest a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:DepthMode ; dc:description "Whether the depth test should be enabled or not." .
:enabled a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :AudioClip ,
:X3DComposableVolumeRenderStyleNode , :X3DVolumeRenderStyleNode , :X3DDragSensorNode ,
:ProjectionVolumeStyle , :RigidBody , :RigidBodyCollection ; dc:description "For audio
clips, it specifies whether the clip is enabled. For EdgeEnhancementVolumeStyle, it
specifies whether the render style is enabled or disabled." .
:enabledAxes a owl:DatatypeProperty ; rdfs:range xsd:integer; rdfs:domain :MotorJoint ;
dc:description "The currently enabled axis." .
:endAngle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Arc2D ;
dc:description "The arc extends from the startAngle counterclockwise to the
endAngle." .
:endCap a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Extrusion ;
dc:description "Specifies whether the endCap should exist." .
:environmentFactor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of environmentTexture is multiplied by
this value. If no texture is set, the value is used directly." .
:environmentTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "ChannelMask for environmentTexture
in the form of a glsl swizzle (e.g., 'rgb', 'a')." .
```

```
:environmentTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader .
:environmentTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:equation a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List ;
rdf:first "none" ; rdf:rest [ a rdf:List ; rdf:first "func_add" ; rdf:rest [ a rdf:List
; rdf:first "func_subtract" ; rdf:rest [ a rdf:List ; rdf:first "func_reverse_subtract"
; rdf:rest [ a rdf:List ; rdf:first "min" ; rdf:rest [ a rdf:List ; rdf:first "max" ;
rdf:rest [ a rdf:List ; rdf:first "logic_op" ; rdf:rest rdf:nil ] ] ] ] ] ] ] ] ;
rdfs:domain :BlendMode ; dc:description "An additional equation used to combine source,
destination and the constant value." .
:errorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBodyCollection ; dc:description "Describes how quickly the system should resolve
intersection errors due to floating point inaccuracies." .
:explorationMode a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a
rdf:List ; rdf:first "all" ; rdf:rest [ a rdf:List ; rdf:first "pan" ; rdf:rest [ a
rdf:List ; rdf:first "zoom" ; rdf:rest [ a rdf:List ; rdf:first "rotate" ; rdf:rest [ a
rdf:List ; rdf:first "none" ; rdf:rest rdf:nil ] ] ] ] ] ] ; rdfs:domain
:NavigationInfo ; dc:description "Allows restricting examine and turntable navigation,
overrides mouse buttons." .
:family a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :FontStyle ;
dc:description "Defines the text family." .
:fieldOfView a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:GeoViewpoint , :OrthoViewpoint , :Viewpoint ; dc:description "Preferred minimum
viewing angle from this viewpoint in radians." .
:finalLine a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:MPRVolumeStyle ; dc:description "Specifies the second line to calculate the normal
plane." .
:finiteRotationAxis a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RigidBody ; dc:description "Specifies a vector around which the object rotates." .
```

```
:first a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :TimeSensor .
:firstCycle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :TimeSensor .
:fixed a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :RigidBody ;
dc:description "Indicates that this body does not move." .
:fogType a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List ;
rdf:first "LINEAR" ; rdf:rest [ a rdf:List ; rdf:first "EXPONENTIAL" ; rdf:rest rdf:nil
] ] ] ; rdfs:domain :X3DFogNode ; dc:description "Controls how much of the fog colour
is blended with the object as a function of distance." .
:forceOutput a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DRigidJointNode ; dc:description "Controls which output fields are to be generated
for the next frame." .
:forceTransitions a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DLODNode ; dc:description "Specifies whether browsers are allowed to disregard level
distances in order to provide better performance." .
:forces a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :RigidBody ;
dc:description "The torques and forces fields define zero or more sets of torque and
force values that are applied to the object every frame. These are continuously applied
until reset to zero by the user." .
:format a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :RefinementTexture
; dc:description "Specifies the image format of the dataset." .
:fraction_changed a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:TimeSensor ; dc:description "Outputs a floating point value in the closed interval [0,
1]." .
:fresnelBlend a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:CommonSurfaceShader ; dc:description "To what degree the Fresnel equation for
dielectrics should be used to blend the perfect specular reflection and
transmission." .
:frictionCoefficients a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CollisionCollection ; dc:description "Friction coefficients." .
:frontUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
```

```
rdfs:domain :Background .
:frustumCulling a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ElevationGrid ; dc:description "If TRUE, objects outside the viewing frustum are
ignored." .
:gammaCorrectionDefault a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:ElevationGrid ; dc:description "The gamma correction to apply by default." .
:generateMipMaps a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:TextureProperties ; dc:description "Describes whether mipmaps should be generated for
the texture. " .
:geoCoords a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:GeoLocation ; dc:description "The geometry of the nodes in children is to be specified
in units of metres in X3D coordinates relative to the location specified by the
geoCoords field. The geoCoords field can be used to dynamically update the geospatial
location of the model." .
:geoGridOrigin a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:GeoElevationGrid ; dc:description "Specifies the geographic coordinate for the south-
west corner (bottom-left) of the dataset." .
:geoSystem a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:GeoElevationGrid , :GeoLocation , :GeoOrigin , :GeoPositionInterpolator ,
:GeoViewpoint ; dc:description "Defines the spatial reference frame." .
:global a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :X3DLightNode ;
dc:description "Specifies whether the light is global or scoped." .
:globalGeoOrigin a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:GeoTransform ; dc:description "Specifies whether a GeoOrigin should be applied to
child nodes." .
:gradientThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:EdgeEnhancementVolumeStyle ; dc:description "The gradientThreshold field is used to
adjust the edge detection." .
:gravity a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RigidBodyCollection ; dc:description "Indicates direction and strength (in
```

```
acceleration base units) of the local gravity vector for this collection of bodies. The
default gravity is standard earth gravity of 9.8 meters/second2 downwards." .
:groundAngle a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBackgroundNode ; dc:description "Angle of the ground." .
:groundColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBackgroundNode ; dc:description "Color of the ground." .
:hasHelperColors a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Box ;
dc:description "Specifies whether helper colors should be used, which will color each
vertex with a different color." .
:headlight a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :GeoViewpoint
, :NavigationInfo ; dc:description "Enable/disable directional light that always points
in the direction the user is looking." .
:height a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Cone , :Dish
, :Extrusion , :GeoElevationGrid , :Nozzle , :Pyramid , :RectangularTorus ,
:SlopedCylinder , :Snout ; dc:description "Specifies the height of the cone from the
centre of the base to the apex, or defines the maximum height of the dished surface
above the base. For a Nozzle, it defines the height of the pipe. For a pyramid, it
defines the distance between the bottom and the top faces." .
:hideChildren a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Texture ;
dc:description "Specifies whether the children are shown or hidden outside the
texture." .
:horizontal a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :FontStyle ;
dc:description "Specifies whether the text is shown horizontally or vertically." .
:idsPerVertex a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:BinaryGeometry ; dc:description "Flag that specifies whether vertex IDs are given as
texture coordinates." .
:image a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :PixelTexture
; dc:description "The image that delivers the pixel data." .
:implicitMeshSize a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:ImageGeometry .
```

```
:index a owl:DatatypeProperty ; rdfs:range xsd:anyURI ; rdfs:domain :BinaryGeometry ,
:ImageGeometry , :ParticleSet ; dc:description "The URL to the binary file that
contains the index data." .
:indexType a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :BinaryGeometry
; dc:description "Specifies the byte format of the index data." .
:indexedRendering a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:PopGeometry ; dc:description "Specifies whether this PopGeometry was encoded for
indexed rendering." .
:inertia a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :RigidBody ;
dc:description "The inertia field represents a 3x2 inertia tensor matrix. If the set
values are less than six items, the results are implementation dependent. If the value
set is greater than six values, only the first six values of the array are used." .
:info a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :HAnimHumanoid
, :WorldInfo .
:initialDestination a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:ColorChaser , :CoordinateDamper , :OrientationChaser , :OrientationDamper ,
:PositionChaser , :PositionChaser2D , :PositionDamper , :PositionDamper2D ,
:ScalarChaser , :ScalarDamper , :TexCoordDamper2D ; dc:description "The same value as
initialValue unless a transition to a certain value is to be created right after the
scene is loaded or right after the ColorChaser/CoordinateChaser node is created
dynamically." .
:initialValue a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:ColorChaser , :CoordinateDamper , :OrientationChaser , :OrientationDamper ,
:PositionChaser , :PositionChaser2D , :PositionDamper , :PositionDamper2D ,
:ScalarChaser , :ScalarDamper , :TexCoordDamper2D ; dc:description "Set the initial
value." .
:initialVisibility a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a
rdf:List ; rdf:first "auto" ; rdf:rest [ a rdf:List ; rdf:first "visible" ; rdf:rest [
a rdf:List ; rdf:first "invisible" ; rdf:rest rdf:nil ] ] ] ] ; rdfs:domain :MultiPart
; dc:description "Set the initial visibility." .
```

```
:innerRadius a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Disk2D ,
:Nozzle , :RectangularTorus , :Torus ; dc:description "Specifies the inner dimension of
the Disk2D/pipe/torus." .
:insideOutsideRadius a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:Torus ; dc:description "Use a different interpretation mode for the inside and outside
radius." .
:intensity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :X3DLightNode ;
dc:description "Specifies the brightness of the direct emission from the light." .
:intensityLimits a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RadarVolumeStyle ; dc:description "Scales voxel intensities lying in
(intensityLimits[0] ~ intensityLimits[1]) to (0.0 ~ 1.0)." .
:intensityThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ProjectionVolumeStyle ; dc:description "Defines a local maximum or minimum value along
the ray traversal. It is ignored on the AVERAGE intensity projection." .
:interpupillaryDistance a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RenderedTexture ; dc:description "Sets the eye distance for stereo rendering." .
:invertAlphaTexture a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:CommonSurfaceShader ; dc:description "If true, (1-sampledValue) is used as alpha. If
false the sampled value is used." .
:invisibleNodes a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RemoteSelectionGroup ; dc:description "Defines a set of labels to disable nodes. The
label must include the prefix." .
:isActive a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DBackgroundNode , :Collision , :X3DDamperNode , :X3DEnvironmentNode , :X3DFogNode ,
:X3DViewpointNode , :X3DNavigationInfoNode , :OrientationChaser , :X3DChaserNode ,
:X3DDamperNode , :ScalarChaser , :TimeSensor .
:isPaused a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :TimeSensor ;
dc:description "Outputs whether the timer is paused." .
```

```
:isPickable a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DVolumeDataNode , :MultiPart , :X3DShapeNode ; dc:description "Defines whether the
shape is pickable." .
:isoSurfaceCutoffValue a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RadarVolumeStyle ; dc:description "Specifies a threshold such that data with intensity
below this is ignored during ray casting." .
:iterations a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:RefinementTexture , :RigidBodyCollection ; dc:description "Maximum level that should
be loaded." .
:justify a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :FontStyle ;
dc:description "Specifies where the text is anchored." .
:key a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DInterpolatorNode , :GeoPositionInterpolator , :NormalInterpolator ; dc:description
"The list of key times." .
:keyVelocity a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:SplinePositionInterpolator ; dc:description "Defines the set of velocity vectors, that
are used for interpolation." .
:keyvalue a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:ColorInterpolator , :CoordinateInterpolator , :GeoPositionInterpolator ,
:NormalInterpolator , :OrientationInterpolator , :PositionInterpolator ,
:ScalarInterpolator , :SplinePositionInterpolator ; dc:description "The set of data
points used for interpolation." .
:label a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RemoteSelectionGroup ; dc:description "Defines a list of subsequent id/object
pairs." .
:language a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :X3DShaderNode ,
:FontStyle ; dc:description "Indicates to the browser which shading language is used
for the source file(s). For FontStyle, it specifies the context of the language for the
text string in the form of a language and a country in which that language is used." .
```

```
:latitude a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:SphereSegment ; dc:description "Defines an array of latitude values." .
:leftToRight a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :FontStyle ;
dc:description "Specifies whether the text is shown from left to right or from right to
left." .
:leftUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
rdfs:domain :Background .
:length a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Text ;
dc:description "Specifies the length of each text string in the local coordinate
system." .
:lightFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:OpacityMapVolumeStyle ; dc:description "A factor to specify the amount of global light
to be considered on each sampled point along the ray traversal." .
:lighting a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ShadedVolumeStyle .
:limitOrientation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:HAnimJoint .
:lineSegments a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:Polyline2D ; dc:description "Specifies the vertices to be connected." .
:linearDampingFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBody ; dc:description "Linear damping factor." .
:linearVelocity a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RigidBody ; dc:description "Linear velocity." .
:linetype a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain :LineProperties
; dc:description "Selects a line pattern." .
:linewidthScaleFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:LineProperties ; dc:description "A multiplicative value that scales a the
linewidth." .
:lit a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DPlanarGeometryNode , :X3DBinaryContainerGeometryNode , :Box ,
```

```
:X3DSpatialGeometryNode , :GeoElevationGrid , :IndexedFaceSet , :X3DGeometryNode ,
:X3DComposedGeometryNode , :PointSet , :X3DPlanarGeometryNode ; dc:description
"Specifies whether this geometry should be rendered with or without lighting." .
:llimit a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :HAnimJoint .
:load a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Inline ;
dc:description "Specifies whether the X3D file specified by the url field is loaded." .
:location a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :PointLight
, :SpotLight ; dc:description "The position of the Light." .
:loop a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :AudioClip ,
:MovieTexture , :TimeSensor ; dc:description "Specifies whether the clip/timer cycle
loops when finished." .
:lowPriorityCulling a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ElevationGrid ; dc:description "If TRUE and occlusion culling supported, only
threshold fraction of objects, sorted by their screen space coverage, are rendered." .
:lowPriorityFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Factor of low priority culling for controlling speed-
performance trade-off." .
:lowPriorityThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Only threshold fraction of objects, sorted by their
screen space coverage, are rendered" .
:magnificationFilter a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Describes the way textures are filtered when the
image is smaller than the screen space representation." .
:mapDEFToID a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Inline ;
dc:description "Specifies whether the DEF value is used as id when no other id is
set." .
:maskA a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ColorMaskMode ;
dc:description "Masks a color channel." .
:maskB a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ColorMaskMode ;
dc:description "Masks b color channel." .
```

```
:maskG a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ColorMaskMode ;
dc:description "Masks g color channel." .
:maskR a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ColorMaskMode ;
dc:description "Masks r color channel." .
:mass a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :HAnimSegment ,
:RigidBody .
:matrix a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:MatrixTransform , :TextureTransformMatrix3D ; dc:description "Defines the
transformation matrix." .
:maxAngle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SingleAxisHingeJoint ; dc:description "The minAngle and maxAngle fields are used to
control the maximum angles through which the hinge is allowed to travel. A hinge may
not travel more than π radians (or the equivalent angle base units) in either direction
from its initial position." .
:maxAngle1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The minAngle1 and maxAngle1 fields are used to
control the maximum angles through which the hinge is allowed to travel. A hinge may
not travel more than π radians (or the equivalent angle base units) in either direction
from its initial position." .
:maxBBSize a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:PopGeometry ; dc:description "The size of the bounding box used to quantize data in
this geometry, which is usually the largest bounding box of all sub-meshes of a given
mesh." .
:maxCorrectionSpeed a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RigidBodyCollection ; dc:description "Maximal correction speed." .
:maxDepth a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain :StaticGroup .
:maxExtent a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Text ;
dc:description "Limits and compresses all of the text strings if the length of the
maximum string is longer than the maximum extent, as measured in the local coordinate
```

```
system. If the text string with the maximum length is shorter than the maxExtent, then
there is no compressing." .
:maxFrameRate a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Defines maximal target frame rate for dynamic moments
and quality-speed trade-off." .
:maxLevel a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:RefinementTexture ; dc:description "Maximum level that should be loaded." .
:maxObjectsPerNode a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:StaticGroup .
:maxPosition a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:PlaneSensor ; dc:description "The minPosition and maxPosition fields allow to
constrain the 2D output of the plane sensor, along each 2D component. If the value of a
component in maxPosition is smaller than the value of a component in minPosition,
output is not constrained along the corresponding direction." .
:maxPrecisionLevel a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Maximum precision level of this PopGeometry node. This
can be used to clamp displayed precision - if the value is -1, no clamping takes
place." .
:maxRenderedIds a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:RemoteSelectionGroup ; dc:description "Sets the maximum number of items that are
rendered." .
:maxSeparation a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SliderJoint
; dc:description "If minSeparation is greater than maxSeparation, the stops become
ineffective as if the object has no stops at all." .
:maxTorque1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The maxTorque1 field defines the maximum amount
of torque that the motor can apply on axis 1 in order to achieve the desired
desiredAngularVelocity1 value." .
:maxTorque2 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The maxTorque2 field defines the maximum amount
```

```
of torque that the motor can apply on axis 1 in order to achieve the desired
desiredAngularVelocity2 value." .
:metadata a owl:DatatypeProperty ; rdfs:domain :X3DNode ; dc:description "Metadata
information" .
:minAngle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SingleAxisHingeJoint ; dc:description "The minAngle and maxAngle fields are used to
control the maximum angles through which the hinge is allowed to travel. A hinge may
not travel more than π radians (or the equivalent angle base units) in either direction
from its initial position." .
:minAngle1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The minAngle1 and maxAngle1 fields are used to
control the maximum angles through which the hinge is allowed to travel. A hinge may
not travel more than π radians (or the equivalent angle base units) in either direction
from its initial position." .
:minBounceSpeed a owl:DatatypeProperty ; rdfs:range[ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [
xsd:maxExclusive "+INF"^^xsd:float ] ) ] ; rdfs:domain :CollisionCollection ;
dc:description "Indicates the minimum speed, in speed base units, that an object shall
have before an object will bounce." .
:minFrameRate a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Defines minimal target frame rate for static moments
and quality-speed trade-off." .
:minPosition a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:PlaneSensor ; dc:description "The minPosition and maxPosition fields allow to
constrain the 2D output of the plane sensor, along each 2D component. If the value of a
component in maxPosition is smaller than the value of a component in minPosition,
output is not constrained along the corresponding direction." .
:minPrecisionLevel a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Minimum precision level of this PopGeometry node. This
```

```
can be used to clamp displayed precision - if the value is -1, no clamping takes
place." .
:minRelativeBBoxSize a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:StaticGroup .
:minSeparation a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SliderJoint
; dc:description "If minSeparation is greater than maxSeparation, the stops become
ineffective as if the object has no stops at all." .
:minificationFilter a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Describes the way textures are filtered when the
image is larger than the screen space representation." .
:mode a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List ;
rdf:first "ViewDirQuads" ; rdf:rest [ a rdf:List ; rdf:first "Points" ; rdf:rest [ a
rdf:List ; rdf:first "Lines" ; rdf:rest [ a rdf:List ; rdf:first "Arrows" ; rdf:rest [
a rdf:List ; rdf:first "ViewerArrows" ; rdf:rest [ a rdf:List ; rdf:first "ViewerQuads"
; rdf:rest [ a rdf:List ; rdf:first "Rectangles" ; rdf:rest rdf:nil ] ] ] ] ] ] ] ] ;
rdfs:domain :ParticleSet ; dc:description "Drawing mode: ViewDirQuads - Draws quads
directed to the viewpoint (default). Points - Draw points. Lines - Draw lines. These
modes must not match the finally supported modes." .
:modelview a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:Viewfrustum ; dc:description "Camera modelview matrix." .
:momentsOfInertia a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:HAnimSegment .
:motor1Axis a owl:DatatypeProperty ; rdfs:range xsd:complexType; rdfs:domain
:MotorJoint ; dc:description "Defines the axis vector of the corresponding axis." .
:motor2Axis a owl:DatatypeProperty ; rdfs:range xsd:complexType; rdfs:domain
:MotorJoint ; dc:description "Defines the axis vector of the corresponding axis." .
:motor3Axis a owl:DatatypeProperty ; rdfs:range xsd:complexType; rdfs:domain
:MotorJoint ; dc:description "Defines the axis vector of the corresponding axis." .
:name a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :CADAssembly ,
:X3DField , :HAnimDisplacer , :X3DMetadataObject .
```

```
:nameSpaceName a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Block
, :Inline .
:navType a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:GeoViewpoint ; dc:description "Specifies the navigation type." .
:normal a owl:DatatypeProperty ; rdfs:range xsd:anyURI ; rdfs:domain :BinaryGeometry ,
:X3DComposedGeometryNode , :ParticleSet ; dc:description "The URL to the binary file
that contains the normals." .
:normalAsSphericalCoordinates a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
rdfs:domain :BinaryGeometry ; dc:description "Specifies whether the normals are encoded
as spherical coordinates." .
:normalBias a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "Bias to apply to normal sampled from
normalTexture" .
:normalFormat a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a
rdf:List ; rdf:first "UNORM" ; rdf:rest rdf:nil ] ] ; rdfs:domain :CommonSurfaceShader
; dc:description "Determines how normals are stored in normalTexture." .
:normalIndex a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:IndexedFaceSet ; dc:description "The index data for the normal data." .
:normalOffset a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Offset, given in bytes, for the normal attribute inside
the interleaved attribute array." .
:normalPerVertex a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:BinaryGeometry , :ElevationGrid , :IndexedFaceSet , :X3DComposedGeometryNode ;
dc:description "Specifies whether normals are stored per vertex or per face." .
:normalPrecision a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Precision, given in bytes, for the components of the
normal attribute." .
:normalScale a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "Scale to apply to normal sampled from
normalTexture" .
```

```
:normalSpace a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a
rdf:List ; rdf:first "TANGENT" ; rdf:rest [ a rdf:List ; rdf:first "OBJECT" ; rdf:rest
rdf:nil ] ] ] ; rdfs:domain :CommonSurfaceShader ; dc:description "Space in which
normals in normalTexture are defined." .
:normalTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:CommonSurfaceShader ; dc:description "ChannelMask for normalTexture in the form of a
glsl swizzle (e.g., 'rgb', 'a')." .
:normalTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for normalTexture." .
:normalTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:normalType a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :BinaryGeometry
, :PopGeometry ; dc:description "Specifies the byte format of the normals." .
:normalUpdateMode a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DComposedGeometryNode .
:normalizeVelocity a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:SplinePositionInterpolator ; dc:description "Specifies whether the velocity vectors
are to be transformed into normalized tangency vectors." .
:nozzleHeight a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Nozzle ;
dc:description "Defines the height of the nozzle." .
:nozzleRadius a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Nozzle ;
dc:description "Defines the radius of the nozzle." .
:numAnchorVertices a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Number of anchor vertices (can be 0). Anchor vertices
are used to keep some vertices on the bordes between sub-meshes fixed during
refinement." .
:numColorComponents a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:ImageGeometry ; dc:description "Specifies the number of color components." .
```

```
:numComponents a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:FloatVertexAttribute ; dc:description "Specifies how many consecutive floating point
values should be grouped together per vertex." .
:numTexCoordComponents a owl:DatatypeProperty ; rdfs:range xsd:positiveInteger ;
rdfs:domain :ImageGeometry ; dc:description "Specifies the number of texture coordinate
components." .
:numberOfMaxSegments a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SegmentedVolumeData ; dc:description "Specifies the number of segments on the volume
data. It is used to correctly match each segment identifier to an index of the
renderStyle list." .
:numberOfSlices a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:ImageTextureAtlas ; dc:description "Specifies the number of slices in the texture
atlas." .
:occlusionCulling a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ElevationGrid ; dc:description "If TRUE and occlusion culling supported, objects
occluding less than the threshold below are ignored." .
:occlusionVisibilityFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Factor of occlusion culling for controlling speed-
performance trade-off." .
:occlusionVisibilityThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ;
rdfs:domain :ElevationGrid ; dc:description "Objects occluding less than the threshold
below are ignored." .
:oculusRiftVersion a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RenderedTexture ; dc:description "Changes between DK1 and DK2." .
:offset a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :CylinderSensor ,
:PlaneSensor , :SphereSensor .
:on a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ClipPlane ,
:X3DLightNode .
```

```
:onGreatCircle a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:GeoPositionInterpolator ; dc:description "Specifies whether coordinates will be
interpolated along a great circle path." .
:opacityFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:BoundaryEnhancementVolumeStyle , :OpacityMapVolumeStyle ; dc:description "An exponent
factor that specifies the slope of the opacity curve to highlight the boundary." .
:order a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain :X3DDamperNode ;
dc:description "Specifies the smoothness of the transition." .
:orientation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:Extrusion , :GeoViewpoint , :OrthoViewpoint , :RigidBody , :Viewpoint ; dc:description
"Defines an array of orientations for each extrusion step." .
:origChannelCount a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:X3DEnvironmentTextureNode , :X3DTextureNode , :GeneratedCubeMapTexture ,
:X3DTexture3DNode ; dc:description "Specifies the channel count of the texture. 0 means
the system should figure out the count automatically." .
:originLine a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:MPRVolumeStyle ; dc:description "Specifies the base line of the slice plane." .
:originalVertexCount a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Vertex count at the highest possible level of
precision." .
:orthogonalColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CartoonVolumeStyle ; dc:description "Specifies the color to be used when the surface
normal is perpendicular to the view direction" .
:outerRadius a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Disk2D ,
:RectangularTorus , :Nozzle ; dc:description "Specifies the radius of the outer
dimension of the Disk2D/torus/pipe." .
:parallelColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CartoonVolumeStyle ; dc:description "Specifies the color to be used when the surface
normal is parallel to the view direction." .
```

```
:parameter a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Anchor ;
dc:description "Passed parameter that signals web browser how to redirect url
loading" .
:pauseTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :MovieTexture ,
:TimeSensor ; dc:description "Sets a time to pause the video/timer." .
:phaseFunction a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:ShadedVolumeStyle .
:pickMode a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :Scene ;
dc:description "The picking mode for the scene." .
:pitch a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :MovieTexture .
:plane a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :ClipPlane ;
dc:description "Specifies a single plane equation that will be used to clip the
geometry." .
:point a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Coordinate ,
:GeoCoordinate , :Polyline2D , :TextureCoordinate , :TextureCoordinate3D ;
dc:description "Contains the 3D coordinates." .
:position a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBinaryContainerGeometryNode , :GeoViewpoint , :OrthoViewpoint , :RigidBody ,
:Viewpoint .
:positionLine a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:MPRVolumeStyle ; dc:description "Specifies the position along the line where the slice
plane is rendered." .
:positionOffset a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Offset, given in bytes, for the position attribute
inside the interleaved attribute array." .
:positionPrecision a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Precision, given in bytes, for the components of the
position attribute." .
```

```
:precisionFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:PopGeometry ; dc:description "Additional precision multiplication factor, for tuning
the displayed precision." .
:preferAccuracy a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:RigidBodyCollection ; dc:description "Provides a performance hint to the underlying
evaluation about whether the user prefers to have very accurate models or fast models.
Accuracy comes at a large penalty in both speed and memory usage, but may not be needed
most of the time. The default setting is to optimize for speed rather than accuracy." .
:primType a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBinaryContainerGeometryNode , :Mesh , :Plane .
:projection a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:Viewfrustum ; dc:description "Camera projection matrix." .
:quality a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :FontStyle ;
dc:description "Sets the quality of the text rendering as an oversampling factor." .
:radius a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [xsd:minExclusive "0"^^xsd:float] [xsd:maxExclusive
"+INF"^^xsd:float] ) ] ; rdfs:domain :Arc2D , :Dish , :PointLight , :SlopedCylinder ,
:SphereSegment , :SpotLight ; dc:description "The radius of the circle of which the arc
is a portion, or defines the radius of the third semi-principal axes of the ellipsoid.
For sphere segments, it defines the radius of the sphere." .
:range a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :LOD , :GeoLOD
; dc:description "Camera-to-object distance transitions for each child level, where
range values go from near to far (LOD), or the level of detail is switched depending
upon whether the user is closer or farther than range length base units from the
geospatial coordinate center (GeoLOD)." .
:readOnly a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :DepthMode ;
dc:description "Whether the depth buffer is enabled for writing or not." .
:reconnect a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:RemoteSelectionGroup ; dc:description "Sets whether a reconnect is attempted on a
connection loss." .
```

```
:reference a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DMetadataObject ; dc:description "Identifies the metadata standard or other
specification that defines the name field." .
:referenceBindableDescription a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :GeoLOD .
:reflectionFactor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of reflectionTexture is multiplied by
this value. If no texture is set, the value is used directly." .
:reflectionTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "ChannelMask for reflectionTexture in
the form of a glsl swizzle (e.g., 'rgb', 'a')." .
:reflectionTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for reflectionTexture." .
:reflectionTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:relativeIndexOfRefraction a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:CommonSurfaceShader ; dc:description "Relative IOR for perfect specular component." .
:render a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :X3DGroupingNode
; dc:description "Flag to enable/disable rendering." .
:render a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :X3DLODNode ,
:X3DTransformNode , :X3DGroupingNode , :X3DVolumeDataNode , :X3DTransformNode ,
:X3DShapeNode ; dc:description "Flag to enable/disable rendering" .
:renderStyleEnabled a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DComposableVolumeRenderStyleNode ; dc:description "Specifies whether the render
style is enabled or disabled." .
:repeatS a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DEnvironmentTextureNode , :X3DTextureNode , :X3DTexture3DNode ; dc:description
"Specifies whether the texture is repeated in s direction." .
```

```
:repeatT a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DEnvironmentTextureNode , :X3DTextureNode , :X3DTexture3DNode ; dc:description
"Specifies whether the texture is repeated in t direction." .
:resumeTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :MovieTexture ,
:TimeSensor ; dc:description "Sets a time to resume from pause." .
:retainedOpacity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:BoundaryEnhancementVolumeStyle ; dc:description "Specifies the amount of original
opacity to retain." .
:rgbaColors a owl:DatatypeProperty ; rdfs:range xsd:boolean ; dc:description "Enables
RGBA colors." .
:rightUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ;
rdfs:domain :Background .
:rootUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :GeoLOD ;
dc:description "The rootUrl and rootNode fields provide two different ways to specify
the geometry of the root tile. You may use one or the other. The rootNode field lets
you include the geometry for the root tile directly within the X3D file." .
:rotateYUp a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :GeoOrigin ;
dc:description "Specify whether coordinates of nodes that use this GeoOrigin are to be
rotated such that their up direction is aligned with the X3D Y axis." .
:rotation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Transform
, :TextureTransform , :TextureTransform3D ; dc:description "Specifies a rotation of the
coordinate system." .
:scale a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Transform ,
:X3DEnvironmentTextureNode , :X3DTexture3DNode , :Extrusion , :HAnimHumanoid ,
:X3DTextureNode , :TextureTransform , :TextureTransform3D ; dc:description "Specifies a
non-uniform scale of the coordinate system." .
:scaleOrientation a owl:DatatypeProperty ; rdfs:domain :Transform , :GeoTransform ,
:HAnimHumanoid , :HAnimSite , :TextureTransform3D ; dc:description "Specifies a
rotation of the coordinate system before the scale (to specify scales in arbitrary
orientations)." .
```

```
:scaleRenderedIdsOnMove a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RemoteSelectionGroup ; dc:description "Sets the scaling factor to reduce the number of
render calls during navigation." .
:scaleTexture a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Texture ;
dc:description "Specifies whether the texture is scaled to the next highest power of
two." .
:set_fraction a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:X3DInterpolatorNode , :GeoPositionInterpolator ; dc:description "The set_fraction
inputOnly field receives an SFFloat event and causes the interpolator node function to
evaluate, resulting in a value_changed output event of the specified type with the same
timestamp as the set_fraction event." .
:shadowCascades a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:DirectionalLight , :SpotLight .
:shadowExcludeTransparentObjects a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
rdfs:domain :ElevationGrid ; dc:description "Transparent objects like glass do not
throw much shadow, enable this IR convenience flag with TRUE." .
:shadowFilterSize a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:DirectionalLight , :X3DLightNode ; dc:description "Sets the smoothness of the shadow
umbra." .
:shadowIntensity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:X3DLightNode ; dc:description "Defines the attenuation of the shadows." .
:shadowMapSize a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:X3DLightNode ; dc:description "Specifies the resolution of the used shadow map." .
:shadowObjectIdMapping a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:Scene ; dc:description "The url of the shadow object id mapping." .
:shadowOffset a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :X3DLightNode
; dc:description "Defines the shadow offset for the back projection of the shadow
map." .
:shadowSplitFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DirectionalLight , :SpotLight .
```

```
:shadowSplitOffset a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DirectionalLight , :SpotLight .
:shadows a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ShadedVolumeStyle .
:shininess a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :X3DMaterialNode
; dc:description "Determines the specular highlights." .
:shininessFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of shininessTexture is multiplied by
this value. If no texture is set, the value is used directly." .
:shininessTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "ChannelMask for shininessTexture in
the form of a glsl swizzle (e.g., 'rgb', 'a')." .
:shininessTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for shininessTexture." .
:shininessTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:showDebugBoxVolumes a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:StaticGroup .
:showNormals a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:RenderedTexture ; dc:description "Specifies whether normals are shown." .
:side a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Cone ;
dc:description "Specifies whether sides of the cone are created." .
:silhouetteBoundaryOpacity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SilhouetteEnhancementVolumeStyle ; dc:description "A factor to specify the amount of
silhouette enhancement to use." .
:silhouetteRetainedOpacity a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SilhouetteEnhancementVolumeStyle ; dc:description "A factor to specify the amount of
original opacity to retain." .
```

```
:silhouetteSharpness a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SilhouetteEnhancementVolumeStyle ; dc:description "An exponent factor to specify the
silhouette sharpness." .
:size a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBinaryContainerGeometryNode , :DynamicLOD , :Box , :FontStyle , :ParticleSet ,
:Plane , :X3DPlanarGeometryNode .
:skinCoordIndex a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:HAnimJoint .
:skinCoordWeight a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:HAnimJoint .
:skyAngle a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBackgroundNode ; dc:description "Angle of the sky" .
:skyColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBackgroundNode ; dc:description "Color of the sky." .
:slicesOverX a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:ImageTextureAtlas ; dc:description "Specifies the slices in x." .
:slicesOverY a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:ImageTextureAtlas ; dc:description "Specifies the slices in y." .
:slipFactors a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CollisionCollection ; dc:description "Slip factors." .
:smallFeatureCulling a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:ElevationGrid ; dc:description "If TRUE, objects smaller than the threshold below are
ignored." .
:smallFeatureFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Factor of small feature culling for controlling speed-
performance trade-off." .
:smallFeatureThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Objects smaller than the threshold below are
ignored." .
```

```
:SSAO a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ElevationGrid ;
dc:description "Flag to enable Screen Space Ambient Occlusion." .
:SSAOamount a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :ElevationGrid
; dc:description "Value that determines the amount of contribution of SSAO (from 0 to
1)." .
:SSAOblurDepthTreshold a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Determines the maximum depth difference for the SSAO
blurring pass." .
:SSAOradius a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :ElevationGrid
; dc:description "Determines the radius in which the SSAO is sampled in world space." .
:SSAOrandomTextureSize a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Determines the size of the random texture used for
sparse sampling of SSAO." .
:softnessConstantForceMix a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:CollisionCollection ; dc:description "A constant force value to make the colliding
surfaces appear to be soft." .
:softnessErrorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:CollisionCollection ; dc:description "Determines how much of the collision error
should be fixed in a set of evaluations." .
:solid a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :Arc2D ,
:X3DBinaryContainerGeometryNode , :X3DSpatialGeometryNode , :Dish ,
:X3DComposedGeometryNode , :X3DGeometryNode , :X3DComposedGeometryNode , :PointSet ,
:X3DPlanarGeometryNode , :Plane ; dc:description "Specifies whether backface-culling is
used. If solid is TRUE, only front-faces are drawn." .
:sortKey a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain :MultiPart ;
dc:description "Change render order manually." .
:sortTrans a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :ElevationGrid
; dc:description "If TRUE, transparent objects are sorted from back to front (allows
explicitly disabling sorting)." .
```

```
:sortType a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List ;
rdf:first "auto" ; rdf:rest [ a rdf:List ; rdf:first "transparent" ; rdf:rest [ a
rdf:List ; rdf:first "opaque" ; rdf:rest rdf:nil ] ] ] ] ; rdfs:domain :Appearance ,
:MultiPart ; dc:description "Defines the shape type for sorting." .
:spacing a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :FontStyle ;
dc:description "Sets the spacing between lines of text, relative to the text size." .
:specularColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DMaterialNode ; dc:description "Determine the specular highlights." .
:specularFactor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of specularTexture is multiplied by
this value. If no texture is set, the value is used directly." .
:specularTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "ChannelMask for specularTexture in
the form of a glsl swizzle (e.g., 'rgb', 'a')." .
:specularTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for specularTexture." .
:specularTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:speed a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :MovieTexture ,
:NavigationInfo ; dc:description "Specifies the plaback speed." .
:speedFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :GeoViewpoint
; dc:description "The speedFactor field of the GeoViewpoint node is used as a
multiplier to the elevation-based velocity that the node sets internally; i.e., this is
a relative value and not an absolute speed as is the case for the NavigationInfo
node." .
:sphericalNormals a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:PopGeometry ; dc:description "Specifies whether this PopGeometry was encoded for
rendering with spherical normals." .
```

```
:spine a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Extrusion ;
dc:description "Describes the spine as a 3D piecewise linear curve (series of
conntected vertices)." .
:srcFactor a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List
; rdf:first "none" ; rdf:rest [ a rdf:List ; rdf:first "zero" ; rdf:rest [ a rdf:List ;
rdf:first "one" ; rdf:rest [ a rdf:List ; rdf:first "dst_color" ; rdf:rest [ a rdf:List
; rdf:first "src_color" ; rdf:rest [ a rdf:List ; rdf:first "one_minus_dst_color" ;
rdf:rest [ a rdf:List ; rdf:first "one_minus_src_color" ; rdf:rest [ a rdf:List ;
rdf:first "src_alpha, one_minus_src_alpha" ; rdf:rest [ a rdf:List ; rdf:first
"dst_alpha" ; rdf:rest [ a rdf:List ; rdf:first "one_minus_dst_alpha" ; rdf:rest [ a
rdf:List ; rdf:first "src_alpha_saturate" ; rdf:rest [ a rdf:List ; rdf:first
"constant_color" ; rdf:rest [ a rdf:List ; rdf:first "one_minus_constant_color" ;
rdf:rest [ a rdf:List ; rdf:first "constant_alpha" ; rdf:rest [ a rdf:List ; rdf:first
"one_minus_constant_alpha" ; rdf:rest rdf:nil ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ;
rdfs:domain :BlendMode ; dc:description "The incoming pixel is scaled according to the
method defined by the source factor." .
:stamp0 a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :RefinementTexture
; dc:description "Specifies the first stamp texture." .
:stamp1 a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :RefinementTexture
; dc:description "Specifies the second stamp texture." .
:startAngle a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Arc2D ;
dc:description "The arc extends from the startAngle counterclockwise to the
endAngle." .
:startTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :MovieTexture ,
:TimeSensor ; dc:description "Sets a start time for the video/timer." .
:stepSize a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:SphereSegment ; dc:description "Defines an array of stepSizes." .
:stereoMode a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List
; rdf:first "NONE" ; rdf:rest [ a rdf:List ; rdf:first "LEFT_EYE" ; rdf:rest [ a
```

```
rdf:List ; rdf:first "RIGHT_EYE" ; rdf:rest rdf:nil ] ] ] ] ; rdfs:domain
:RenderedTexture ; dc:description "Sets render information for stereo rendering." .
:stop1Bounce a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "Describes how much the joint should bounce the body back on the
corresponding axis if the joint limit has been reached or exceeded." .
:stop1ErrorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain
:MotorJoint , :UniversalJoint ; dc:description "Describes the amount of error
correction to be performed in a time step when the joint reaches the limit on the
corresponding axis." .
:stop2Bounce a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ,
:UniversalJoint ; dc:description "Describes how much the joint should bounce the body
back on the corresponding axis if the joint limit has been reached or exceeded." .
:stop2ErrorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain
:MotorJoint , :UniversalJoint ; dc:description "Describes the amount of error
correction to be performed in a time step when the joint reaches the limit on the
corresponding axis." .
:stop3Bounce a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain :MotorJoint ;
dc:description "Describes how much the joint should bounce the body back on the
corresponding axis if the joint limit has been reached or exceeded." .
:stop3ErrorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float; rdfs:domain
:MotorJoint ; dc:description "Describes the amount of error correction to be performed
in a time step when the joint reaches the limit on the corresponding axis." .
:stopBounce a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SingleAxisHingeJoint , :SliderJoint ; dc:description "Describes how much the joint
should bounce the body back if the joint limit has been reached or exceeded." .
:stopBounce1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The stopBounce1 field is used to set how bouncy
the minimum and maximum angle stops are for axis 1. A value of zero means they are not
bouncy while a value of 1 means maximum bounciness (full reflection of force arriving
at the stop)." .
```

```
:stopConstantForceMix1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The stopConstantForceMix1 and suspensionForce
fields can be used to apply damping to the calculations by violating the normal
constraints by applying a small, constant force to those calculations." .
:stopErrorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:SingleAxisHingeJoint , :SliderJoint ; dc:description "Describes the amount of error
correction to be performed in a time step when the joint reaches the limit." .
:stopErrorCorrection1 a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The stopErrorCorrection1 and
suspensionErrorCorrection fields describe how quickly the system should resolve
intersection errors due to floating point inaccuracies." .
:stopTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :MovieTexture ,
:TimeSensor ; dc:description "Sets a stop time for the video/timer." .
:string a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Text ;
dc:description "The text strings are contained in the string field." .
:style a owl:DatatypeProperty ; rdfs:range [ a xsd:string ; owl:oneOf [ a rdf:List ;
rdf:first "PLAIN" ; rdf:rest [ a rdf:List ; rdf:first "BOLD" ; rdf:rest [ a rdf:List ;
rdf:first "ITALIC" ; rdf:rest [ a rdf:List ; rdf:first "BOLDITALIC" ; rdf:rest rdf:nil
] ] ] ] ] .
:subScale a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :DynamicLOD .
:subdivision a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Arc2D , :Cone
, :Dish , :DynamicLOD , :Nozzle , :Plane , :X3DPlanarGeometryNode , :RectangularTorus ,
:Torus , :SlopedCylinder , :Snout , :SolidOfRevolution , :Sphere ; dc:description
"Number of lines into which the arc is subdivided or number of segments the circle is
composed of or number of segments of a disc. For cones, subdivison specifies the number
of faces that are generated to approximate the sides of the cone. For dishes, it
specifies the number of faces that are generated to approximate the sides of the dish.
For nozzles, it specifies the number of faces that are generated to approximate the
sides of the pipe and nozzle." .
```

```
:summary a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :GeoMetadata
; dc:description "The summary string array contains a set of keyword/value pairs, with
each keyword and its subsequent value contained in a separate string; i.e., there
should always be an even (or zero) number of strings." .
:surfaceSpeed a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CollisionCollection ; dc:description "Defines the speed in the two friction directions
in speed base units." .
:surfaceTolerance a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:IsoSurfaceVolumeData ; dc:description "A threshold to adjust the boundary of the
isosurface." .
:surfaceValues a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:IsoSurfaceVolumeData ; dc:description "A list containing the surface values to be
extracted." .
:suspensionErrorCorrection a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The stopErrorCorrection1 and
suspensionErrorCorrection fields describe how quickly the system should resolve
intersection errors due to floating point inaccuracies." .
:suspensionForce a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:DoubleAxisHingeJoint ; dc:description "The stopConstantForceMix1 and suspensionForce
fields can be used to apply damping to the calculations by violating the normal
constraints by applying a small, constant force to those calculations." .
:tangent a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DBinaryContainerGeometryNode .
:tangentTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel that
contains the tangents in u." .
:tangentType a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DBinaryContainerGeometryNode ; dc:description "Specifies the byte format of the
tangents." .
```

```
:tau a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain :X3DDamperNode ;
dc:description "Specifies the time-constant of the internal filters and thus the speed
that the output of an X3DDamperNode responds to the input." .
:tessellationDetailCulling a owl:DatatypeProperty ; rdfs:range xsd:boolean ;
rdfs:domain :ElevationGrid ; dc:description "If TRUE, shape tesselation is lowered as
long as resulting error is lower than threshold." .
:tessellationErrorFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Factor of tesselation error for controlling speed-
performance trade-off." .
:tessellationErrorThreshold a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Shape tesselation is lowered as long as resulting
error is lower than threshold." .
:texCoord a owl:DatatypeProperty ; rdfs:range xsd:anyURI ; rdfs:domain
:X3DBinaryContainerGeometryNode , :X3DComposedGeometryNode , :ElevationGrid ;
dc:description "The URL to the binary file that contains the texture coordinates. For
ElevationGrid, it specifies per-vertex texture coordinates." .
:texCoordIndex a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:IndexedFaceSet ; dc:description "The index data for the texcoord data." .
:texCoordType a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:X3DBinaryContainerGeometryNode , :PopGeometry ; dc:description "Specifies the byte
format of the texture coordinates." .
:texcoordOffset a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Offset, given in bytes, for the texture coordinate
attribute inside the interleaved attribute array." .
:texcoordPrecision a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Precision, given in bytes, for the components of the
texture coordinate attribute." .
:textureCompression a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Specifies the preferred image compression method
to be used during rendering." .
```

```
:texturePriority a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:TextureProperties ; dc:description "Describes the texture residence priority for
allocating texture memory. Zero indicates the lowest priority and 1 indicates the
highest priority." .
:textureZ a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:ParticleSet ; dc:description "Contains z-values for the texture of a particle (used
with 3D textures)." .
:tightSize a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:PopGeometry ; dc:description "The size of the bounding box of this geometry, as it is
used for culling." .
:time a owl:DatatypeProperty ; rdfs:range xsd:time ; dc:description "Sends the absolute
time for a given tick of the TimeSensor." .
:tolerance a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :X3DDamperNode ;
dc:description "If tolerance is set to its default value -1, the browser implementation
is allowed to find a good way for detecting the end of a transition. Browsers that do
not have an elaborate algorithm can just use .001 as the tolerance value instead. If a
value larger than zero is specified for tolerance, the browser shall calculate the
difference between output and input for each internal filter being used and stop the
animation only when all filters fall below that limit or are equal to it. If zero is
specified for tolerance, a transition should be stopped only if input and output match
exactly for all internal filters." .
:topRadius a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Cone ;
dc:description "Specifies the radius of the cone at the apex." .
:topToBottom a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :FontStyle ;
dc:description "Specifies whether the text flows from top to bottom or from bottom to
top." .
:topUrl a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Background .
:torques a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :RigidBody ;
dc:description "The torques and forces fields define zero or more sets of torque and
```

```
force values that are applied to the object every frame. These are continuously applied
until reset to zero by the user." .
:transitionTime a owl:DatatypeProperty ; rdfs:range xsd:time ; rdfs:domain
:NavigationInfo ; dc:description "Specifies the duration of any viewpoint
 transition." .
:transitionType a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:NavigationInfo ; dc:description "Specifies the transition mode." .
:translation a owl:DatatypeProperty ; rdfs:domain :Transform , :GeoTransform ,
:HAnimHumanoid , :TextureTransform , :TextureTransform3D ; dc:description "Specifies a
translation to the coordinate system." .
:transmissionFactor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:CommonSurfaceShader ; dc:description "The value of transmissionTexture is multiplied
by this value. If no texture is set, the value is used directly." .
:transmissionTextureChannelMask a owl:DatatypeProperty ; rdfs:range xsd:string ;
rdfs:domain :CommonSurfaceShader ; dc:description "ChannelMask for transmissionTexture
in the form of a glsl swizzle (e.g., 'rgb', 'a')." .
:transmissionTextureCoordinatesId a owl:DatatypeProperty ; rdfs:range xsd:integer ;
rdfs:domain :CommonSurfaceShader ; dc:description "Texture coordinate channel to use
for transmissionTexture." .
:transmissionTextureId a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:CommonSurfaceShader ; dc:description "The texture unit." .
:transparency a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [
xsd:maxInclusive "1"^^xsd:float ] ) ] ; rdfs:domain :X3DBackgroundNode , :Material ,
:RadarVolumeStyle ; dc:description "Transparency of the background." .
:type a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :X3DField ,
:NavigationInfo , :ShaderPart , :OpacityMapVolumeStyle , :ProjectionVolumeStyle .
:typeParams a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:NavigationInfo ; dc:description "Specifies the view angle and height for helicopter
```

```
mode and min/max rotation angle for turntable in ]0, π[, starting from +y (0) down to -
y (π)." .
:ulimit a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :HAnimJoint .
:update a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:GeneratedCubeMapTexture , :RenderedTexture .
:url a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DEnvironmentTextureNode , :GeoMetadata , :Texture , :X3DTexture3DNode ,
:X3DGroupingNode , :Inline , :RenderedTexture , :RemoteSelectionGroup ; dc:description
"URL." .
:urlCenter a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :DynamicLOD .
:urlHead a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :DynamicLOD .
:urlIDMap a owl:DatatypeProperty ; rdfs:range xsd:anyURI ; rdfs:domain :MultiPart ;
dc:description "Specifies the url to the IDMap." .
:urlTail a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :DynamicLOD .
:useFiniteRotation a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:RigidBody ; dc:description "Influences the way the body's rotation is calculated." .
:useGeoCache a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:X3DBinaryContainerGeometryNode , :X3DPlanarGeometryNode , :Box ,
:X3DSpatialGeometryNode , :X3DSpatialGeometryNode , :X3DGeometryNode ,
:X3DComposedGeometryNode , :Nozzle , :PointSet , :Plane , :X3DPlanarGeometryNode ,
:Snout ; dc:description "Enable or disable geocache." .
:useGlobalGravity a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain
:RigidBody ; dc:description "Indicates whether this particular body should be
influenced by the containing RigidBodyCollection's gravity setting." .
:userDataFactor a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:ElevationGrid ; dc:description "Factor of user data for controlling speed-performance
trade-off." .
:value a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :ColorDamper ,
:CoordinateDamper , :X3DField , :FloatVertexAttribute , :MetadataBoolean ,
:OrientationChaser , :PositionChaser , :PositionDamper , :PositionDamper2D ,
```

```
:ScalarChaser , :ScalarDamper , :TexCoordDamper2D ; dc:description "The current color
value." .
:vector a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain :Normal ;
dc:description "Set of unit-length normal vectors, corresponding to indexed polygons or
vertices." .
:version a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain :HAnimHumanoid .
:vertexBufferSize a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:PopGeometry ; dc:description "Size of the vertex buffer, used to pre-allocate the
buffer before downloading data." .
:vertexCount a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:X3DBinaryContainerGeometryNode , :LineSet , :PopGeometry .
:vertices a owl:DatatypeProperty ; rdfs:range xsd:complexType ; xsd:domain
:TriangleSet2D ; dc:description "Specifies the triangles to be displayed." .
:visibilityLimit a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:NavigationInfo ; dc:description "Geometry beyond the visibilityLimit may not be
rendered (far culling plane of the view frustrum). visibilityLimit=0.0 indicates an
infinite visibility limit." .
:visibilityRange a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Fog ;
dc:description "Specifies the distance in length base units (in the local coordinate
system) at which objects are totally obscured by the fog." .
:warmColor a owl:DatatypeProperty ; rdfs:range xsd:complexType ; dc:description
"Specifies the color to be used for surfaces facing towards the light direction." .
:weight a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :HAnimDisplacer .
:weightConstant1 a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:BlendedVolumeStyle ; dc:description "Specifies a constant weight value to be use on
the parent volume data." .
:weightConstant2 a owl:DatatypeProperty ; rdfs:range xsd:string ; rdfs:domain
:BlendedVolumeStyle ; dc:description "A constant weight value to be use on the volume
data to be blended." .
```

```
:whichChoice a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain :Switch ;
dc:description "Specifies the index of the child to traverse, with the first child
having index 0." .
:xDimension a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:GeoElevationGrid ; dc:description "Defines the grid size in x." .
:xSectionOrientation a owl:DatatypeProperty ; rdfs:range xsd:complexType ; rdfs:domain
:RadarVolumeStyle ; dc:description "Defines the rotation for plane used for cutting
data." .
:xSectionPosition a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain
:RadarVolumeStyle ; dc:description "Defines the position of the cutting plane along an
axis passing through (0,0,0) and perpendicular to the plane." .
:xSpacing a owl:DatatypeProperty ; rdfs:range xsd:double ; rdfs:domain :ElevationGrid ,
:GeoElevationGrid ; dc:description "When the geoSystem is 'GD', xSpacing refers to the
number of units of longitude in angle base units between adjacent height values. When
the geoSystem is 'UTM', xSpacing refers to the number of eastings (length base units)
between adjacent height values." .
:xbottom a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [
xsd:maxInclusive "+INF"^^xsd:float ] ) ] ; rdfs:domain :Pyramid ; dc:description
"Defines the bottom length in x direction." .
:xbshear a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SlopedCylinder ;
dc:description "Defines the shear for the bottom in x direction." .
:xoff a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Pyramid , :Snout ;
dc:description "Defines the displacement along the x axis." .
:xtop a owl:DatatypeProperty ;rdfs:range [ a rdfs:Datatype ; owl:onDatatype xsd:float;
owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [ xsd:maxInclusive
"+INF"^^xsd:float ] ) ] ; rdfs:domain :Pyramid ; dc:description "Defines the top length
in x direction." .
:xtshear a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SlopedCylinder ;
dc:description "Defines the shear for the top in x direction." .
```

```
:yScale a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :GeoElevationGrid ;
dc:description "The yScale value can be used to produce a vertical exaggeration of the
data when it is displayed. If this value is set greater than 1.0, all heights will
appear larger than actual." .
:ybottom a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [
xsd:maxInclusive "+INF"^^xsd:float ] ) ] ; rdfs:domain :Pyramid ; dc:description
"Defines the bottom length in y direction." .
:ybshear a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SlopedCylinder ;
dc:description "Defines the shear for the bottom in y direction." .
:yoff a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :Pyramid , :Snout ;
dc:description "Defines the displacement along the y axis." .
:ytop a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype xsd:float;
owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [ xsd:maxInclusive
"+INF"^^xsd:float ] ) ] ; rdfs:domain :Pyramid ; dc:description "Defines the top length
in y direction." .
:ytshear a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :SlopedCylinder ;
dc:description "Defines the shear for the top in y direction." .
:zDimension a owl:DatatypeProperty ; rdfs:range xsd:integer ; rdfs:domain
:ElevationGrid , :GeoElevationGrid ; dc:description "Defines the grid size in z." .
:zFar a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :X3DLightNode ,
:OrthoViewpoint , :Viewpoint ; dc:description "Specifies the placement of the far plane
of the light projection." .
:zFarRange a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :DepthMode ;
dc:description "The far value for the depth range." .
:zNear a owl:DatatypeProperty ; rdfs:range xsd:float ; rdfs:domain :X3DLightNode ,
:OrthoViewpoint , :Viewpoint ; dc:description "Specifies the placement of the near
plane of the light projection." .
:zNearRange a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :DepthMode ;
dc:description "The near value for the depth range." .
```

```
:zSpacing a owl:DatatypeProperty ; rdfs:range xsd:double ; rdfs:domain :ElevationGrid ,
:GeoElevationGrid ; dc:description "When the geoSystem is 'GD', zSpacing refers to the
number of units of latitude in angle base units between vertical height values. When
the geoSystem is 'UTM', zSpacing refers to the number of northings (length base units)
between vertical height values." .
:x a owl:AllDisjointProperties ; owl:members (:zFar :zNear) .
:hatched rdfs:subPropertyOf :filled .

### Additional Roles ###

:baseForm a owl:ObjectProperty .
:createdBy a owl:ObjectProperty ; owl:inverseOf :creatorOf .
:createdIn a owl:ObjectProperty .
:creatorOf a owl:ObjectProperty ; owl:inverseOf :createdBy .
:designedBy a owl:ObjectProperty .
:designedFor a owl:ObjectProperty .
:formedFrom a owl:ObjectProperty .
:hasCompound a owl:ObjectProperty .

:animated a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:domain :3DModel .
:hasEdges a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger ; rdfs:domain
:3DObject .
:hasFaces a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger ; rdfs:domain
:3DObject .
:hasVertices a owl:DatatypeProperty ; rdfs:range xsd:nonNegativeInteger ; rdfs:domain
:3DObject .

###  3ds Max/Maya ###
:curvePoint a owl:ObjectProperty ; rdfs:domain :NURBSCurve .
:editPoint a owl:ObjectProperty ; rdfs:domain :NURBSCurve .
```

```
:span a owl:ObjectProperty ; rdfs:domain :NURBSCurve .
:controlVertex a owl:ObjectProperty ; rdfs:domain :NURBSCurve .
:hull a owl:ObjectProperty ; rdfs:domain :NURBSCurve .
:XYZCoordinates a owl:DatatypeProperty ; rdfs:range xsd:complexType .

### Reused Roles ###

foaf:currentProject a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
cc:permits a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
cc:prohibits a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
dc:rightsHolder rdfs:subPropertyOf owl:topObjectProperty .
foaf:surname a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain foaf:Person .
foaf:givenName a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain foaf:Person .

dc:created a owl:DatatypeProperty ; rdfs:range xsd:dateTime .
dc:modified a owl:DatatypeProperty ; rdfs:range xsd:dateTime .

### Individuals ###

:aqua a :Color .
:black a :Color .
:blue a :Color .
:fuchsia a :Color .
:gray a :Color .
:green a :Color .
:lime a :Color .
:maroon a :Color .
:navy a :Color .
```

```
:olive a :Color .
:purple a :Color .
:red a :Color .
:silver a :Color .
:teal a :Color .
:white a :Color .
:yellow a :Color .

:carbonFiber a :Material .
:fiberglass a :Material .
:nickel a :Material .
:aluminium a :Material .
:brass a :Material .
:bronze a :Material .
:cement a :Material .
:ceramic a :Material .
:concrete a :Material .
:copper a :Material .
:glass a :Material .
:gold a :Material .
:granite a :Material .
:iron a :Material .
:limestone a :Material .
:magnesium a :Material .
:manganese a :Material .
:marble a :Material .
:metal a :Material .
:paper a :Material .
:polycarbonate a :Material .
:polyvinylChloride a :Material .
```

```
:quartz a :Material .
:silver a :Material .
:soil a :Material .
:steel a :Material .
:textile a :Material .
:titanium a :Material .
:tungsten a :Material .
:wood a :Material .
:zinc a :Material .

:AC3D a :3DComputerGraphicsSoftware .
:Aladdin4D a :3DComputerGraphicsSoftware .
:Anim8or a :3DComputerGraphicsSoftware .
:AutoDesk3dsMax a :3DComputerGraphicsSoftware .
:AutoDeskMaya a :3DComputerGraphicsSoftware .
:AutodeskMotionBuilder a :3DComputerGraphicsSoftware .
:Blender a :3DComputerGraphicsSoftware .
:Carrara a :3DComputerGraphicsSoftware .
:Cinema4D a :3DComputerGraphicsSoftware .
:Claraio a :3DComputerGraphicsSoftware .
:DAZStudio a :3DComputerGraphicsSoftware .
:ElectricImageAnimationSystem a :3DComputerGraphicsSoftware .
:Houdini a :3DComputerGraphicsSoftware .
:iClone a :3DComputerGraphicsSoftware .
:K3D a :3DComputerGraphicsSoftware .
:LightWave3D a :3DComputerGraphicsSoftware .
:Messiah a :3DComputerGraphicsSoftware .
:MikuMikuDance a :3DComputerGraphicsSoftware .
:Modo a :3DComputerGraphicsSoftware .
:Moviestorm a :3DComputerGraphicsSoftware .
```

```
:Poser a :3DComputerGraphicsSoftware .
:Seamless3d a :3DComputerGraphicsSoftware .
:Shade3D a :3DComputerGraphicsSoftware .
:SketchUp a :3DComputerGraphicsSoftware .
:ZBrush a :3DComputerGraphicsSoftware .

:Pixar a :animationStudio .
:WaltDisneyAnimationStudios a :animationStudio .
:DreamWorksAnimation a :animationStudio .
:BlueSkyStudios a :animationStudio .
:IlluminationEntertainment a :animationStudio .
:SonyPicturesAnimation a :animationStudio .
:WarnerBrosAnimation a :animationStudio .

<http://3dontology.org/3dmodels/heart/> a :3dsMaxModel ; :createdIn :AutoDesk3dsMax ;
:baseForm :prolateSpheroid ; :hasFaces "177454"^^xsd:nonNegativeInteger ; :hasEdges
"532362"^^xsd:nonNegativeInteger ; :hasVertices "92026"^^xsd:nonNegativeInteger ;
:diffuseColor "0.745 0.090 0.090"^^xsd:complexType ; :specularColor "0.098 0.098
0.098"^^xsd:complexType .

:inf a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype xsd:float;
owl:withRestrictions ( [ xsd:minInclusive "-INF"^^xsd:float ] [ xsd:maxInclusive
"+INF"^^xsd:float ] ) ] .
:zeroone a owl:DatatypeProperty ; rdfs:range [ a rdfs:Datatype ; owl:onDatatype
xsd:float; owl:withRestrictions ( [ xsd:minInclusive "0"^^xsd:float ] [
xsd:maxInclusive "1"^^xsd:float ] ) ] .
```

A.3 The Video Ontology (VidOnt)

Type: core reference ontology

Standard alignment: ISO 15836-2009, ISO/IEC 15938, EBU CCDM 3351

Website: http://vidont.org

Namespace: `http://purl.org/ontology/video/`

Prefix: `vidont`

```
@prefix : <http://purl.org/ontology/video/> .
@prefix cc: <http://creativecommons.org/ns#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix ebuccdm: <http://www.ebu.ch/metadata/ontologies/ebuccdm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix owl2: <http://www.w3.org/2006/12/owl2#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix mpeg-7: <http://mpeg7.org/> .
@prefix po: <http://purl.org/ontology/po/> .
@prefix schema: <http://schema.org/> .
@prefix snomedct: <http://purl.bioontology.org/ontology/SNOMEDCT/> .
@prefix temporal: <http://swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl> .
@prefix movie: <http://data.linkedmdb.org/resource/movie/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
### Provenance ###

<http://purl.org/ontology/video/> a owl:Ontology ; foaf:maker dbpedia:Leslie_Sikos ;
dc:description "The Video Ontology (VidOnt) is a core reference ontology for video."@en
; dc:title "Video Ontology (VidOnt)"@en ; rdfs:seeAlso <http://www.vidont.org/> .

### Concepts ###

dc:Agent a owl:Class .
  schema:copyrightHolder a owl:Class ; rdfs:subClassOf dc:Agent .
    foaf:Organization a owl:Class ; rdfs:subClassOf schema:copyrightHolder .
      movie:film_company a owl:Class ; rdfs:subClassOf foaf:Organization .
      movie:film_distributor a owl:Class ; rdfs:subClassOf foaf:Organization .
      :FilmStudio a owl:Class ; rdfs:subClassOf foaf:Organization .
      schema:productionCompany a owl:Class ; rdfs:subClassOf foaf:Organization .
      po:Broadcaster a owl:Class ; dc:description "An organization responsible of some
broadcasting services. It can hold a set of services and outlets." ; rdfs:isDefinedBy
po: ; rdfs:label "Broadcaster" ; rdfs:subClassOf foaf:Organization .
      schema:TelevisionStation a owl:Class ; rdfs:subClassOf foaf:Organization .
    foaf:Person a owl:Class ; rdfs:subClassOf schema:copyrightHolder .
      schema:actor a owl:Class ; owl:equivalentClass [ a owl:Class ; owl:unionOf (
:protagonist :supportingActor ) ] ; rdfs:subClassOf foaf:Person ; dc:description "An
actor or actress, e.g. in a TV show, movie, video game, etc. Actors can be associated
with individual items or with a series, episode, or video clip."@en .
        :Protagonist a owl:Class ; rdfs:subClassOf schema:actor ; dc:description "Actor
or actress in the lead role." .
        :SupportingActor a owl:Class ; rdfs:subClassOf schema:actor ; dc:description
"Actor or actress supporting the actor/actress in the lead role." .
      :Choreographer a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:cinematographer a owl:Class ; rdfs:subClassOf foaf:Person .
```

```
      :Composer a owl:Class ; rdfs:subClassOf foaf:Person ; dc:description "Film score
       composer." .
      schema:director a owl:Class ; rdfs:subClassOf foaf:Person ; dc:description "A
       director of e.g. tv, radio, movie, video games etc. content. Directors can be
       associated with individual items or with a series, episode, clip." .
      :DirectorOfPhotography a owl:Class ; rdfs:subClassOf foaf:Person .
      schema:editor a owl:Class ; rdfs:subClassOf foaf:Person ; dc:description "A
       person who edited the Creative Work." .
      movie:film_art_director a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_casting_director a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_costume_designer a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_crewmember a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_critic a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_festival_sponsor a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_production_designer a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_set_designer a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_story_contributor a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:film_theorist a owl:Class ; rdfs:subClassOf foaf:Person .
      :Lector a owl:Class ; owl:equivalentClass :narrator ;
       rdfs:subClassOf foaf:Person .
      :MovieCharacter a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:music_contributor a owl:Class ; rdfs:subClassOf foaf:Person .
      schema:producer a owl:Class ; rdfs:subClassOf foaf:Person .
      :ScriptWriter a owl:Class ; rdfs:subClassOf foaf:Person .
      :Stuntman a owl:Class ; rdfs:subClassOf foaf:Person .
      movie:writer a owl:Class ; rdfs:subClassOf foaf:Person .
schema:award a owl:Class .
schema:BroadcastService a owl:Class .
```

```
  schema:BroadcastChannel a owl:Class ; owl:equivalentClass [ a owl:Class ; owl:unionOf
( :webcastChannel schema:TelevisionChannel ) ] ; rdfs:subClassOf
schema:BroadcastService .
    schema:TelevisionChannel a owl:Class ; rdfs:subClassOf schema:BroadcastChannel .
    :WebcastChannel a owl:Class ; rdfs:subClassOf schema:BroadcastChannel .
po:Channel a owl:Class ; dc:description "A physical channel on which a broadcast
occurs. A single outlet or service can be associated with multiple channels. For
example, Radio 4 LW broadcasts on Analogue Long Wave and on Digital Satellite." ;
rdfs:isDefinedBy po: .
schema:CreativeWork a owl:Class .
  schema:Review a owl:Class ; rdfs:subClassOf schema:CreativeWork ; dc:description "A
   review of an item, e.g., of a movie."@en .
  :Video a owl:Class ; rdfs:subClassOf schema:CreativeWork .
    :BroadcastVideo a owl:Class ; rdfs:subClassOf :Video .
    :Frame a owl:Class ; rdfs:subClassOf :Video .
      :B-Frame a owl:Class ; rdfs:subClassOf :Frame ; dc:description "Bi-predictive
       picture"@en .
      :Field a owl:Class ; rdfs:subClassOf :Frame ; dc:description "One of the many
       still images which are displayed sequentially to create the impression of motion
       on the screen. Two fields comprise a video frame."@en .
      :I-Frame a owl:Class ; rdfs:subClassOf :Frame ; dc:description "Intra-coded
        picture"@en .
      :Keyframe a owl:Class ; rdfs:subClassOf :Frame .
      :P-Frame a owl:Class ; rdfs:subClassOf :Frame ; dc:description "Predicted
       picture"@en .
      :Macroblock  a owl:Class ; rdfs:subClassOf :Frame ; dc:description "A frame
       segment (an individual prediction type)."@en .
      :Slice a owl:Class ; rdfs:subClassOf :Frame ; dc:description "A spatially
       distinct region of a frame that is encoded separately from any other region in
       the same frame."@en .
```

```
    :RegionOfInterest a owl:Class ; rdfs:subClassOf :Video .
    :TelecastVideo a owl:Class ; rdfs:subClassOf :BroadcastVideo .
    :WebcastVideo a owl:Class ; rdfs:subClassOf :BroadcastVideo .
  schema:Clip a owl:Class ; rdfs:subClassOf :Video ; owl:sameAs po:Clip .
    schema:MovieClip a owl:Class ; rdfs:subClassOf schema:Clip ; dc:description "A
     short TV or radio program or a segment/part of a program."@en .
    schema:TVClip a owl:Class ; rdfs:subClassOf schema:Clip ; dc:description "A short
     TV program or a segment/part of a TV program."@en .
  schema:MediaObject a owl:Class ; rdfs:subClassOf :Video ; dc:description "An image,
video, or audio object embedded in a web page. Note that a creative work may have many
media objects associated with it on the same web page. For example, a page about a
single song (MusicRecording) may have a music video (VideoObject), and a high and low
bandwidth audio stream (2 AudioObjects)."@en .
    schema:VideoObject a owl:Class ; rdfs:subClassOf schema:MediaObject ;
dc:description "A video file."@en .
      schema:MusicVideoObject a owl:Class ; rdfs:subClassOf schema:VideoObject ;
owl:disjointWith schema:TVSeason , schema:trailer ; dc:description "A music video
file."@en .
      schema:trailer a owl:Class ; rdfs:subClassOf schema:VideoObject ; dc:description
"The trailer of a movie or tv/radio series, season, episode, etc."@en .
  schema:Movie a owl:Class ; rdfs:subClassOf :Video ; owl:disjointWith
schema:MusicVideoObject , schema:TVSeason , schema:trailer .
    :AnimatedMovie a owl:Class ; rdfs:subClassOf schema:Movie ; dc:description
"Differentiates films from their releases. An animated movie can be a cartoon, a
cartoon with motion picture, a computer animaton, or live action." .
      :Cartoon a owl:Class ; rdfs:subClassOf :AnimatedMovie ; owl:disjointWith
:LiveAction .
      :LiveAction a owl:Class ; rdfs:subClassOf :AnimatedMovie ; owl:disjointWith
:Cartoon .
        :Action a owl:Class ; rdfs:subClassOf :LiveAction .
```

```
        :Adventure a owl:Class ; rdfs:subClassOf :LiveAction .
        :Comedy a owl:Class ; rdfs:subClassOf :LiveAction .
        :Documentary a owl:Class ; rdfs:subClassOf :LiveAction .
        :Drama a owl:Class ; rdfs:subClassOf :LiveAction .
        :FilmNoir a owl:Class ; rdfs:subClassOf :LiveAction .
        :Horror a owl:Class ; rdfs:subClassOf :LiveAction .
        :Romance a owl:Class ; rdfs:subClassOf :LiveAction .
        :SoapOpera a owl:Class ; rdfs:subClassOf :LiveAction .
        :Thriller a owl:Class ; rdfs:subClassOf :LiveAction .
        :Western a owl:Class ; rdfs:subClassOf :LiveAction .
      :CartoonWithMotionPicture a owl:Class ; rdfs:subClassOf :AnimatedMovie ;
owl:equivalentClass  [ a owl:Class ; owl:intersectionOf  ( :Cartoon :LiveAction ) ] ;
dc:description "A combination of cartoon and live action." .
      :ComputerAnimation a owl:Class ; rdfs:subClassOf :AnimatedMovie .
    :Release a owl:Class ; rdfs:subClassOf schema:Movie .
      :DiscRelease a owl:Class ; rdfs:subClassOf :Release .
        :Blu-Ray a owl:Class ; rdfs:subClassOf :DiscRelease ; dc:description "Blu-Ray
disc release." .
        :DVD-Video a owl:Class ; rdfs:subClassOf :DiscRelease .
        :HD-DVD a owl:Class ; rdfs:subClassOf :DiscRelease .
        :VCD a owl:Class ; rdfs:subClassOf :DiscRelease .
      :FileRelease a owl:Class ; rdfs:subClassOf :Release .
      :OnlineRelease a owl:Class ; owl:equivalentClass [ a owl:Class ; owl:complementOf
:DiscRelease ] .
        :GoogleVideo a owl:Class ; rdfs:subClassOf :OnlineRelease .
        :VimeoVideo a owl:Class ; rdfs:subClassOf :OnlineRelease .
        :YouTubeVideo a owl:Class ; rdfs:subClassOf :OnlineRelease .
      :VHS a owl:Class ; rdfs:subClassOf :Release .
  schema:TVSeason a owl:Class ; rdfs:subClassOf :Video ; owl:disjointWith schema:Movie
, schema:trailer .
```

```
    schema:TVSeries a owl:Class ; rdfs:subClassOf schema:TVSeason , [ a owl:Restriction
; owl:onProperty :hasEpisode ; owl:onClass schema:Episode ; owl:minQualifiedCardinality
"2"^^xsd:nonNegativeInteger ] ; rdfs:subclassOf schema:TVSeries , [rdf:type
owl:Restriction ; owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ; owl:onProperty
:partOf ; owl:onClass :TVSeries ] ; owl:disjointWith schema:Movie , schema:trailer .
      schema:Episode a owl:Class ; rdfs:subClassOf schema:TVSeries ; owl:disjointWith
schema:Movie , schema:trailer ; owl:sameAs po:Episode .
:EditingSoftware a owl:Class .
  :AudioEditingSoftware a owl:Class ; rdfs:subClassOf :EditingSoftware .
  :VideoEditingSoftware a owl:Class ; rdfs:subClassOf :EditingSoftware .
    :VideoConverter a owl:Class ; rdfs:subClassOf :VideoEditingSoftware .
    :3DComputerGraphicsSoftware a owl:Class ; rdfs:subClassOf :VideoEditingSoftware .
schema:Event a owl:Class .
  schema:BroadcastEvent a owl:Class ; rdfs:subClassOf schema:Event ; dc:description "An
over the air or online broadcast event."@en ; owl:sameAs po:Broadcast .
  movie:film_awards_ceremony a owl:Class ; rdfs:subClassOf schema:Event .
dc:LicenseDocument a owl:Class .
   cc:License a owl:Class ; dc:description "A set of requests/permissions to users of a
Work, e.g., a copyright license, the public domain, information for distributors" ;
rdfs:subClassOf dc:LicenseDocument .
   ebuccdm:IPRights a owl:Class ; dc:description "The intellectual property rights,
which transform EditorialObjects or MediaResources into Assets." .
ebuccdm:MediaResource a owl:Class ; dc:description "An audiovisual media resource (or
an image or document according to model extensions), which can be composed of one or
more fragments." .
dc:MediaType a owl:Class .
  :Disc a owl:Class ; rdfs:subClassOf dc:MediaType ; owl:disjointWith :Tape .
    :BDdisc a owl:Class ; rdfs:subClassOf :Disc ; owl:disjointWith :CDdisc , :DVDdisc ,
:Tape ; dc:description "Blu-Ray disc (medium)."@en .
    :CDdisc a owl:Class ; rdfs:subClassOf :Disc ; owl:disjointWith :DVDdisc , :Tape .
```

```
    :DVDdisc a owl:Class ; rdfs:subClassOf :Disc ; owl:disjointWith :Tape .
  :Tape a owl:Class ; rdfs:subClassOf dc:MediaType ; dc:description "Magnetic tape."@en
; owl:disjointWith :Disc .
ebuccdm:ProductionDevice a owl:Class .
  :RecordingEquipment a owl:Class ;rdfs:subClassOf ebuccdm:ProductionDevice .
    :VideoCamera a owl:Class ; rdfs:subclassOf :RecordingEquipment .
      :ConsumerVideoCamera a owl:Class ; rdfs:subClassOf :VideoCamera .
      :Camcorder a owl:Class ; rdfs:subClassOf :VideoCamera .
      :PTZCamera a owl:Class ; rdfs:subClassOf :VideoCamera .
      :Webcam a owl:Class ; rdfs:subClassOf :VideoCamera .
      :CameraPhone a owl:Class ; rdfs:subClassOf :VideoCamera .
      :ProfessionalVideoCamera a owl:Class ; rdfs:subClassOf :VideoCamera .
      :CinematographyCamera a owl:Class ; rdfs:subClassOf :VideoCamera .
    :DSLR a owl:Class ; rdfs:subClassOf :RecordingEquipment .
    :Microphone a owl:Class ; rdfs:subClassOf :RecordingEquipment .
:Scene a owl:Class ; rdfs:subClassOf :Video .
  :CinematicScene a owl:Class ; rdfs:subClassOf :Scene .
  :StagePlayScene a owl:Class ; rdfs:subClassOf :Scene .
  :MovieScene a owl:Class ; owl:equivalentClass :CinematicScene ; rdfs:subClassOf
:Scene .
    :ConversationalScene a owl:Class ; rdfs:subClassOf :MovieScene .
    :ActionScene a owl:Class ; rdfs:subClassOf :MovieScene .
      :CarChaseScene a owl:Class ; rdfs:subClassOf :ActionScene .
      :CrashScene a owl:Class ; rdfs:subClassOf :ActionScene .
      :ExplosionScene a owl:Class ; rdfs:subClassOf :ActionScene .
      :ShootingScene a owl:Class ; rdfs:subClassOf :ActionScene .
      :MurderScene a owl:Class ; rdfs:subClassOf :ActionScene .
      :FightingScene a owl:Class ; rdfs:subClassOf :ActionScene .
    :DreamingScene a owl:Class ; rdfs:subClassOf :MovieScene .
    :LoveScene a owl:Class ; rdfs:subClassOf :MovieScene .
```

```
    :NatureScene a owl:Class ; rdfs:subClassOf :MovieScene .
  :IndoorScene a owl:Class ; rdfs:subClassOf :Scene .
  :OutdoorScene a owl:Class ; rdfs:subClassOf :Scene .
dc:Standard a owl:Class .
  :AnnotationStandard a owl:Class ; owl:equivalentClass [ a owl:Class ; owl:oneOf (
:RDFa :Microdata :JSON-LD ) ] ; rdfs:subClassOf dc:Standard .
  :MetadataStandard a owl:Class ; rdfs:subClassOf dc:Standard .
  :BroadcastingStandard a owl:Class ; rdfs:subClassOf dc:Standard .
    po:DVB a owl:Class ; dc:description "Digital Video Broadcasting" .
  :StandardResolution a owl:Class ; rdfs:subClassOf dc:Standard .

### Roles ###

:basedOn a owl:ObjectProperty , owl:TransitiveProperty ; rdfs:subPropertyOf
owl:topObjectProperty .
  :cutOf a owl:ObjectProperty ; rdfs:subPropertyOf :basedOn .
  :installmentOf a owl:ObjectProperty ; rdfs:subPropertyOf :basedOn ;
owl:equivalentProperty :sequelOf .
  :recutOf a owl:ObjectProperty ; rdfs:subPropertyOf :basedOn .
  :remakeOf a owl:ObjectProperty ; rdfs:subPropertyOf :basedOn .
  :sequelOf a owl:ObjectProperty ; rdfs:subPropertyOf :basedOn .
:broadcastedBy a owl:ObjectProperty ; rdfs:comment "Telecast or webcast."@en ;
rdfs:subPropertyOf owl:topDataProperty .
  :Ad-ID a owl:ObjectProperty ; dc:description "The industry standard Ad-ID unique
identifier to identify commercial advertisers."@en ;
 rdfs:subPropertyOf :broadcastedBy .
  :broadcastFormat a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :callSign a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :colorZone a owl:ObjectProperty ; dc:description "CEA color zone to express TV signal
reception. Permissible values are blue, green, light green, pink, violet, or
```

```
yellow."@en ; rdfs:subPropertyOf :broadcastedBy ; rdfs:range [ a rdfs:Literal ;
owl:oneOf [ a rdf:List ; rdf:first "blue" ; rdf:rest [ a rdf:List ; rdf:first "green" ;
rdf:rest [ a rdf:List ; rdf:first "light green" ; rdf:rest [ a rdf:List ; rdf:first
"pink" ; rdf:rest [ a rdf:List ; rdf:first "red" ; rdf:rest [ a rdf:List ; rdf:first
"violet" ; rdf:rest [ a rdf:List ; rdf:first "yellow" ;
rdf:rest rdf:nil ] ] ] ] ] ] ] ] .
  :compassOrientation a owl:ObjectProperty ; dc:description "Indicates which direction
to aim directional outdoor antennas in for best signal reception. E.g., 310°."@en ;
rdfs:subPropertyOf :broadcastedBy .
  :deliveryFormat a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :FEC a owl:ObjectProperty ; dc:description "Forward Error Correction (FEC), also
known as channel coding. The technique used for controlling errors in data transmission
over unreliable or noisy communication channels. For example, 4/5."@en ;
rdfs:subPropertyOf :broadcastedBy ; rdfs:range [ a rdfs:Literal ; owl:oneOf [ a
rdf:List ; rdf:first "1/2" ; rdf:rest [ a rdf:List ; rdf:first "2/3" ; rdf:rest [ a
rdf:List ; rdf:first "3/4" ; rdf:rest [ a rdf:List ; rdf:first "3/5" ; rdf:rest [ a
rdf:List ; rdf:first "4/5" ; rdf:rest [ a rdf:List ; rdf:first "5/6" ; rdf:rest rdf:nil
] ] ] ] ] ] ] .
  :frequencyAssignment a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :modulation a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :transmissionStandard a owl:ObjectProperty ; dc:description "Transmission standard
used for broadcasting, e.g., DVB-T2" ; rdfs:subPropertyOf :broadcastedBy .
  :transmitPower a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :TVband a owl:ObjectProperty ; dc:description "TV band, such as UHF or VHF." ;
rdfs:subPropertyOf :broadcastedBy ; rdfs:range [ a rdfs:Literal ; owl:oneOf [ a
rdf:List ; rdf:first "UHF" ; rdf:rest [ a rdf:List ; rdf:first "VHF" ;
rdf:rest rdf:nil ] ] ] .
  :TVchannel a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
  :TVnetwork a owl:ObjectProperty ; rdfs:subPropertyOf :broadcastedBy .
```

```
  :TVstationType a owl:ObjectProperty ; dc:description "TV station type, e.g., DTV."@en
; rdfs:subPropertyOf :broadcastedBy .
:characterFrom a owl:ObjectProperty .
dc:conformsTo rdfs:subPropertyOf owl:topObjectProperty ; dc:description "Standard(s)
the media resource or hardware device conforms to."@en ; rdfs:subPropertyOf
owl:topObjectProperty .
:convertedFrom a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
foaf:currentProject a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:depicts a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:distributedBy a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:exportedTo a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:filmAdaptationOf a owl:ObjectProperty .
movie:film_featured_song a owl:ObjectProperty ; rdfs:subPropertyOf :topObjectProperty .
movie:film_festival_sponsor a owl:ObjectProperty ; rdfs:subPropertyOf
:topObjectProperty .
movie:film_screening_venue a owl:ObjectProperty ;
rdfs:subPropertyOf :topObjectProperty .
:filmmakerOf a owl:ObjectProperty ; rdfs:range foaf:Person ; rdfs:subPropertyOf
owl:topObjectProperty .
  :authorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ; owl:inverseOf
:authoredBy .
  :choreographerOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ;
owl:inverseOf :choreographedBy .
  :cinematographerOf a owl:ObjectProperty ; owl:inverseOf :cinematographyBy ;
owl:equivalentProperty :directorOfPhotographyOf ; rdfs:subPropertyOf :filmmakerOf .
  :composerOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :contributorOf a owl:ObjectProperty ; owl:inverseOf dc:contributor ;
rdfs:subPropertyOf :filmmakerOf .
  :costumeDesignerOf a owl:ObjectProperty ; owl:inverseOf :costumeDesignBy ;
rdfs:subPropertyOf :filmmakerOf .
```

```
  :creatorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :directorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ; owl:inverseOf
:directedBy .
  :directorOfPhotographyOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :editorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ;  owl:inverseOf
:editedBy .
  :makeupArtistOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ;
owl:inverseOf :makeupBy .
  :narratorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ; owl:inverseOf
:narratedBy .
  :playedIn a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :producerOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ; owl:inverseOf
:producedBy .
  :soundEditorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :soundMixerOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :starredIn a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
  :stuntmanOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ; owl:inverseOf
:stuntsBy .
  :visualEffectsDesignerOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf ;
owl:inverseOf :visualEffectsBy .
  :visualEffectsDirectorOf a owl:ObjectProperty ; rdfs:subPropertyOf :filmmakerOf .
:frameFrom a owl:ObjectProperty .
foaf:givenName a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain foaf:Person .
schema:genre a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:hasEpisode a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:hasSeason a owl:ObjectProperty ; owl:inverseOf :seasonOf ; owl:propertyChainAxiom (
:hasEpisode :hasSeries ) ; rdfs:subPropertyOf owl:topObjectProperty .
:hasSeries a owl:AsymmetricProperty , owl:ObjectProperty ; rdfs:subPropertyOf
owl:topObjectProperty .
```

```
temporal:before a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
temporal:after a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
temporal:hasStartTime a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
temporal:hasFinishTime a owl:ObjectProperty ;
 rdfs:subPropertyOf owl:topObjectProperty .
:importedFrom a owl:IrreflexiveProperty , owl:ObjectProperty ; rdfs:subPropertyOf
:topObjectProperty ; dc:description "The media format the video was imported
from."@en  .
:isActing a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
  :isAttackedBy a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isAttacking a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isFightingWith a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isHavingA a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
  :isHolding a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isKissing a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isLookingAt a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isPlayingThe a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isShootingAt a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isShouting a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isSpeaking a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isSwearing a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isTalkingTo a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isWhispering a owl:ObjectProperty ; rdfs:subPropertyOf :isActing .
  :isWithA a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
cc:license a owl:ObjectProperty ; rdfs:subPropertyOf :topObjectProperty .
:madeBy a owl:ObjectProperty , owl:TransitiveProperty ; rdfs:subPropertyOf
owl:topObjectProperty ; owl:inverseOf :filmmakerOf .
  :authoredBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:authorOf .
```

```
  :choreographedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:choreographerOf .
  :cinematographyBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:cinematographerOf .
  :commissionBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  dc:contributor rdfs:subPropertyOf :madeBy .
  :castMemberOf a owl:ObjectProperty ; rdfs:subPropertyOf dc:contributor .
  :costumeDesignBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  dc:creator rdfs:subPropertyOf :madeBy .
  :directedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:directorOf .
  :dubbedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  :editedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ;
   owl:inverseOf :editorOf .
    mpeg-7:ColorQuantization a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:EditedMovingRegion a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:Filter a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    :hasFileSystem a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    :MIMEType a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:MovingRegion a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:PointOfView a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:Scaling a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:SignLanguage a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:StillRegion a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:SubRegion a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:VideoSegment a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
    mpeg-7:VisualDefects a owl:ObjectProperty ; rdfs:subPropertyOf :editedBy .
  :ideaBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  :makeupBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  schema:musicBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
```

```
  :filmScoreBy a owl:ObjectProperty ; rdfs:subPropertyOf schema:musicBy .
  :muxedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
    :demuxedBy a owl:ObjectProperty ; rdfs:subPropertyOf :muxedBy .
    :remuxedBy a owl:ObjectProperty ; rdfs:subPropertyOf :muxedBy .
  :narratedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:narratorOf .
  :producedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:producerOf .
    :casting a owl:ObjectProperty ; rdfs:subPropertyOf :producedBy .
    mpeg-7:Shot a owl:ObjectProperty ; rdfs:subPropertyOf :producedBy .
      :scanning a owl:ObjectProperty ; rdfs:subPropertyOf mpeg-7:Shot ; rdfs:range [ a
rdfs:Datatype ; owl:oneOf [ a rdf:List ; rdf:first "interlaced" ; rdf:rest [ a rdf:List
; rdf:first "progressive" ; rdf:rest rdf:nil ] ] ] .
      :recordedWithVideoCamera a owl:ObjectProperty ; rdfs:subPropertyOf mpeg-7:Shot .
      :VideoFileFormat a owl:ObjectProperty ; rdfs:subPropertyOf mpeg-7:Shot .
      :VideoMode a owl:ObjectProperty ; rdfs:subPropertyOf mpeg-7:Shot .
      :VideoCodec a owl:ObjectProperty ; rdfs:subPropertyOf mpeg-7:Shot .
    :producedInCountry a owl:ObjectProperty ; rdfs:subPropertyOf :producedBy .
    :producedForTVSystem a owl:ObjectProperty ; rdfs:subPropertyOf :producedBy ;
dc:description "TV system, such as PAL, NTSC, or SECAM." ; rdfs:range [ a xsd:string ;
owl:oneOf [ a rdf:List ; rdf:first "NTSC" ; rdf:rest [ a rdf:List ; rdf:first "PAL" ;
rdf:rest [ a rdf:List ; rdf:first "SECAM" ; rdf:rest rdf:nil ] ] ] ] .
    :soundRecorded a owl:ObjectProperty ; rdfs:subPropertyOf :producedBy .
      :recordedWithMicrophone a owl:ObjectProperty ;
       rdfs:subPropertyOf :soundRecorded .
      :syncSound a owl:ObjectProperty ; rdfs:subPropertyOf :soundRecorded .
      :voiceOver a owl:ObjectProperty ; rdfs:subPropertyOf :soundRecorded .
      :AudioBitrateType a owl:ObjectProperty ; rdfs:subPropertyOf :soundRecorded .
      :AudioCodec a owl:ObjectProperty ; rdfs:subPropertyOf :soundRecorded .
      :AudioFileFormat a owl:ObjectProperty ; rdfs:subPropertyOf :soundRecorded .
```

```
      :AudioChannels a owl:ObjectProperty ; rdfs:subPropertyOf :soundRecorded .
    :screened  a owl:ObjectProperty ; rdfs:subPropertyOf :producedBy .
  :recordedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  :screenplayBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  :soundEditedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  :soundMixedBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy .
  :starring a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy  ; owl:inverseOf
:starredIn .
  :starringWith a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty ; a
owl:ReflexiveProperty .
  :storyBy a :ObjectProperty ;rdfs:subPropertyOf :madeBy .
  :stuntsBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:stuntmanOf .
  :visualEffectsBy a owl:ObjectProperty ; rdfs:subPropertyOf :madeBy ; owl:inverseOf
:visualEffectsDesignerOf .
:motionRelated a owl:ObjectProperty .
  :isApproachingTowards a owl:ObjectProperty ; rdfs:subPropertyOf :motionRelated .
  :isPassingThrough a owl:ObjectProperty ; rdfs:subPropertyOf :motionRelated .
  :isTransformingInto a owl:ObjectProperty ; rdfs:subPropertyOf :motionRelated .
:nominatedFor a :ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:partOf a owl:ObjectProperty , owl:TransitiveProperty ; rdfs:subPropertyOf
owl:topObjectProperty .
  :episodeOf a owl:ObjectProperty ; rdfs:subPropertyOf :partOf ; owl:inverseOf
:hasEpisode .
  :seasonOf a owl:ObjectProperty ; rdfs:subPropertyOf :partOf ; owl:inverseOf
:hasSeason .
  :seriesOf a owl:AsymmetricProperty , owl:ObjectProperty ;
   rdfs:subPropertyOf :partOf ; owl:inverseOf :hasSeries .
  :spatialSegmentOf a owl:ObjectProperty ; rdfs:subPropertyOf :partOf .
  :spatioTemporalSegmentOf a owl:ObjectProperty ; rdfs:subPropertyOf :partOf .
```

```
  :temporalSegmentOf a owl:ObjectProperty ; rdfs:subPropertyOf :partOf .
:portrayedBy a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:releaseStatus a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
  :releasedOn a owl:ObjectProperty ; rdfs:subPropertyOf :releaseStatus ; dc:description
"The media type(s) the video has been released on, e.g., disc or tape."@en ;
rdfs:subPropertyOf owl:topObjectProperty .
  :releasedInFormat a owl:ObjectProperty ; rdfs:subPropertyOf :releaseStatus ;
dc:description "The media format the video has been released in, e.g., DVD-Video or
Blu-Ray."@en .
  :promoted a owl:ObjectProperty ; rdfs:subPropertyOf :releaseStatus .
  :reception a owl:ObjectProperty ; rdfs:subPropertyOf :releaseStatus .
  :labelledAs a owl:ObjectProperty ; rdfs:subPropertyOf :releaseStatus ; dc:description
"Film category such as full length film, re-cut, remake, sequel, spin-off, short film,
and trailer."@en .
  :userRestriction a owl:ObjectProperty ; rdfs:subPropertyOf :releaseStatus .
cc:permits a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
cc:prohibits a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
:recordedFrom a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
dc:rightsHolder rdfs:subPropertyOf owl:topObjectProperty .
:sceneFrom a owl:ObjectProperty .
:supportedBy a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .
foaf:surname a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain foaf:Person .
:spatiallyRelated a owl:ObjectProperty .
  :distanceRelated a owl:ObjectProperty ; rdfs:subPropertyOf :spatiallyRelated .
    :isAt a owl:ObjectProperty ; rdfs:subPropertyOf :distanceRelated .
    :isFarFrom a owl:ObjectProperty ; rdfs:subPropertyOf :distanceRelated .
    :isInTheVicinityOf a owl:ObjectProperty ; rdfs:subPropertyOf :distanceRelated .
    :isNearTo a owl:ObjectProperty ; rdfs:subPropertyOf :distanceRelated .
    :isNearby a owl:ObjectProperty ; rdfs:subPropertyOf :distanceRelated .
```

```
  :directionallyRelated a owl:ObjectProperty .
    :internalDirectionalRelated a owl:ObjectProperty ; rdfs:subPropertyOf
:directionallyRelated  ; dc:description "An internal directional relation specifies
where an object is located inside the reference object." .
       :abaft a owl:ObjectProperty ; rdfs:subPropertyOf :internalDirectionalRelated .
       :athwart a owl:ObjectProperty ; rdfs:subPropertyOf :internalDirectionalRelated .
       :left a owl:ObjectProperty ; rdfs:subPropertyOf :internalDirectionalRelated .
       :right a owl:ObjectProperty ; rdfs:subPropertyOf :internalDirectionalRelated .
       :onTheBack a owl:ObjectProperty ;
        rdfs:subPropertyOf :internalDirectionalRelated .
    :externalDirectionalRelated a owl:ObjectProperty ; rdfs:subPropertyOf
:directionallyRelated ; dc:description "An external relations specifies where the
object is located outside of the reference objects." .
      :abeam a owl:ObjectProperty ; rdfs:subPropertyOf :externalDirectionalRelated .
      :astern a owl:ObjectProperty ; rdfs:subPropertyOf :externalDirectionalRelated .
      :isAbove a owl:ObjectProperty ; rdfs:subPropertyOf :externalDirectionalRelated .
      :isBehind a owl:ObjectProperty ; rdfs:subPropertyOf :externalDirectionalRelated .
      :isBelow a owl:ObjectProperty ; rdfs:subPropertyOf :externalDirectionalRelated .
      :isInFrontOf a owl:ObjectProperty ;
       rdfs:subPropertyOf :externalDirectionalRelated .
      :onTheLeftOf a owl:ObjectProperty ;
       rdfs:subPropertyOf :externalDirectionalRelated .
      :onTheRightOf a owl:ObjectProperty ;
       rdfs:subPropertyOf :externalDirectionalRelated .
  :sizeRelated a owl:ObjectProperty ; rdfs:subPropertyOf :spatiallyRelated .
    :isDifferentFrom a owl:ObjectProperty ; rdfs:subPropertyOf :sizeRelated .
    :isIdenticalTo a owl:ObjectProperty ; rdfs:subPropertyOf :sizeRelated .
    :isLargerThan a owl:ObjectProperty ; rdfs:subPropertyOf :sizeRelated .
    :isSmallerThan a owl:ObjectProperty ; rdfs:subPropertyOf :sizeRelated .
```

```
 :topologicallyRelated a owl:ObjectProperty ; rdfs:subPropertyOf :spatiallyRelated ;
dc:description "Topological relations including, but not limited to, the standard DE-
9IM topological relations and the RCC8 calculus." .
    :contains a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated  ;
dc:description "Corresponds to non-tangential proper part (NTPP)/non-tangential proper
part inverse (NTPPi) of RCC8." .
    :coveredBy a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated .
    :covers a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated .
    :disjointWith a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated  ;
dc:description "Corresponds to disjoint of DE-9IM and disconnected (DC) of RCC8." .
    :equals a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated  ;
dc:description "Corresponds to equals of DE-9IM and equal (EQ) of RCC8." .
    :intersects a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated  ;
dc:description "Corresponds to intersects of DE-9IM and partially overlapping (PO) of
RCC8." .
    :isIn a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated .
    :isOverlappedBy a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated .
    :isPartiallyOverlappedBy a owl:ObjectProperty ; rdfs:subPropertyOf
:topologicallyRelated .
    :touches a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated  ;
dc:description "Corresponds to touches of DE-9IM and externally connected (EC) of
RCC8." .
    :within a owl:ObjectProperty ; rdfs:subPropertyOf :topologicallyRelated .
    :tangentialProperPartInverseOf a owl:ObjectProperty ; rdfs:subPropertyOf
:topologicallyRelated  ; dc:description "Corresponds to tangential proper part inverse
(TPPi) of RCC8." .
    :tangentialProperPartOf a owl:ObjectProperty ; rdfs:subPropertyOf
:topologicallyRelated  ; dc:description "Corresponds to tangential proper part (TPP) of
RCC8." .
```

```
:starredWith a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty ; a
owl:ReflexiveProperty .
:temporallyRelated a owl:ObjectProperty  ; dc:description "Temporal relations
including, but not limited to, Allen’s temporal relations." .
  :after a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:before .
  :before a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:after .
  :contains a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :during .
  :during a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:contains .
  :earlier a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:later .
  :equals a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated .
  :finishedBy a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :finishes .
  :finishes a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :finishedBy .
  :follows a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:precedes .
  :later a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:earlier .
  :meets a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:metBy .
  :metBy a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:meets .
  :overlappedBy a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :overlaps .
```

```
  :overlaps a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :overlappedBy .
  :precedes a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :follows .
  :startedBy a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ;
owl:inverseOf :starts .
  :starts a owl:ObjectProperty ; rdfs:subPropertyOf :temporallyRelated ; owl:inverseOf
:startedBy .
:transmittedUsing a owl:ObjectProperty ; rdfs:subPropertyOf owl:topObjectProperty .

mpeg-7:BinCounts a owl:DatatypeProperty .
mpeg-7:CbACCoeff2 a owl:DatatypeProperty .
mpeg-7:CbDCCoeff a owl:DatatypeProperty .
mpeg-7:ColorTemperature a owl:DatatypeProperty ;
rdfs:range mpeg-7:ColorTemperatureType .
mpeg-7:CrACCoeff2 a owl:DatatypeProperty .
mpeg-7:CrDCCoeff a owl:DatatypeProperty .
mpeg-7:DominantColor a owl:DatatypeProperty ; rdfs:range mpeg-7:DominantColorType .
:hasAudioChannel a owl:DataProperty ; rdfs:range xsd:boolean .
  :AudioSamplingRate a owl:DataProperty ; rdfs:subPropertyOf :hasAudioChannel ;
dc:description "Audio sampling rate in kHz, e.g., 48." ; rdfs:range xsd:decimal .
  :AudioBitrate a owl:DataProperty ; rdfs:subPropertyOf :hasAudioChannel ; rdfs:range
xsd:positiveInteger .
:hasVideoChannel a owl:DataProperty ; rdfs:range xsd:boolean .
  :frameWidth a owl:DatatypeProperty ; dc:description "Video width in pixels, e.g.,
1920." ; rdfs:subPropertyOf :hasVideoChannel ; rdfs:range xsd:positiveInteger .
  :frameHeight a owl:DatatypeProperty ; dc:description "Video height in pixels, e.g.,
1080." ; rdfs:subPropertyOf :hasVideoChannel ; rdfs:range xsd:positiveInteger .
  :AspectRatio a owl:DatatypeProperty ; dc:description "The aspect ratio expresses how
many times longer is the width than the height of the video, e.g., 1.78 represents
```

```
1.78:1 (and also 16:9). Typical values are 1.33-2.76." ; rdfs:subPropertyOf
:hasVideoChannel ; rdfs:range xsd:decimal .
  :PAR a owl:DatatypeProperty ; dc:description "Pixel aspect ratio." ;
rdfs:subPropertyOf :hasVideoChannel .
  :VideoBitrate a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ;
rdfs:range xsd:positiveInteger .
  :FrameRate a owl:DatatypeProperty ; dc:description "Frame rate in frames per second
(fps). The most common values are 24, 25, 29.976, 30, 50, and 60." ; rdfs:range
xsd:decimal .
  :ColorTemperature a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ;
rdfs:range xsd:positiveInteger ; dc:description "Color temperature used for recording
in Kelvin, e.g., 5700." .
  :is3D a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ; rdfs:range
xsd:boolean .
  :Gamma a owl:DatatypeProperty ; rdfs:range xsd:decimal .
  :KneePoint a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ; rdfs:range
xsd:decimal ; dc:description "Professional video cameras allow the mapping of high
dynamic ranges into lower dynamic ranges, i.e., brighten the content without losing the
highlight details. The threshold of the knee is called the Knee Point." .
  :KneeSlope a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ;
dc:description "Professional video cameras allow the mapping of high dynamic ranges
into lower dynamic ranges, i.e., brighten the content without losing the highlight
details. The knee ratio is called the Knee Slope." .
  :RefreshRate a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ;
rdfs:range xsd:decimal ; dc:description "The vertical scan frequency expressed in Hz.
The most common values are 50 and 60." .
  :resolution a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel ; rdfs:range
[ a  rdfs:Datatype ; owl:onDatatype xsd:string ; owl:oneOf  ( "CGA" "CIF" "HVGA" "QVGA"
"SIF" "VGA" "PAL" "SVGA" "XGA" "XGA+" "SXGA" "SXGA+" "UXGA" "QXGA" "WQXGA" "QSXGA"
"WVGA" "WVGA" "NTSC" "WSVGA" "HD" "WXGA" "WXGA" "WSXGA+" "FHD" "2K" "WUXGA" "UQHD"
```

```
"WQHD" "UWQHD" "UHD-1" "QFHD" "4K" "8K" "16K" ) ] ; dc:description "The name of the
standard resolution of the video resource, e.g., QFHD" .
  :Subtitle a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel . # rdfs:range
xsd:language
  mpeg-7:MaxNumOfKeyFrames a owl:DatatypeProperty ;
   rdfs:subPropertyOf :hasVideoChannel .
  mpeg-7:cameraFollows a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel .
  mpeg-7:bitsPerBin a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel .
  mpeg-7:viewpoint a owl:DatatypeProperty ; rdfs:subPropertyOf :hasVideoChannel .
:completed a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  dc:created a owl:DatatypeProperty ; rdfs:subPropertyOf :completed ; rdfs:range
xsd:datetime .
  dc:modified a owl:DatatypeProperty ; rdfs:subPropertyOf :completed ; rdfs:range
xsd:datetime .
  :SNRatio a owl:DatatypeProperty ; rdfs:subPropertyOf :completed ; dc:description
"Signal-Noise Ratio" ; rdfs:range xsd:decimal .
  :plot a :DatatypeProperty ; rdfs:subPropertyOf :completed ; rdfs:range xsd:string .
:isArmed a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isAlive a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isDead a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isHappy a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isAngry a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isSurprised a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isDisappointed a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isPleased a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isSatisfied a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isHurt a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isInPain a owl:DatatypeProperty ; rdfs:range xsd:boolean ; rdfs:subPropertyOf
:isHurt .
:isShocked a owl:DatatypeProperty ; rdfs:range xsd:boolean .
```

```
:isOpening a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isClosing a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isExploding a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isMoving a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isAppearing a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isDisappearing a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isGrowing a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isMovingDownwards a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isMovingLeft a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isMovingRight a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isMovingUpwards a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isPassing a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isShrinking a owl:DatatypeProperty ; rdfs:range xsd:boolean .
  :isTransitioning a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isSpeedingUp a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isSlowingDown a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:hasSubtitle a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isDriving a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isEating a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:isSleeping a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:startedAt a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:finishedAt a owl:DatatypeProperty ; rdfs:range xsd:boolean .
:released a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:boolean .
    :Angles a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ; rdfs:range
xsd:positiveInteger .
    :Chapters a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ; rdfs:range
xsd:positiveInteger .
    :hasBDRelease a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
```

```
    :hasChapterMenu a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
    :hasDVDRelease a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
    :hasHDDVDRelease a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
    :hasRootMenu a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
    :hasSVCDRelease a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
    :hasSeamlessBranching a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean ; dc:description "A mechanism used on DVDs and Blu-ray Discs to
allow the player to jump to a different scene after finishing one." .
    :hasVCDRelease a owl:DatatypeProperty ; rdfs:subPropertyOf :hasDiscRelease ;
rdfs:range xsd:boolean .
  :FanWebsite a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:anyURI .
  :OfficialWebsite a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:anyURI .
  :boxOffice a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:positiveInteger .
  :budget a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:positiveInteger .
  :grossRevenue a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:positiveInteger ; dc:description "Gross revenue in USD."@en .
  :hasDiscRelease a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:boolean .
  :liveDate a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:dateTime .
```

```
  :premiereDate a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:dateTime .
  :rating a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:nonNegativeInteger .
  :releaseDate a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:dateTime .
  :runningTime a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:positiveInteger ; dc:description "The running time in minutes" .
  dc:available a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:dateTime .
  dc:dateCopyrighted a owl:DatatypeProperty ; rdfs:subPropertyOf :released ; rdfs:range
xsd:dateTime .
  movie:film_regional_release_date a owl:DataProperty ; rdfs:subPropertyOf :released ;
rdfs:range xsd:dateTime .
  mpeg-7:MinimumAge a owl:DatatypeProperty ; rdfs:subPropertyOf :released .
  mpeg-7:ParentalGuidance a owl:DatatypeProperty ; rdfs:subPropertyOf :released ;
rdfs:range xsd:string ; dc:description "Parental guidance categories, such as sex and
nudity, violence and gore, profanity, alcohol/drugs/smoking, or frightening/intense
scenes." .
  mpeg-7:ParentalRating a owl:DatatypeProperty ; rdfs:subPropertyOf :released ;
rdfs:range xsd:string ; dc:description "Parental rating, e.g., G - general audiences
(all ages permitted), PG - parental guidance suggested (some material may not be
suitable for children), PG-13 - parents strongly cautioned (some material may be
inappropriate for children under 13), R - restricted (under 17 requires accompanying
parent or adult guardian), NC-17 - no one 17 and undef admitted" .
  schema:copyrightYear a owl:DatatypeProperty ; rdfs:subPropertyOf :released ;
dc:description "The year during which the claimed copyright for the CreativeWork was
first asserted."@en . # ; rdfs:range xsd:date .
:isSeries a owl:DatatypeProperty ; rdfs:range xsd:boolean .
```

```
  :hasEpisodes a owl:DatatypeProperty ; rdfs:subPropertyOf :isSeries ; rdfs:range
xsd:positiveInteger .
  :hasSeasons a owl:DatatypeProperty ; rdfs:subPropertyOf :isSeries ; rdfs:range
xsd:positiveInteger .
mpeg-7:YACCoeff5 a owl:DatatypeProperty .
mpeg-7:YDCCoeff a owl:DatatypeProperty .

### Individuals ###


dbpedia:Warner_Bros a :filmStudio .
dbpedia:Walt_Disney_Pictures a :filmStudio .
dbpedia:Universal_Studios a :filmStudio .
dbpedia:Columbia_Pictures a :filmStudio .
dbpedia:20th_Century_Fox a :filmStudio .
dbpedia:Paramount_Pictures a :filmStudio .
dbpedia:Lionsgate_Films a :filmStudio .
dbpedia:The_Weinstein_Company a :filmStudio .
dbpedia:Relativity_Media a :filmStudio .
dbpedia:Open_Road_Films a :filmStudio .
dbpedia:CBS_Films a :filmStudio .
dbpedia:DreamWorks a :filmStudio .
dbpedia:DreamWorks_Animation a :filmStudio .
dbpedia:Gaumont_Film_Company a :filmStudio .
dbpedia:Metro-Goldwyn-Mayer a :filmStudio .

dbpedia:Australian_Broadcasting_Corporation a po:Broadcaster .
dbpedia:Seven_Network a po:Broadcaster .

dbpedia:Adobe_Premiere_Elements a :VideoEditingSoftware .
dbpedia:Adobe_Premiere_Pro a :VideoEditingSoftware .
```

```
dbpedia:Avid_Free_DV a :VideoEditingSoftware .
dbpedia:Autodesk_Smoke a :VideoEditingSoftware .
dbpedia:Xpress_Pro a :VideoEditingSoftware .
dbpedia:Media_Composer a :VideoEditingSoftware .
dbpedia:Avidemux a :VideoEditingSoftware .
dbpedia:AVS_Video_Editor a :VideoEditingSoftware .
dbpedia:Cinelerra a :VideoEditingSoftware .
dbpedia:Corel_VideoStudio a :VideoEditingSoftware .
dbpedia:Montage_Extreme a :VideoEditingSoftware .
dbpedia:EditDV a :VideoEditingSoftware .
dbpedia:Edius a :VideoEditingSoftware .
dbpedia:Final_Cut_Express a :VideoEditingSoftware .
dbpedia:Final_Cut_Pro a :VideoEditingSoftware .
dbpedia:FORscene a :VideoEditingSoftware .
dbpedia:IMovie a :VideoEditingSoftware .
dbpedia:IvsEdits a :VideoEditingSoftware .
dbpedia:Kdenlive a :VideoEditingSoftware .
dbpedia:Lightworks a :VideoEditingSoftware .
dbpedia:LiVES a :VideoEditingSoftware .
dbpedia:Magix_Movie_Edit_Pro a :VideoEditingSoftware .
dbpedia:MainActor a :VideoEditingSoftware .
dbpedia:MoviePlus a :VideoEditingSoftware .
dbpedia:MPEG_Video_Wizard_DVD a :VideoEditingSoftware .
dbpedia:VideoPad a :VideoEditingSoftware .
dbpedia:Nero_Multimedia_Suite a :VideoEditingSoftware .
dbpedia:OpenShot_Video_Editor a :VideoEditingSoftware .
dbpedia:Pinnacle_Studio a :VideoEditingSoftware .
dbpedia:PiTiVi a :VideoEditingSoftware .
dbpedia:Sony_Vegas_Pro a :VideoEditingSoftware .
dbpedia:Sony_Vegas_Movie_Studio a :VideoEditingSoftware .
```

```
dbpedia:Ulead_MediaStudio_Pro a :VideoEditingSoftware .
dbpedia:VideoThang a :VideoEditingSoftware .
dbpedia:VirtualDub a :VideoEditingSoftware .
dbpedia:VirtualDubMod a :VideoEditingSoftware .
dbpedia:Windows_Movie_Maker a :VideoEditingSoftware .

dbpedia:Autodesk_3ds_Max a :3DComputerGraphicsSoftware .
dbpedia:Autodesk_Maya a :3DComputerGraphicsSoftware .
dbpedia:Cinema4D a :3DComputerGraphicsSoftware .
dbpedia:E-on_Vue a :3DComputerGraphicsSoftware .
dbpedia:Poser a :3DComputerGraphicsSoftware .

dbpedia:Canon_EOS-1D_C a :CinematographyCamera .
dbpedia:Canon_EOS_C100 a :CinematographyCamera .
dbpedia:Canon_EOS_C300 a :CinematographyCamera .
dbpedia:Canon_EOS_C500 a :CinematographyCamera .

:CanonXA10 a :ProfessionalVideoCamera .
:CanonXA20 a :ProfessionalVideoCamera .
:CanonXA25 a :ProfessionalVideoCamera .
:CanonXF305 a :ProfessionalVideoCamera .

:hMedia a :annotationStandard .
:JSON-LD a :annotationStandard .
:Microdata a :annotationStandard .
:RDFa a :annotationStandard .

<https://www.youtube.com/watch?v=hZibbnjiGuY#t=0:00:48,0:00:50&xywh=721,207,358,323> a
mpeg-7:StillRegion ; :spatioTemporalSegmentOf
```

```
<https://www.youtube.com/watch?v=hZibbnjiGuY> ; :depicts snomedct:15013002 ,
dbpedia:Floater .

### Rules ###

:FHDVideo a owl:Class ; owl:equivalentClass :1080pVideo ; a [ a owl:Class ; owl:unionOf
( [ a owl:Class ; owl:complementOf [ a owl:Restriction ; owl:hasValue
"1080p"^^xsd:string ; owl:onClass :Video ; owl:onProperty :videoMode ] ] [ a
owl:Restriction ; owl:maxQualifiedCardinality "1080"^^xsd:nonNegativeInteger ;
owl:onProperty :frameHeight ] ) ] .

:ProgressiveHDVideo a [ a owl:Class ; owl:unionOf ( [ a owl:Class ; owl:complementOf [
a owl:Restriction ; owl:hasValue "720p"^^xsd:string ; owl:onClass :Video ;
owl:onProperty :videoMode ] ] [ a owl:Restriction ; owl:hasValue
"progressive"^^xsd:string ; owl:onProperty :scanning ] ) ] .

:InterlacedFHDVideo a [ a owl:Class ; owl:unionOf ( [ a owl:Class ; owl:complementOf [
a owl:Restriction ; owl:hasValue "1080i"^^xsd:string ; owl:onClass :Video ;
owl:onProperty :videoMode ] ] [ a owl:Restriction ; owl:hasValue
"interlaced"^^xsd:string ; owl:onProperty :scanning ] ) ] .

:ProgressiveFHDVideo a [ a owl:Class ; owl:unionOf ( [ a owl:Class ; owl:complementOf [
a owl:Restriction ; owl:hasValue "1080p"^^xsd:string ; owl:onClass :Video ;
owl:onProperty :videoMode ] ] [ a owl:Restriction ; owl:hasValue
"progressive"^^xsd:string ; owl:onProperty :scanning ] ) ] .

:BDVideoModes a owl2:DataRange ; owl:oneOf ("720p"^^xsd:string "1080i"^^xsd:string
"1080p"^^xsd:string) .
:BluRayVideo a owl:Restriction ; owl:onProperty :videoMode ; owl:someValuesFrom
:BDVideoModes .
```

```
:4KVideo a [ a owl:Class ; owl:unionOf ( [ a owl:Class ; owl:complementOf [ a
owl:Restriction ; owl:hasValue "2160p"^^xsd:string ; owl:onClass :Video ;
owl:onProperty :videoMode ] ] [ a owl:Restriction ; owl:maxQualifiedCardinality
"2160"^^xsd:nonNegativeInteger ; owl:onProperty :frameHeight ] ) ] .

:Progressive4KVideo a [ a owl:Class ; owl:unionOf ( [ a owl:Class ; owl:complementOf [
a owl:Restriction ; owl:hasValue "2160p"^^xsd:string ; owl:onClass :Video ;
owl:onProperty :videoMode ] ] [ a owl:Restriction ; owl:hasValue
"progressive"^^xsd:string ; owl:onProperty :scanning ] ) ] .

:4KBluRay a owl:Class ; :videoMode "2160p"^^xsd:string .

:UltraHDBluray a owl:Class ; owl:equivalentClass :4KBluRay .
```

References

1 National Information Standards Organization (NISO) (2001) Understanding metadata. http://www.niso.org/publications/press/UnderstandingMetadata.pdf. Accessed 29 Sep 2017

2 Wu, Z., Yao, T., Fu, Y., Jiang, Y.-G. (2016) Deep learning for video classification and captioning. ArXiv: abs/1609.06782

3 Boll, S., Klas, W., Sheth, A. (1998) Overview on using metadata to manage multimedia data. In: Sheth, A., Klas, W. (eds.) Multimedia data management: using metadata to integrate and apply digital media, p. 3. New York: McGraw-Hill

4 Lowe, D. G. (1999) Object recognition from local scale-invariant features. In: 7th IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. Corfu, Greece, 20–25 September 1999. Washington, DC: IEEE Computer Society. doi: 10.1109/ICCV.1999.790410

5 Dollár, P., Rabaud, V., Cottrell, G., Belongie, S. (2005) Behavior recognition via sparse spatio-temporal features. In: 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72. Beijing, China, 15–16 October 2005. Washington, DC: IEEE Computer Society. doi: 10.1109/VSPETS.2005.1570899

6 Dalal, N., Triggs, B. (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893. San Diego, 20–26 June 2005. Washington, DC: IEEE Computer Society. doi: 10.1109/CVPR.2005.177

7 Bay, H., Ess, A., Tuytelaars, T., Van Gool, L. (2008) Speeded-up robust features (SURF). Comput. Vis. Image Und. 110(3):346–359. doi: 10.1016/j.cviu.2007.09.014

8 Chang, C.-C., Lin, C.-J. (2011) LIBSVM: a library for support vector machines. ACM T. Intell. Syst. Technol. 2(3), Article 27. doi: 10.1145/1961189.1961199

9 Merler, M., Huang, B., Xie, L., Hua, G., Natsev, A. (2012) Semantic model vectors for complex video event recognition. IEEE Trans. Multimed. 14(1):88–101. doi: 10.1109/TMM.2011.2168948

10 Luo, J., Savakis, A. E., Singhal, A. (2005) A Bayesian network-based framework for semantic image understanding. Pattern Recogn. 38(6):919–934. doi: 10.1016/j.patcog.2004.11.001

11 Papadopoulos, G. Th., Mezaris, V., Kompatsiaris, I., Strintzis, M. G. (2007) Combining global and local information for knowledge-assisted image analysis and classification. EURASIP J. Adv. Signal Process. doi: 10.1155/2007/45842

12 Janvier, B., Bruno, E., Marchand-Maillet, S., Pun, T. (2005) A contextual model for semantic video structuring. In: 13th European Signal Processing Conference, Antalya, Turkey, 4–8 September 2005

13 Dai, S., Zhan, Y., Mao, Q., Zhang, S. (2013) A video semantic analysis method based on kernel discriminative sparse representation and weighted KNN. In: IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. doi: 10.1109/GreenCom-iThings-CPSCom.2013.154

14 Elguebaly, T., Bouguila, N. (2015) Semantic scene classification with generalized Gaussian mixture models. In: Kamel, M., Campilho, A. (eds.) Image analysis and recognition. ICIAR 2015. Lect. Notes Comput. Sci., vol. 9164. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-20801-5_17

15 Inoue, N., Shinoda, K. (2013) q-Gaussian mixture models for image and video semantic indexing. J. Vis. Commun. Image R. 24(8), 1450–1457. doi: 10.1016/j.jvcir.2013.10.005

16 Hamed, A.A.M., Li, R., Xiaoming, Z., Xu, C. (2013) Video genre classification using weighted kernel logistic regression. Adv. Multimed., Article ID 653687. doi: 10.1155/2013/653687

17 Wang, Z., Liu, G., Ma, Y., Qian, X., Yang, Y. (2010) An improved fusion method based on Ada-boost algorithm for semantic concept extraction. In: Second International Conference on Internet Multimedia Computing and Service, pp. 19–22. doi: 10.1145/1937728.1937733

18 Caicedo, J. C., Gonzalez, F. A., Romero, E. (2008) Content-based medical image retrieval using low-level visual features and modality identification. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D. W., Peñas, A., Petras, V., Santos, D. (eds.) Advances in multilingual and multimodal information retrieval. CLEF 2007. Lect. Notes Comput. Sci., vol. 5152, pp. 615–622. Heidelberg, Germany: Springer. doi: 10.1007/978-3-540-85760-0_78

19 Barb, A., Shyu, C.-R. (2003) Semantics modeling in diagnostic medical image databases using customized fuzzy membership functions. 12th IEEE International Conference on Fuzzy Systems, St Louis, MO, USA, 25–28 May 2003. New York: IEEE. doi: 10.1109/FUZZ.2003.1206595

20 Gibaud, B. (2011) The quest for standards in medical imaging. Eur. J. Radiol. 78(2):190–198. doi: 10.1016/j.ejrad.2010.05.003

21 Yu, W.D., Jonnalagadda, S. R. (2006) Semantic web and mining in healthcare. 8th International Conference on e-Health Networking, Applications and Services. New Delhi, India, 17–19 August 2006. doi: 10.1109/HEALTH.2006.246449

22 Chen, H., Chen, X., Gu, P., Wu, Z., Yu, T. (2014) OWL reasoning framework over big biological knowledge network. BioMed Research International 2014, Article ID 272915. doi: 10.1155/2014/272915

23 Panahiazar, M., Taslimitehrani, V., Jadhav, A., Pathak, J. (2014) Empowering personalized medicine with big data and Semantic Web technology: promises, challenges, and use cases. In: 2014 IEEE International Conference on Big Data, pp. 790–795. Washington, DC, USA, 27–30 October 2014. doi: 10.1109/BigData.2014.7004307

24 Zhu, X., Fan, J., Elmagarmid, A. K., Wu, X. (2003) Hierarchical video content description and summarization using unified semantic and visual similarity. Multimedia Syst. 9(1):31–53. doi: 10.1007/s00530-003-0076-5

25 Batet, M., Sánchez, D., Valls, A. (2011) An ontology-based measure to compute semantic similarity in biomedicine. J. Biomed. Inform. 44(1):118-125. doi: 10.1016/j.jbi.2010.09.002

26 Lux, M., Riegler, M. (2013) Annotation of endoscopic videos on mobile devices: a bottom-up approach. In: 4th ACM Multimedia Systems Conference, pp. 141–145. Oslo, Norway, 27 February–1 March 2013. doi: 10.1145/2483977. 2483996

27 Panayides, A., Eleftheriou, I., Pantziaris, M. (2013) Open-source telemedicine platform for wireless medical video communication. Int. J. Telemed. Appl. 2013, Article ID 457491. doi: 10.1155/2013/457491

28 Martini, M. G., Hewage, C. T. E. R. (2010) Flexible macroblock ordering for context-aware ultrasound video transmission over mobile WiMAX. Int. J. Telemed. Appl. 2010, Article ID 127519. doi: 10.1155/2010/127519

29 Davis, S., Burnett, I., Ritz, C. (2009) Using social networking and collections to enable video semantics acquisition. IEEE MultiMedia PP(99). doi: 10.1109/MMUL.2009.72

30 Gómez-Romero, J., García, J., Patricio, M. A., Serrano, M. A., Molina, J. M. (2016) Context-based situation recognition in computer vision systems. In: Gómez-Romero, J., García, J., Patricio M. A., Serrano, M. A., Molina, J. M. (eds.) Context-enhanced information fusion. Advances in computer vision and pattern recognition, pp. 627–651. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-28971-7_23

31 Sikos, L. F. (2015) Structured data. In: Mastering structured data on the Semantic Web: from HTML5 Microdata to Linked Open Data, pp. 2–5. New York: Apress. doi: 10.1007/978-1-4842-1049-9_1

32 Brickley, D., Guha, R. V. (2014) RDF Schema 1.1. World Wide Web Consortium. https://www.w3.org/TR/rdf-schema/. Accessed 29 September 2017

33 Herman, I. (2010) Why OWL and not WOL? In: Tutorial on Semantic Web technologies. World Wide Web Consortium. https://www.w3.org/People/Ivan/CorePresentations/RDFTutorial/Slides.html#%28114%29. Accessed 29 September 2017

34 Dean, M., Schreiber, G. (eds.), Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., Stein, L. A. (2004) OWL Web Ontology Language

reference. World Wide Web Consortium. https://www.w3.org/TR/owl-ref/. Accessed 29 September 2017

35 Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., Rudolph, S. (eds.) (2012) OWL 2 Web Ontology Language—Primer 2nd ed. World Wide Web Consortium. https://www.w3.org/TR/owl-primer/. Accessed 29 September 2017

36 Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.), Calvanese, D., Carroll, J., De Giacomo, G., Hendler, J., Herman I., Parsia, B., Patel-Schneider, P. F., Ruttenberg, A., Sattler, U., Schneider, M. (2012) OWL 2 Web Ontology Language—profiles. World Wide Web Consortium. https://www.w3.org/TR/owl2-profiles/. Accessed 29 September 2017

37 Krötzsch, M., Simančík, F., Horrocks, I. (2013) A description logic primer. arXiv: abs/1201.4089v3

38 Bertini, M., Del Bimbo, A., Serra G. (2008) Learning rules for semantic video event annotation. In: Sebillo, M., Vitiello, G., Schaefer, G. (eds.) Visual Information Systems. Web-Based Visual Information Search and Management. VISUAL 2008. Lect. Notes Comput. Sci., vol. 5188, pp. 192–203. Heidelberg: Springer. doi: 10.1007/978-3-540-85891-1_22

39 Pantoja, C., Ciapetti, A., Massari, C. (2015) Action recognition in surveillance videos using semantic web rules. In: 6th International Conference on Imaging for Crime Prevention and Detection, London, UK, 15–17 July 2015. doi: 10.1049/ic.2015.0103

40 Tani, K. M. Y., Lablack, A., Ghomari, A., Bilasco, I. M. (2015) Events detection using a video-surveillance ontology and a rule-based approach. In: Agapito, L., Bronstein, M., Rother, C. (eds.) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lect. Notes Comput. Sci., vol. 8926, pp. 299–308. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-16181-5_21

41 Tani, M. Y. K., Lablack, A., Ghomari, A., Bilasco, I. M. (2014) Events detection using a video surveillance ontology and a rule-based approach. In: Agapito, L., Bronstein M. M., Rother, C. (eds.) Computer Vision – ECCV 2014 workshops. Lect. Notes Comput. Sci., vol. 8926, pp. 299–308. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-16181-5_21

42 Haarslev, V. (1999) A logic-based formalism for reasoning about visual representations. J. Visual Lang. Comput. 4(10):421–445. doi: 10.1006/jvlc.1999.0133

43 Kaplunova, A., Haarslev, V., Möller, R. (2002) Adding ternary complex roles to $\mathcal{ALC}_{RP}{}^{(\mathcal{D})}$. In: International Workshop on Description Logics, Toulouse, France, 19–21 April 2002

44 Na, K.-S., Kong, H., Cho, M. (2006) Multimedia information retrieval based on spatiotemporal relationships using description logics for the Semantic Web. Int. J. Intell. Syst. 21(7):679–692. doi: 10.1002/int.20153

45 Wessel, M. (2003) Qualitative spatial reasoning with the $\mathcal{ALCI}_{RCC}$ family—first results and unanswered questions. https://kogs-www.informatik.uni-hamburg.de/publikationen/pub-wessel/report7.pdf. Accessed 29 September 2017

46 Randell, D. A., Cui, Z., Cohn, A. G. (1992) A spatial logic based on regions and connections. In: Nebel, B., Rich, C., Swartout, W. (eds.) In: 3rd International Conference on Knowledge Representation and Reasoning, pp. 165–176. Morgan Kaufmann

47 Egenhofer, M. J. (1991) Reasoning about binary topological relations. In: Günther, O., Schek, H.-J. (eds.) Advances in spatial databases. Lect. Notes Comput. Sci., vol. 525, pp. 141–160. Heidelberg, Germany: Springer. doi: 10.1007/3-540-54414-3_36

48 Cristani, M., Gabrielli, N. (2011) Practical issues of description logics for spatial reasoning. In: AAAI Spring Symposium, Stanford, CA, USA, 21–23 March 2011

49 Bai, L., Lao, S., Zhang, W., Jones, G. J. F., Smeaton, A. F. (2008) Video semantic content analysis framework based on ontology combined MPEG-7. In: Boujemaa, N., Detyniecki, M., Nürnberger, A. (eds.) Adaptive multimedia retrieval: retrieval, user, and semantics. Lect. Notes Comput. Sci., vol. 4918, pp. 237–250. Heidelberg, Germany: Springer. doi: 10.1007/978-3-540-79860-6_19

50 Lutz, C. (2001) Interval-based temporal reasoning with general TBoxes. In: 17th International Joint Conference on Artificial Intelligence, pp. 89–94. Morgan Kaufmann

51 Allen, J. F. (1983) Maintaining knowledge about temporal intervals. Commun. ACM 26(11):832–843. doi: 10.1145/182.358434

52 Artale, A., Franconi, E. (1998) A temporal description logic for reasoning about actions and plans. J. Artif. Intell. Res. 9(1):463–506

53 Liu, W., Xu, W., Wang, D., Liu, Z., Zhang, X. (2012) A temporal description logic for reasoning about action in event. Inf. Technol. J. 11(9):1211–1218. doi: 10.3923/itj.2012.1211.1218

54 Günsel, C., Wittmann, M. (2001) Towards an implementation of the temporal DL $\mathcal{TL}$-$\mathcal{ALC}$. In: 2001 International Description Logics Workshop. Stanford, CA, USA, 1–3 August 2001

55 Artale, A., Lutz, C. (2012) A correspondence between temporal description logics. J. Appl. Non-Class. Log. 14(1–2):209–233. doi: 10.3166/jancl.14.209-233

56 Baader, F., Lippmann, M. (2014) Runtime verification using the temporal description logic $\mathcal{ALC}$-$\mathcal{LTL}$ revisited. J. Appl. Log. 12(4):584–613. doi: 10.1016/j.jal.2014.09.001

57 Kamide, N. (2010) A compatible approach to temporal description logics. In: 23rd International Workshop on Description Logics, Waterloo, ON, Canada, 4–7 May 2010

58 Artale, A., Franconi, E., Wolter, F., Zakharyaschev, M. (2002) A temporal description logic for reasoning over conceptual schemas and queries. In: Flesca, S., Greco, S., Ianni, G., Leone, N. (eds.) Logics in artificial intelligence. Lect. Notes Comput. Sci., vol. 2424, pp. 98–110. Heidelberg, Germany: Springer. doi: 10.1007/3-540-45757-7_9

59 Hu, K., Yu, X., Li, Z., Zhu, H. (2010) The temporal description logic $\mathcal{TL}$-$\mathcal{SI}$ and its decidability algorithm. In: 2010 International Conference on Computational Aspects of Social Networks, pp. 575–578. New York: IEEE. doi: 10.1109/CASoN.2010.133

60 Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M. (2011) Tailoring temporal description logics for reasoning over temporal conceptual models. In: Tinelli, C., Sofronie-Stokkermans, V. (eds.) Frontiers of combining systems. Lect. Notes Comput. Sci., vol. 6989, pp. 1–11. Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-24364-6_1

61 Milea, V., Frasincar, F., Kaymak, U. (2012) tOWL: a temporal web ontology language. IEEE Trans. Syst. Man Cy. B 42(1):268–281. doi: 10.1109/TSMCB.2011.2162582

62 Wang, Y., Chang, L., Li, F., Gu, T. (2014) Verification of branch-time property based on dynamic description logic. In: Shi, Z., Wu, Z., Leake, D., Sattler, U. (eds.) Intelligent information processing. IFIP AICT, vol. 432, pp. 161–170. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-44980-6_18

63 Gutiérrez-Basulto, V., Jung, J. C., Schneider, T. (2014) Lightweight temporal description logics with rigid roles and restricted TBoxes. In: 24th International Joint Conference on Artificial Intelligence. Palo Alto: CAAAAI

64 Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M. (2015) Interval temporal description logics. In: 28th International Workshop on Description Logics, Aachen, Germany: Sun SITE Central Europe

65 Sanati, M. Y. (2015) A metric interval-based temporal description logic. Ph.D. thesis, McMaster University, Hamilton, ON, Canada

66 Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M. (2007) Reasoning over extended ER models. In: Parent, C., Schewe, K. D., Storey, V. C., Thalheim, B. (eds.) Conceptual modeling – ER 2007. Lect. Notes Comput. Sci., vol. 4801, 277–292. Heidelberg, Germany: Springer-Verlag. doi: 10.1007/978-3-540-75563-0_20

67 Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M. (2014) A cookbook for temporal conceptual data modeling with description logics. ACM Trans. Comput. Logic 15(3), article 25. doi: 10.1145/2629565

68 Sotnykova, A., Vangenot, C., Cullot, N., Bennacer, N., Aufaure, M.-A. (2005) Semantic mappings in description logics for spatio-temporal database schema integration. In: Spaccapietra, S., Zimányi, E. (eds.) Journal on data semantics III. Lect. Notes Comput. Sci., vol. 3534, pp. 143–167. Heidelberg, Germany: Springer. doi: 10.1007/11496168_7

69 Elleuch, N., Zarka, M., Ammar, A. B., Alimi, A. M. (2011) A fuzzy ontology-based framework for reasoning in visual video content analysis and indexing. In: 11th International Workshop on Multimedia Data Mining. San Diego, CA, USA, 21 August 2011. New York: ACM. doi: 10.1145/2237827.2237828

70 Stoilos, G., Stamou, G., Pan, J. Z. (2010) Fuzzy extensions of OWL: logical properties and reduction to fuzzy description logics. Int. J. Approx. Reason. 51(6):656–679. doi: 10.1016/j.ijar.2010.01.005

71 Carrive, J., Pachet, F., Ronfard, R. (1998) Using description logics for indexing audiovisual documents. In: 1998 International Workshop on Description Logics, Trento, Italy, 6–8 June 1998. Aachen, Germany: Sun SITE Central Europe

72 Hammiche, S., Benbernou, S., Vakali, A. (2005) A logic based approach for the multimedia data representation and retrieval. In: 7th IEEE International Symposium on Multimedia. Irvine, CA, USA, 14 December 2005. New York: IEEE. doi: 10.1109/ISM.2005.11

73 Bai, L., Lao, S., Jones, G. J. F., Smeaton, A. F. (2007) Video semantic content analysis based on ontology. In: International Machine Vision and Image Processing Conference, pp. 117–124. New York: IEEE. doi: 10.1109/IMVIP.2007.13

74 SanMiguel, J. C., Martinez, J. M., Garcia, Á. (2009) An ontology for event detection and its application in surveillance video. In: 6th IEEE international conference on advanced video and signal based surveillance, pp. 220–225. New York: IEEE. doi: 10.1109/AVSS.2009.28

75 Wang, X., Chang, L., Li, Z., Shi, Z. (2010) A dynamic description logic based system for video event detection. Front. Electr. Electron. Eng. China 5(2):137–142. doi: 10.1007/s11460-009-0078-y

76 Helaoui, R., Riboni, D., Niepert, M., Bettini, C., Stuckenschmidt, H. (2012) Towards activity recognition using probabilistic description logics. In: 26th AAAI Conference on Artificial Intelligence, Toronto, ON, Canada

77 Elbaşi, E. (2013) Fuzzy logic-based scenario recognition from video sequences. J. Appl. Res. Technol. 11(5):702–707. doi: 10.1016/S1665-6423(13)71578-5

78 Santofimia, M. J., Martinez-del-Rincon, J., Nebel, J.-C. (2014) Episodic reasoning for vision-based human action recognition. Sci. World J. 2014, article 270171. doi: 10.1155/2014/270171

79 Ronfard, R. (1997) Shot-level description and matching of video content. In: Jay Kuo, C.-C., Chang, S.-F., Gudivada, V. N. (eds.) Multimedia storage and archiving systems II. The International Society for Optical Engineering, Bellingham. doi: 10.1117/12.290366

80 Grutter, R., Bauer-Messmer, B. (2007) Towards spatial reasoning in the Semantic Web: a hybrid knowledge representation system architecture. LNG&C, pp. 349–364. Heidelberg, Germany: Springer. doi: 10.1007/978-3-540-72385-1_21

81 Batsakis, S., Petrakis, E. G. M., Tachmazidis, I., Antoniou, G. (2016) Temporal representation and reasoning in OWL 2, Semantic Web – Interoperability, Usability, Applicability, 2016

82 Bobillo, F., Straccia, U. (2011) Fuzzy ontology representation using OWL 2. Int. J. Approx. Reason. 52(7):1073–1094. doi: 10.1016/j.ijar.2011.05.003

83 Isaac, A., Troncy, R. (2004) Designing and using an audio-visual description core ontology. Workshop on Core Ontologies in Ontology Engineering. Northamptonshire, UK, 8 October 2004

84 Hunter, J. (2001) Adding multimedia to the Semantic Web—building an MPEG-7 ontology. 1st International Semantic Web Working Symposium. Stanford, CA, USA, 29 July–1 August 2001

85 Tsinaraki, C., Polydoros, P., Moumoutzis, N., Christodoulakis, S. (2004) Integration of OWL ontologies in MPEG-7 and TV-Anytime compliant semantic indexing. In: Persson, A., Stirna, J. (eds.) Advanced information systems engineering. Lect. Notes Comput. Sci., vol. 3084, pp. 398–413. Heidelberg, Germany: Springer. doi: 10.1007/978-3-540-25975-6_29

86 García, R., Celma, O. (2005) Semantic integration and retrieval of multimedia metadata. 5th International Workshop on Knowledge Markup and Semantic Annotation, Galway, Ireland, 7 November 2005

87 Blöhdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y., Handschuh, S., Kompatsiaris, Y., Staab, S., Strintzis, M. (2005) Semantic annotation of images and videos for multimedia analysis. In: Gómez-Pérez, A., Euzenat, J. (eds.) The Semantic Web: research and applications. Lect. Notes Comput. Sci., vol. 3532, pp. 592–607. Heidelberg, Germany: Springer. doi: 10.1007/11431053_40

88 Athanasiadis, T., Tzouvaras, V., Petridis, K., Precioso, F., Avrithis, Y., Kompatsiaris, Y. (2005) Using a multimedia ontology infrastructure for semantic annotation of multimedia content.

5th International Workshop on Knowledge Markup and Semantic Annotation, Galway, Ireland, 7 November 2005

89 Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Baumann, S., Vembu, S., Romanelli, M. (2007) DOLCE ergo SUMO: on foundational and domain models in the SmartWeb integrated ontology (SWIntO). J. Web Semant. Sci. Serv. Agents World Wide Web 5(3):156–174. doi: 10.1016/j.websem.2007.06.002

90 Dasiopoulou, S., Tzouvaras, V., Kompatsiaris, I., Strintzis, M. (2009) Capturing MPEG-7 semantics. In: Sicilia, M.-A., Lytras, M. D. (eds.) Metadata and semantics, pp. 113–122. New York: Springer.

91 Arndt, R., Troncy, R., Staab, S., Hardman, L. (2009) COMM: a core ontology for multimedia annotation. In: Staab, S., Studer, R. (eds.) Handbook on ontologies, pp. 403–421. Heidelberg, Germany: Springer. doi: 10.1007/978-3-540-92673-3_18

92 Saatho, C., Scherp, A. (2009) M3O: The Multimedia Metadata Ontology. 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies. Graz, Austria, 2 December 2009

93 Horvat, M., Bogunović, N., Ćosić, K. (2014) STIMONT: a core ontology for multimedia stimuli description. Multimed. Tools Appl. 73(3):1103–1127. doi: 10.1007/s11042-013-1624-4

94 Naphade, M., Smith, J. R., Tesic, J., Chang, S.-F., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J. (2006) Large-scale concept ontology for multimedia. IEEE MultiMedia 13(3):86–91. doi: 10.1109/MMUL.2006.63

95 Zha, Z.-J., Mei, T., Zheng, Y.-T., Wang, Z., Hua, X.-S. (2012) A comprehensive representation scheme for video semantic ontology and its applications in semantic concept detection. Neurocomputing 95:29–39. doi: 10.1016/j.neucom.2011.05.044

96 Hogenboom, F., Borgman, B., Frasincar, F., Kaymak, U. (2010) Spatial knowledge representation on the Semantic Web. 2010 IEEE Fourth International Conference on Semantic Computing. Pittsburgh, PA, USA, 22–24 September 2010. New York: IEEE. doi: 10.1109/ICSC.2010.31

97 Yildirim, Y., Yazici, A., Yilmaz, T. (2013) Automatic semantic content extraction in videos using a fuzzy ontology and rule-based model. IEEE T. Knowl. Data En. 25(1):47–61. doi: 10.1109/TKDE.2011.189

98 Sikos, L. F. (2011) Advanced (X)HTML5 metadata and semantics for Web 3.0 videos. DESIDOC J. Library Inf. Technol. 31(4):247–252. doi: 10.14429/djlit.31.4.1105

99 Lasierra, N., Roldán, F., Alesanco, A., García, J. (2014) Towards improving usage and management of supplies in healthcare: an ontology-based solution for sharing knowledge. Expert Syst. Appl. 41(14): 6261–6273. doi: 10.1016/j.eswa.2014.04.023

100 Hu, B., Dasmahapatra, S., Dupplaw, D., Lewis, P., Shadbolt, N. (2007) Reflections on a medical ontology. Int. J. Hum.-Comput. St. 65(7):569–582. doi: 10.1016/j.ijhcs.2007.02.005

101 Martínez-Romero, M., Vázquez-Naya, J. M., Pereira, J., Pazos, A. (2014) BiOSS: a system for biomedical ontology selection. Comput. Meth. Prog. Bio. 114(1):125–140. doi: 10.1016/j.cmpb.2014.01.020

102 Konopka, B. M. (2015) Biomedical ontologies—a review. Biocybern. Biomed. Eng. 35(2):75–86. doi: 10.1016/j.bbe.2014.06.002

103 Brochhausen, M., Spear, A. D., Cocos, C., Weiler, G., Martín, L., Anguita, A., Stenzhorn, H., Daskalaki, E., Schera, F., Schwarz, U., Sfakianakis, S., Kiefer, S., Dörr, M., Graf, N., Tsiknakis, M. (2011) The ACGT Master Ontology and its applications – Towards an ontology-driven cancer research and management system. J. Biomed. Inform. 44(1):8-25. doi: 10.1016/j.jbi.2010.04.008

104 Smith, B., Arabandi, S., Brochhausen, M., Calhoun, M., Ciccarese, P., Doyle, S., Gibaud, B., Goldberg, I., Kahn, C. E., Overton, J., Tomaszewski, J., Gurcan, M. (2015) Biomedical imaging ontologies: a survey and proposal for future work. J. Pathol. Inform. 6:37. doi: 10.4103/2153-3539.159214

105 Rubin, D. L., Mongkolwat, P., Kleper, V., Supekar, K., Channin, D. S. (2009) Annotation and image markup: accessing and interoperating with the semantic content in medical imaging. IEEE Intell. Syst. 24(1):57–65. doi: 10.1109/MIS.2009.3

106 Budovec, J. J., Lam, C. A., Kahn, C. E. (2013) Informatics in radiology—Radiology Gamuts Ontology: differential diagnosis for the Semantic Web. Radiographics 34(1):254–264. doi: 10.1148/rg.341135036

107 Lasbleiz, J., Saint-Jalmes, H., Duvauferrier, R., Burgun, A. (2011) Creating a magnetic resonance imaging ontology. Stud. Health Technol. Inform. 169:784–788

108 Temal, L., Dojat, M., Kassel, G., Gibaud, B. (2008) Towards an ontology for sharing medical images and regions of interest in neuroimaging. J. Biomed. Inform. 41(5):766–778. doi: 10.1016/j.jbi.2008.03.002

109 Cudré-Mauroux, P., Enchev, I., Fundatureanu, S., Groth, P., Haque, A., Harth, A., Keppmann, F. L., Miranker, D., Sequeda, J., Wylot, M. (2013) NoSQL databases for RDF: an empirical evaluation. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J. X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web – ISWC 2013. Lect. Notes Comput. Sci., vol. 8219, pp. 310–325. Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-41338-4_20

110 Bizer, C., Heath, T., Berners-Lee, T. (2009) Linked Data—the story so far. Semantic Web Inform. Syst. 5(3):1–22. doi: 10.4018/jswis.2009081901

111 Berners-Lee, T. (2009) Linked Data–design issues. World Wide Web Consortium. https://www.w3.org/DesignIssues/LinkedData.html. Accessed 29 September 2017

112 Choudhury, S., Breslin, J. G., Passant, A. (2009) Enrichment and ranking of the YouTube tag space and integration with the Linked Data Cloud. In: The Semantic Web – ISWC 2009. Lect. Notes Comput. Sci., vol. 5823, pp. 747–762. Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-04930-9_47

113 Yang, N.-C., Chang, W.-H., Kuo, C.-M., Li, T.-H. (2008) A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval. J. Vis. Commun. Image R. 19:92–105. doi: 10.1016/j.jvcir.2007.05.003

114 Xu, F., Zhang, Y.-J. (2006) Evaluation and comparison of texture descriptors proposed in MPEG-7. J. Vis. Commun. Image R. 17:701–716. doi: 10.1016/j.jvcir.2005.10.002

115 Viola, P., Jones, M. (2001) Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 511–518. Kauai, HI, USA, 8–14 December 2001. doi: 10.1109/CVPR.2001.990517

116 Lienhart, R., Maydt, J. (2002) An extended set of Haar-like features for rapid object detection. In: 2002 International Conference on Image Processing, pp. 900–903. New York, 22–25 September 2002. doi: 10.1109/ICIP.2002.1038171

117 Lowe, D. G. (2004) Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2):91–110. doi: 10.1023/B:VISI.0000029664.99615.94

118 Khedher, M. I., El Yacoubi, M. A. (2015) Local sparse representation based interest point matching for person re-identification. In: Arik, S., Huang, T., Lai, W., Liu, Q. (eds.) Neural information processing. Lect. Notes Comput. Sc., vol. 9491, pp. 241–250. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-26555-1_28

119 Rublee, E., Rabaud, V., Konolige, K., Bradski, G. (2011) ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision, pp. 2564–2571, Barcelona, Spain, 6–13 November 2011. New York: IEEE. doi: 10.1109/ICCV.2011.6126544

120 Duong, T. H., Nguyen, N. T., Truong, H. B., Nguyen, V. H. (2015) A collaborative algorithm for semantic video annotation using a consensus-based social network analysis. Expert Syst. Appl. 42(1):246–258. doi: 10.1016/j.eswa.2014.07.046

121 Guo, K., Zhang, S. (2013) A semantic medical multimedia retrieval approach using ontology information hiding. Comput. Math. Meth. Med. 2013, Article ID 407917. doi: 10.1155/2013/407917

122 Ballan, L., Bertini, M., Del Bimbo, A., Serra, G. (2010) Semantic annotation of soccer videos by visual instance clustering and spatial/temporal reasoning in ontologies. Multimed. Tools Appl. 48:313–337. doi: 10.1007/s11042-009-0342-4

123 Jiang, Y.-G., Bhattacharya, S., Chang, S.-F., Shah, M. (2013) High-level event recognition in unconstrained videos. Int. J. Multimed. Info. Retr. 2:73–101. doi: 10.1007/s13735-012-0024-2

124 Shah, C. P. (2017) YAG laser vitreolysis vs sham YAG vitreolysis for symptomatic vitreous floaters: a randomized clinical trial. JAMA Ophthalmol. doi: 10.1001/jamaophthalmol.2017.2388

125 Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G. (2011) Event detection and recognition for semantic annotation of video. Multimed. Tools Appl. 51(1):279–302. doi: 10.1007/s11042-010-0643-7

126 Gómez-Romero, J., Patricio, M. A., García, J., Molina, J. M. (2010) Ontology-based context representation and reasoning for object tracking and scene interpretation in video. Expert Syst. Appl. 38:7494–7510. doi: 10.1016/j.eswa.2010.12.118

127 Lombardo, V., Pizzo, A. (2014) Ontology–based visualization of characters' intentions. In: Mitchell, A., Fernández-Vara, C., Thue, D. (eds.) Interactive storytelling. ICIDS 2014. Lect. Notes Comput. Sci., vol. 8832, pp. 176–187. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-12337-0_18

128 Poppe, C., Martens, G., De Potter, P., Van de Walle, R. (2012) Semantic web technologies for video surveillance metadata. Multimed. Tools Appl. 56(3):439–467. doi: 10.1007/s11042-010-0600-5

129 Bohlken, W., Neumann, B., Hotz, L., Koopmann, P. (2011) Ontology-based realtime activity monitoring using beam search. In: Crowley, J. L., Draper, B. A., Thonnat, M. (eds.) Computer vision systems. ICVS 2011. Lect. Notes Comput. Sci., vol. 6962, pp. 112–121. Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-23968-7_12

130 Elleuch, N., Zarka, M., Ammar, A. B., Alimi, A. M. (2011) A fuzzy ontology-based framework for reasoning in visual video content analysis and indexing. In: 11th International Workshop on Multimedia Data Mining, San Diego, CA, USA, 21–24 August 2011, Article 1. doi: 10.1145/2237827.2237828

131 Hunter, J., Newmarch, J. (1999) An indexing, browsing, search and retrieval system for audio-visual libraries. In: Abiteboul, S., Vercoustre, A. M. (eds.) Research and advanced technology

for digital libraries. ECDL 1999. Lect. Notes Comput. Sci., vol. 1696, pp. 76–91. Heidelberg, Germany: Springer. doi: 10.1007/3-540-48155-9_7

132 Heggland, J. (2002) Ontolog: temporal annotation using ad hoc ontologies and application profiles. In: Agosti, M., Thanos, C. (eds.) Research and advanced technology for digital libraries. ECDL 2002. Lect. Notes Comput. Sc., vol. 2458, pp. 118–128. Heidelberg, Germany: Springer. doi: 10.1007/3-540-45747-X_9

133 Bertini, M., Del Bimbo, A., Torniai, C., Cucchiara, R., Grana, C. (2006) MOM: Multimedia Ontology Manager. A framework for automatic annotation and semantic retrieval of video sequences. ACM Multimedia 2006, pp. 787–788

134 Haslhofer, B., Jochum, W., King, R., Sadilek, C., Schellner, K. (2009) The LEMO annotation framework: weaving multimedia annotations with the web. Int. J. Digital. Libr. 10(1):15–32. doi: 10.1007/s00799-009-0050-8

135 Weiss, W., Bürger, T., Villa, R., Punitha, P., Halb, W. (2009) Statement-based semantic annotation of media resources. In: Chua, T. S., Kompatsiaris, Y., Mérialdo, B., Haas, W., Thallinger, G., Bailer, W. (eds.) Semantic multimedia. SAMT 2009. Lect. Notes Comput. Sci., vol. 5887, pp. 52–64. Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-10543-2_7

136 Steiner, T., Hausenblas, M. (2010) SemWebVid—Making video a first class Semantic Web citizen and a first class web bourgeois. In: Ninth International Semantic Web Conference, Shanghai, China, 7–11 November 2010

137 Choudhury, S., Breslin, J. G. (2010) Enriching videos with light semantics. In: Fourth International Conference on Advances in Semantic Processing, pp. 126–131. Florence, Italy, 25–30 October 2010

138 Haslhofer, B., Momeni, E., Gay, M., Simon, R. (2010) Augmenting Europeana content with Linked Data resources. In: 6th International Conference on Semantic Systems, Graz, Austria, 1–3 September 2010, Article 40. doi: 10.1145/1839707.1839757

139 Bertini, M., d'Amico, G., Ferracani, A., Meoni, M., Serra, G. (2010) Sirio, Orione and Pan: an integrated web system for ontology-based video search and annotation. In: ACM International-

al Conference on Multimedia, pp. 1625–1628. Firenze, Italy, 25–29 October 2010. doi: 10.1145/1873951.1874305

140 Simon, R., Jung, J., Haslhofer, B. (2011) The YUMA media annotation framework. In: Gradmann, S., Borri, F., Meghini, C., Schuldt, H. (eds) Research and advanced technology for digital libraries. TPDL 2011. Lect. Notes Comput. Sci., vol. 6966, pp. 434–437. Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-24469-8_43

141 Nixon, L., Bauer, M., Bara, C., Kurz, T., Pereira, J. (2012) ConnectME: semantic tools for enriching online video with web content. In: 8th International Conference on Semantic Systems, pp. 55–62. Graz, 5–7 September 2012

142 Grassi, M., Morbidoni, C., Nucci, M. (2012) A collaborative video annotation system based on Semantic Web technologies. Cogn. Comput. 4(4):497–514. doi: 10.1007/s12559-012-9172-1

143 Bellini, P., Nesi, P., Serena, M. (2015) MyStoryPlayer: experiencing multiple audiovisual content for education and training. Multimed Tools Appl. 74:8219–8259. doi: 10.1007/s11042-014-2052-9

144 Hitzler, P., Krötzsch, M., Rudolph, S. (2009). Foundations of semantic web technologies. Boca Raton, FL: CRC Press

145 Simou, N., Tzouvaras, V., Avrithis, Y., Stamou, G., Kollias, S. (2005). A visual descriptor ontology for multimedia reasoning. 6th International Workshop on Image Analysis for Multimedia Interactive Services, Montreux, April 2005

146 Sikos, L. F. (2011) Advanced (X)HTML5 metadata and semantics for Web 3.0 videos. DESIDOC J. Libr. Inform. Technol. 31(4):247–252. doi: 10.14429/djlit.31.4.1105

147 Whitehead, A. N., Russell, B. (1925) Principia Mathematica (second edition). Cambridge: Cambridge University Press

148 Motik, B., Sattler, U., Studer, R. (2005) Query answering for OWL-DL with rules. J. Web Semant. Sci. Serv. Agents World Wide Web 3(1):41–60. doi: 10.1016/j.websem.2005.05.001

149 Simperl, E. (2009) Reusing ontologies on the Semantic Web: a feasibility study. Data Knowl. Eng. 68(10):905–925. doi: 10.1016/j.datak.2009.02.002

150 Suárez-Figueroa, M. C., Atemezing, G. A., Corcho, O. (2013) The landscape of multimedia ontologies in the last decade. Multimed. Tools Appl. 62(2):377–399. doi: 10.1007/s11042-011-0905-z

151. González, G. R. (2005) A Semantic Web approach to digital rights management. Ph.D. thesis, Pompeu Fabra University, Barcelona, Spain, 2005. http://rhizomik.net/html/~roberto/thesis/Thesis.pdf

152. Gruber, T. R. (1995) Towards principles for the design of ontologies used for knowledge sharing? Int. J. Hum.-Comput. St. 43(5–6):907–928. doi: 10.1006/ijhc.1995.1081

153 Smith, J. R. (2015) Semantics of visual discrimination. International Conference on Computing, Networking and Communications, Anaheim, CA, USA, 16–19 February 2015

154 Levy, M. A., O'Connor, M. J., Rubin, D. L. (2009) Semantic reasoning with image annotations for tumor assessment. AMIA Annual Symposium Proceedings, pp. 359–363

155 Mechouche, A., Golbreich, C., Morandi, X., Gibaud, B. (2008) Ontology-based annotation of brain MRI images. AMIA Annual Symposium Proceedings, pp. 460–464

156 Luque, E. F., Rubin, D. L., Moreira, D. A. (2015) Automatic classification of cancer tumors using image annotations and ontologies. 28th International Symposium on Computer-Based Medical Systems, Sao Carlos, Brazil. doi: 10.1109/CBMS.2015.8

157 Bhuvaneswari, K. S., Geetha, P. (2014) Semantic segmentation and categorization of brain MRI images for glioma grading. J. Med. Imag. Health In. 4(4):554–566. doi: 10.1166/jmihi.2014.1286

158 Zillner, S., Sonntag, D. (2012) Image metadata reasoning for improved clinical decision support. Netw. Model Anal. Health Inform. Bioinforma. 1(1-2):37-46. Doi: 10.1007/s13721-012-0003-9

159 Web3D Consortium (2017) Medically accurate 3D printed hearts. http://www.web3d.org/case-studies/preoperative-planning-3d-printed-hearts/medically-accurate-3d-printed-hearts. Accessed 29 September 2017

160 Olivieri, L. J., Su, L., Hynes, C. F., Krieger, A., Alfares, F. A., Ramakrishnan, K., Zurakowski, D., Marshall, M. B., Kim, P. C., Jonas, R. A., Nath, D. S. (2016) Just-in-time simulation training using 3-D printed cardiac models after congenital cardiac surgery. World J. Pediatric Congenit. Heart Surg. 7(2):164–168. doi: 10.1177/2150135115623961

161 Gao, B., Zheng, H., Zhang, S. (2009) An overview of semantics processing in content-based 3D model retrieval. 2009 International Conference on Artificial Intelligence and Computational Intelligence, pp. 54–59. Shanghai, China, 7–8 November 2009. doi: 10.1109/AICI.2009.482

162 Callet, P. (2014) 3D reconstruction from 3D cultural heritage models. In: Ioannides, M., Quak, E. (Eds.) 3D research challenges in cultural heritage, pp. 135–142. Heidelberg, Germany: Springer. doi: 10.1007/978-3-662-44630-0_10

163 Sikos, L. F. (2015) Mastering structured data on the Semantic Web: from HTML5 Microdata to Linked Open Data. New York: Apress. doi: 10.1007/978-1-4842-1049-9

164 Flotyński, J., Walczak, K. (2015) Conceptual knowledge-based modeling of interactive 3D content. Vis. Comput. 31(10):1287–1306. doi: 10.1007/s00371-014-1011-9

165 Wang, X., Lv, T., Wang, S., Wang, Z. (2008) An ontology and SWRL-based 3D model retrieval system. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (Eds.) Information retrieval technology, pp. 335–344. Heidelberg, Germany: Springer. doi: 10.1007/978-3-540-68636-1_32

166 Koenderink, N. J. J. P., Top, J. L., van Vliet, L. J. (2006) Supporting knowledge-intensive inspection tasks with application ontologies. Int. J. Hum.-Comput. St. 64(10):974–983. doi: 10.1016/j.ijhcs.2006.05.004

167 Vasilakis, G., Garcia-Rojas, A., Papaleo, L., Catalano, C. E., Spagnuolo, M., Robbiano, F., Vavalis, M., Pitikakis, M. (2007) A common ontology for multi-dimensional shapes. In: 1st

International Workshop on Multimedia Annotation and Retrieval Enabled by Shared Ontologies, pp. 31–43. Genova, Italy, 5 December 2007

168 Luukkainen, M., Karhela, T. (2007) Ontology approach for co-use of 3D plant modelling and large scale process simulation. 48th Scandinavian Conference on Simulation and Modeling, Göteborg, Sweden, 30–31 October 2007

169 Marinc, A. Stockloew, C., Tazari, S. (2012) 3D interaction in AAL environments based on ontologies. In: AAL-Kongress 2012, pp. 289–302. Berlin, Germany, 24–25 January 2012. doi: 10.1007/978-3-642-27491-6_21

170 Verdonck, M., Gailly, F., Poels, G. (2014) 3D vs. 4D ontologies in enterprise modeling. In: Indulska, M., Purao, S. (Eds.) Advances in conceptual modeling, pp. 13–22. Cham, Switzerland: Springer. doi: 10.1007/978-3-319-12256-4_2

171 Lee, S., Lee, S., Seo, D., Yoo, K.-H., Kim, S. (2015) Development of ontology and 3D software for the diseases of spine. Adv. Multimed. 2015, Article ID 420848. doi: 10.1155/2015/420848

172 Gutiérrez, M., García-Rojas, A., Thalmann, D., Vexo, F., Moccozet, L., Magnenat-Thalmann, N., Mortara, M., Spagnuolo, M. (2007) An ontology of virtual humans: incorporating semantics into human shapes. Vis. Comput. 23(3):207–218. doi: 10.1007/s00371-006-0093-4

173 Kalogerakis, E., Christodoulakis, S., Moumoutzis, N. (2006) Coupling ontologies with graphics content for knowledge-driven visualization. In: IEEE Virtual Reality Conference, pp. 43–50. Alexandria, VA, USA, 25–29 March 2006. doi: 10.1109/VR.2006.41

174 Niccolucci, F., D'Andrea, A. (2006) An ontology for 3D cultural objects. 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, Nicosia, Cyprus, 30 October–4 November 2006

175 Boccon-Gibod, H. (2011) A story of an initiative to extend the X3D CAD profile adding BREP solid representation. 16th International Conference on 3D Web Technology, Paris, France, 20–22 June 2011

176 Petit, M., Boccon-Gibod, H., Mouton, C. (2012) Evaluating the X3D Schema with Semantic Web tools. In: 17th International Conference on 3D Web Technology, pp. 131–138. Los Angeles, 4–5 August 2012. doi: 10.1145/2338714.2338737

177 Berrueta, D., Phipps, J. (eds.) (2008) Best practice recipes for publishing RDF vocabularies. https://www.w3.org/TR/swbp-vocab-pub/. Accessed 30 September 2017

178 Orgel, T., Höffernig, M., Bailer, W., Russegger, S. (2015) A metadata model and mapping approach for facilitating access to heterogeneous cultural heritage assets. Int. J. Digit. Libr. 15(2):189–207. doi: 10.1007/s00799-015-0138-2

179 Rabattu, P.-Y., Massé, B., Ulliana, F., Rousset, M.-C., Rohmer, D., Léon, J.-C., Palombi, O. (2015) My Corporis Fabrica Embryo: an ontology-based 3D spatio-temporal modeling of human embryo development. J. Biomed. Semant. 6(36). doi: 10.1186/s13326-015-0034-0

180 Hayes, P. (ed.) (2004) RDFS entailment rules. In: RDF semantics. https://www.w3.org/TR/2004/REC-rdf-mt-20040210/#RDFSRules. Accessed 30 September 2017

181 Ter Horst, H. J. (2005) Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Web Semant. Sci. Serv. Agents World Wide Web 3(2–3):79–115. doi: 10.1016/j.websem.2005.06.001

182 Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.) (2012) Reasoning in OWL 2 RL and RDF graphs using rules. In: OWL 2 Web Ontology Language profiles (2nd ed.). https://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules. Accessed 30 September 2017

183 Motik, B., Shearer, R., Horrocks, I. (2009) Hypertableau reasoning for description logics. J. Artif. Intell. Res. 36(1):165–228

184 Du, K., Shi, Y., Lei, B., Chen, J., Sun, M. (2016) A method of human action recognition based on spatio-temporal interest points and PLSA. 2016 International Conference on Industrial Informatics, Wuhan, China, 3–4 December 2016. doi: 10.1109/ICIICII.2016.0028

185 Kumar, C. R., Suguna, S. (2016) Visual semantic based 3D video retrieval system using HDFS. Data Min. Knowl. Disc. 10(8):3806–3825

186 Tangelder, J. W. H., Veltkamp, R. C. (2008) A survey of content-based 3D shape retrieval methods. Multimed. Tools Appl. 39, 441–471. doi: 10.1007/s11042-007-0181-0

187 Jia, J., Qin, Z., Zhang, Q., Chen, J., Liu, Y. (2008) An overview of content-based three-dimensional model retrieval methods. In: 2008 IEEE International Conference on System of Systems Engineering Proceedings, pp. 1–6. 2–4 June 2008, Singapore. New York: IEEE. doi: 10.1109/SYSOSE.2008.4724168

188 Qin, Z., Jia, J., Qin, J. (2008) Content-based 3D model retrieval: a survey. In: 2008 International Workshop on Content-Based Multimedia Indexing Proceedings, pp. 249–256. London, UK, 18–20 June 2008. New York: IEEE. doi: 10.1109/CBMI.2008.4564954