A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain

Leslie F. Sikos
Centre for Knowledge and Interaction Technologies, School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, Australia
Email: leslie.sikos@flinders.edu.au

Abstract: Because of the growing popularity of 3D modelling, there is a great demand for efficient mechanisms to automatically process 3D contents. Owing to the lack of semantics, however, most 3D scenes cannot be interpreted by software agents. 3D ontologies can provide formal definitions for 3D objects; however, many of them are semi-structured only, cover a narrow knowledge domain, do not provide comprehensive coverage for geometric primitives, and do not exploit the full expressivity of the implementation language. This paper presents the most comprehensive formally grounded 3D ontology to date that maps the entire XSD-based vocabulary of the industry standard X3D (ISO/IEC 19775–19777) to OWL 2, complemented by fundamental concepts and roles of the 3D modelling industry not covered by X3D. This upper ontology can be used for the representation, annotation, and efficient indexing of 3D models, and their retrieval by 3D characteristics rather than by associated category labels.

Keywords: 3D model semantics; multimedia ontology; medical 3D printing; feature-based 3D model retrieval; X3D; MPEG-7; medically accurate 3D models; content-based medical video retrieval.


Biographical notes: Leslie F. Sikos, PhD, is an Australian researcher specialising in the knowledge representation of multimedia resources, multimedia ontology engineering, and automated video scene interpretation via spatio-temporal reasoning and information fusion. He has worked in both academia and the industry, thereby acquiring hands-on skills in Semantic Web technologies and multimedia semantics. Being an internationally recognised expert in intelligent information systems, he is an invited speaker, author, reviewer, and editor.

This paper is a revised and expanded version of a paper entitled ‘Rich Semantics for Interactive 3D models of Cultural Artifacts’ presented at the ‘10th International Conference on Metadata and Semantics Research’, Göttingen, Germany, 23 November 2016.

1 Introduction

3D models play an important role in a wide range of applications, including engineering, medicine, 3D printing, scientific visualisation, education, and entertainment. However, the automated processing of 3D models is rather limited owing 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) (Web3D Consortium, 2017), 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 (Olivieri et al., 2016), 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, .stl, .ztl), classification category (e.g. people, cars, animals, landmarks, organs), 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, such as various histogram-based visual descriptors, 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), generalised 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 (Kazmi et al., 2013). 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 to video semantics (Sikos, 2016c), also poses a challenge for 3D model descriptions (Gao et al., 2009).

Photos and 2D videos of real-world objects, whether they are human organs, machines, tools, equipment, or cultural artefacts, often do not provide the information needed to fully understand their geometry and material-related features (Sikos, 2016b). 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 photomodelling (Callet, 2014). While many only think of the dimensions 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, knowledge, and experience, humans are capable of defining these rich semantics for 3D models in the form of high-level descriptors, many of which are correlated to modelling choices and parameters that have been standardised as terms, classes and properties, and parameter value ranges. The first prominent 3D annotation standard was the Virtual Reality Modelling 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 semi-structured XML Schema (XSD), which is machine-readable yet not machine-interpretable, making the implementations of the standard inefficient for automation, data sharing, and data reuse. These limitations can be addressed using Semantic Web standards, such as the Resource Description Framework (RDF), the RDF Schema (RDFS), and the Web Ontology Language (OWL), so that the structured representation of the corresponding concepts becomes machine-interpretable by linking them to their formal definitions and other, related concepts from the Linked Open Data (LOD) Cloud (Sikos, 2015), and by mapping them to geometric primitives (Floityński and Walczak, 2015). 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 (Wang et al., 2008).

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 (Sikos, 2016a).

2 Related work

Owing to the limitations of unstructured thesauri and semi-structured controlled vocabularies, this paper focuses exclusively on those 3D ontologies that provide structured data, specifically RDFS and OWL ontologies. Structured ontologies in the 3D modelling domain without X3D alignment include the Geometrical Application Ontology (Koenderink et al., 2006), the Common Shape Ontology (Vasilakis et al., 2007), the 3D Graphics Ontology, the Constructive Solid Geometry Modelling Ontology, the 3D Plant Modelling Ontology (Luukkainen and Karhela, 2007), 3D interaction ontologies (Marinc et al., 2012), the Kinect Ontology, the Ontology of 3D Visualisation Techniques (O3VT), and 3D and 4D enterprise modelling ontologies (Verdonck et al., 2014). 3D domain ontologies without X3D alignment cover a narrow knowledge domain in fields such as medicine (e.g. Spine Ontology (Lee et al., 2015)), engineering (e.g. Furniture Ontology), and human modelling (e.g. Ontology for Virtual Humans (Gutiérrez et al., 2007)). 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 solely for representing 3D models and animations, such as MPEG-7,11,12,13 and CityGML, there have been multiple attempts to map the XML Schema file of X3D to OWL.

The first OWL mapping of X3D, OntologyX3D, was created by mapping X3D node elements representing graphics and virtual reality concepts into OWL classes (Kalogerakis et al., 2006). 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 (Niccolucci and D’Andrea, 2006). 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. This mapping has been built by loading the source and target ontology schemata into the Ontology Mapping FRAMework Toolkit (MAFRA), 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 (Bocon-Gibod, 2011), and has been generated using XSLT (Petit et al., 2012). 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, resulted in a basic role hierarchy. While the resulting ontology overcame many issues of its predecessors and covered a later version of X3D, it also suffered from ontology engineering issues. For example, it defined property ranges with VRML and X3D data types, e.g. SFBool, MFInt32, rather than the globally deployed XML Schema data types, such as xsd:boolean and xsd:integer, which would have been a better choice to maximise interoperability. Moreover, the mapping inherently made the false assumption that X3D nodes logically correspond to OWL classes, and incorrectly mapped xsd:complexType declarations, which are supposed 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 data type properties in X3D, this approach was unsatisfactory for many X3D roles. Another issue of this mapping, and all the previous X3D mappings for that matter, are the inadequate description of the defined concepts and roles.

3 A novel ontology for 3D models and scenes

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 Modelling Ontology (3DMO), has been developed with standards alignment and mathematical grounding using a description logic (DL) formalism. 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.

3.1 Modelling challenges

Several stages of ontology engineering indicated modelling 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 ontology versioning and takes into account the proposed (but not yet standardised) 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) (Berrueta and Phipps, 2008).

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 data type 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. Owing 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 Modelling Ontology.

The naming conventions of X3D made property modelling 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 colour used by the blend mode constant, in the second case it corresponds to a set of RGBA colours. 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, which was addressed by modifying, and sometimes extending, their descriptions, and declaring them using Dublin Core (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 data type 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 data type 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 data types. 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 data type of the two, which was applicable only to those data types that are supersets of the other. To add context to the corresponding properties, the descriptions have been updated accordingly.

The X3D specification features proprietary data types, many of which, such as SFNode and SFColor, are inherited from VRML. Wherever possible, all data types have been converted to standard XSD data types to maximise interoperability. For example, xsd:Boolean has been used instead of x3d:SFFboolean, and the x3d:SFFVec3f 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 data type instead. The majority of X3D properties are data type properties, many of which have an array of permissible values (representing vectors or matrices) rather than just one value of a specific datatype. Such data type properties have been defined as xsd:complexType rather than the more specific but single-value data types, such as xsd:integer, xsd:float, xsd:string, or xsd:anyURI.

The discrepancy between the X3D terminology and the 3D modelling terms used in AutoDesk 3D modelling 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 differentiating between similar concepts.

3.2 Conceptual modelling with best practices
In contrast to the common practice of creating OWL ontologies in the ontology editor Protégé without formal grounding, the 3D Modelling Ontology has been developed manually and in accordance with ontology engineering best practices (Sikos, 2017b). Hence, the development of the ontology included the following steps: specification, knowledge acquisition, conceptualisation, term enumeration, concept hierarchy building, the definition of roles and the role hierarchy, individual inclusion, evaluation, and documentation.

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

The in-depth analysis of the X3D and X3DOM documentation,23 the X3D Object Model,24 and the X3D nodes, fields, and permissible property values in the XSD file of X3D 3.3 served as the basis for the conceptualisation, 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 serialisation,25 the 3D Modelling Ontology was written in Turtle,26 mainly because the RDF/XML serialisation would have been too verbose for representing the large number of concepts and properties.

As it was infeasible to use automated ontology engineering techniques, such as XSLT transformation, natural language processing-based concept extraction, statistical and machine learning techniques, during ontology engineering, the components of the 3D Modelling 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 a sharp contrast with those ontologies that have been generated using the OWL API employed by the Protégé GUI.

The terminological box (TBox) of the 3D Modelling 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: animated, t3dmo:span, t3dmo:designedBy, t3dmo:hasVertices, t3dmo:baseForm) used by 3D graphic designers have also been added to maximise the implementation potential.

3.3 Formal grounding
The most common variant of OWL, the DL flavour, is an implementation of a description logic; OWL DL roughly corresponds to $SH(CIF)^{67}$ and OWL 2 DL to $SROIQ^{70}$ (Sikos and Powers, 2015). The 3D Modelling Ontology is grounded in the $SROIQ^{70}$ description logic, which can be defined briefly as follows (Horrocks et al., 2006). Assume three countably finite and pairwise disjoint sets of atomic concepts ($N_C$), atomic roles ($N_R$), and individual names ($N_I$), denoted by Internationalised Resource Identifiers (IRIs). The set of
SRQITQ concept expressions is defined as $C := N_C \cap (C \cap D)$, where $C \subseteq D$ for concepts $C$ and $D$, or an individual assertion of the form $C(a)$, where $a \in N_D$ denote individual names, $C \in C$ denotes concept expressions, and $R \in R$ denotes roles, or a role assertion of the form $R \subseteq S$, where $R \in R$ denote roles, or for roles $R$, $R_n$, and $S$. A SRQITQ interpretation $I$ is a pair of the form $I = (\Delta^I, \cdot^I)$, where $\Delta^I$ is a nonempty set (the object domain), and $\cdot^I$ is the interpretation function, which includes the class interpretation function $\cdot^C$, which assigns a subset of the object domain to each class, i.e., $\cdot^C: A \in N_C \rightarrow A^I \subseteq \Delta^I$, the role interpretation function $\cdot^R$, which assigns a set of tuples over the object domain to each role, i.e., $\cdot^R: R \in N_R \rightarrow \Delta^I \times \Delta^I$, and the individual interpretation function $\cdot^I$, which maps each individual $a \in N_I$ to an element $a^I \subseteq \Delta^I$. This formalism enables complex reasoning tasks for all the 3D applications that utilise the 3D Modelling 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, while their problems than first-order predicate logic and the expressivity of both a concept and a role, the 3D Modelling Ontology applies an extended name and different capitalisation 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 data type properties. The 3D Modelling Ontology features proprietary data types with XSD-alignment for its data type 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. Owing to the number of role overlaps and the different meanings and characteristics of homonymous atomic roles, however, the X3D roles cannot be categorised efficiently in a taxonomical flat structure, or the incorrect representation of the majority of concept differences as role values.

Since the DL expressivity of the 3D Modelling Ontology is SRQITQ$^{79}$, it exploits all the mathematical constructors of the implementation language, OWL 2 DL. While this makes the worst-case 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, the panels and settings of AutoDesk 3ds Max, AutoDesk Maya, AutoDesk AutoCAD, and their official documentation at the AutoDesk Knowledge Network. Based on these terminologies, atomic concepts have been created with X3D-alignment, and enumerated for inclusion in the 3D Modelling 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 $C \subseteq D$, 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 Modelling Ontology applies an extended name and different capitalisation 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.

Table 1 Examples for description logic to OWL 2 translation

<table>
<thead>
<tr>
<th>DL syntax</th>
<th>Turtle syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3dmo:Torus rdfs:subClassOf</td>
<td>t3dmo:Torus rdfs:subClassOf</td>
</tr>
<tr>
<td>X3D SpatialGeometry rdfs:domain</td>
<td>X3D SpatialGeometry rdfs:domain</td>
</tr>
<tr>
<td>t3dmo:BoxX3DSpatialGeometryNode</td>
<td>t3dmo:BoxX3DSpatialGeometryNode</td>
</tr>
<tr>
<td>t3dmo:Pyramid rdfs:range</td>
<td>t3dmo:Pyramid rdfs:range</td>
</tr>
<tr>
<td>t3dmo:topRadius rdfs:domain</td>
<td>t3dmo:topRadius rdfs:domain</td>
</tr>
<tr>
<td>t3dmo:transparency</td>
<td>t3dmo:transparency</td>
</tr>
<tr>
<td>t3dmo:creatorOf</td>
<td>t3dmo:creatorOf</td>
</tr>
<tr>
<td>owl:ObjectProperty t3dmo:creatorOf</td>
<td>owl:ObjectProperty t3dmo:creatorOf</td>
</tr>
<tr>
<td>[ a rdfs:Datatype ;</td>
<td>[ a rdfs:Datatype ;</td>
</tr>
<tr>
<td>owl:onDatatype xsd:float</td>
<td>owl:onDatatype xsd:float</td>
</tr>
<tr>
<td>owl:withRestrictions</td>
<td>owl:withRestrictions</td>
</tr>
<tr>
<td>[ xsd:minInclusive &quot;0&quot;]</td>
<td>[ xsd:maxInclusive &quot;1&quot;]</td>
</tr>
<tr>
<td>transparency.</td>
<td>transparency.</td>
</tr>
<tr>
<td>rdfs:range t3dmo:zeroone ;</td>
<td>rdfs:range t3dmo:zeroone ;</td>
</tr>
<tr>
<td>rdfs:domain t3dmo:Material</td>
<td>rdfs:domain t3dmo:Material</td>
</tr>
</tbody>
</table>

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 data type properties. The 3D Modelling Ontology features proprietary data types with XSD-alignment for its data type 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. Owing to the number of role overlaps and the different meanings and characteristics of homonymous atomic roles, however, the X3D roles cannot be categorised 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 Modelling Ontology defines an assertional box (ABox) with individuals such as colours, materials, 3D computer graphics software, and design studios.

The first version of the 3D Modelling Ontology features 526 classes, 780 properties, and 85 individuals in the form of 3817 axioms, covering the entire vocabulary of X3D 3.3, and core concepts and roles of the 3D modelling industry.

4 Evaluation

The 3D Modelling Ontology has been evaluated according to the five ontology engineering principles of Gruber, the researcher who introduced ontologies in the context of artificial intelligence (Gruber, 1993):

- **Clarity**: the intended semantics of the 3D Modelling 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 US English spellchecker. Although the atomic concepts and roles defined in the ontology characterise the 3D modelling domain, their definition is independent from the modelling context.

- **Coherence**: the integrity of the 3D Modelling 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 data type, which is used extensively in the 3D Modelling 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 organisation that defined the X3D standard.

- **Minimal encoding bias**: in the 3D Modelling Ontology, the conceptualisation 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, serialisations, or file formats.

- **Minimal ontological commitment**: the 3D Modelling Ontology has been designed to be as lightweight as possible through the inclusion of truly relevant concepts, the correct use of namespaces, the shorthand notation of RDF triples of the same subject, and the use of the compact serialisation Turtle. De facto standard ontology terms have been reused from mainstream ontologies, such as Creative Commons, Dublin Core, DBpedia, and Schema.org, according to the Semantic Web best practices described by Simperl (2009).

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

The description logic expressivity of the 3D Modelling Ontology, $\mathcal{SROIQ}$, is significantly higher than that of other 3D ontologies, which makes the 3D Modelling Ontology the most expressive ontology to date in the 3D modelling domain and the straightforward choice for reasoning over 3D models and scenes (see Table 2).

### Table 2 Comparison of 3D ontologies

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Language</th>
<th>DL expressivity</th>
<th>X3D alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML 2.0 Mapping</td>
<td>OWL</td>
<td>$\mathcal{ALCHQ}$</td>
<td>-</td>
</tr>
<tr>
<td>Furniture Ontology</td>
<td>OWL 2</td>
<td>$\mathcal{ALCHQ}$</td>
<td>-</td>
</tr>
<tr>
<td>Gilson-Silva X3D Mapping</td>
<td>RDFS</td>
<td>$\mathcal{AC}$</td>
<td>+</td>
</tr>
<tr>
<td>Kinect Ontology</td>
<td>OWL 2</td>
<td>$\mathcal{ALCRIE}$</td>
<td>-</td>
</tr>
<tr>
<td>O3DVT</td>
<td>OWL 2</td>
<td>$\mathcal{ALCHQ}$</td>
<td>-</td>
</tr>
<tr>
<td>Ontology X3D</td>
<td>OWL</td>
<td>$\mathcal{ALC}$</td>
<td>+</td>
</tr>
<tr>
<td>X3D OWL Mapping of</td>
<td>OWL</td>
<td>$\mathcal{ALC}$</td>
<td>+</td>
</tr>
<tr>
<td>Petit et al.</td>
<td>OWL</td>
<td>$\mathcal{ALC}$</td>
<td>+</td>
</tr>
<tr>
<td>3D Modelling Ontology</td>
<td>OWL 2</td>
<td>$\mathcal{SROIQ}$</td>
<td>+</td>
</tr>
</tbody>
</table>

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

5 Case studies

The 3D Modelling Ontology has a potential in any 3D application that can benefit from the semantic enrichment of 3D models, two of which are presented here.

5.1 Case study 1: ontology-based 3D model indexing

The 3D Modelling Ontology has been evaluated in content-based 3D model retrieval by formally describing 3D models of human anatomy, retrieving the ones with particular 3D characteristics using the RDF query language SPARQL, and comparing the results with traditional keyword-based search.

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

To demonstrate how the 3D Modelling 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 Modelling Ontology (see Figure 1).
A novel ontology for 3D semantics

Figure 1  The 3D Modelling Ontology provides structured representation for 3D models with X3D-aligned descriptors for geometric primitives and 3D characteristics that constitute a model

@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 ;
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 .

While 3ds Max scenes can be exported to X3D using the InstantExport plugin of Fraunhofer IDG, it only supports semi-structured formats, such as XML and XHTML, and leaves rich semantics unexploited (by typically utilising less than ten 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 modelling terms in OWL 2 with DL-based formal grounding, the 3D Modelling Ontology is the most suitable ontology to date in the 3D modelling domain for interlinking 3D models and model segments with Linked Data. Structured 3D model representations correspond to RDF graphs which, when interlinked with LOD resource identifiers, naturally merge to the LOD Cloud. The structured representations obtained this way can be queried and updated from diverse data sources manually and programmatically using SPARQL (e.g. Orgel et al., 2015). 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 operation, such as the ones with more than 50,000 vertices, and order them alphabetically, as demonstrated in Figure 2.

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.

Based on the results of this case study, Table 3 compares the efficiency of the 3D Modelling Ontology in 3D model retrieval by characteristics with the performance of traditional keyword-based search.

The semantic descriptions utilising the 3D Modelling Ontology can be stored in LOD data sets, 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 Modelling 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.
Figure 2 Using a SPARQL query to retrieve medically accurate heart models from a 3D model repository to be 3D printed for preoperative surgical discussions

Table 3 Precision and recall with keywords, previous 3D ontologies, and the 3D Modelling Ontology

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>OntologyX3D</td>
<td>0.72</td>
<td>0.75</td>
</tr>
<tr>
<td>Gilson-Silva Mapping</td>
<td>0.66</td>
<td>0.69</td>
</tr>
<tr>
<td>Mapping of Petit et al.</td>
<td>0.77</td>
<td>0.80</td>
</tr>
<tr>
<td>3DMO</td>
<td>0.91</td>
<td>0.87</td>
</tr>
</tbody>
</table>

5.2 Case study 2: content-based video retrieval

Accurate and detailed medical 3D animations and their spatio-temporal representation can visualise specific details, functions, and development of human organs and complex scientific concepts for medical training (Rabattu et al., 2015). 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 Modelling Ontology, together with spatio-temporal annotation of video segments and their regions of interest, the associated semantics can be efficiently captured and indexed (Sikos, 2017a).

In this case study, cardiology animations have been used to compare the concept coverage of previous X3D-based ontologies to that of the 3D Modelling 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 spatio-temporal segmentation using MPEG-7 terms and Media Fragment URIs (see Figure 3), and the 3D models depicted in the scenes with previous X3D-based ontologies and the 3D Modelling Ontology.
Once annotated, complex reasoning tasks have been performed on the represented 3D models of the video scenes using the RDFS entailment rules (Hayes, 2004), the Ter Horst reasoning rules (Ter Horst, 2005), and the OWL reasoning rules (Motik et al., 2012), by employing the hypertableau calculus (Motik et al., 2009).

The results have been mapped to Linked Data definitions, such as dbpedia:Mitral_valve_replacement and dbpedia:Ventricle_(heart), which can be utilised in intelligent video recommendations for medical training and hypervideo applications that display rich semantics about the represented models during playback (Sikos, 2017d).

Table 4 demonstrates the impact of 3D Modelling Ontology terms on the retrieval of 3D models with predetermined 3D characteristics from animations about the Tendyne mitral valve implant and the Mardil Ventouch ventricular reshaping device.37

<table>
<thead>
<tr>
<th>Model</th>
<th>OntologyX3D</th>
<th>3DMO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Tendyne</td>
<td>0.75</td>
<td>0.70</td>
</tr>
<tr>
<td>Mardil Ventouch</td>
<td>0.68</td>
<td>0.63</td>
</tr>
</tbody>
</table>

The results indicate that the 3D Modelling Ontology provides a more comprehensive coverage of 3D modelling terms than its predecessors, such as OntologyX3D or the OWL mapping of X3D by Petit et al. Note, however, that the semantic enrichment of videos with structured annotations about the represented 3D space and its features using terms from the 3D Modelling Ontology is best used in combination with domain ontologies and common-sense knowledge bases, similar to every other upper ontology (Sikos, 2017c). The 3D Modelling Ontology can also provide the formally described background knowledge to complement other content-based indexing approaches, such as the ones that utilise the 3D Extended Histogram of Oriented Gradients (3DHOG) Descriptor (Du et al., 2016), the 3D Distance Field Descriptor (3D DFD), bag of visual words (BOVW), or 3D co-occurrence matrices (Kumar and Suguna, 2016).

6 Conclusions and future work

Reasoning over the formal representation of 3D models helps 3D models reach their full potential in medicine, education, CAD applications, virtual museums, and other 3D applications. Rich semantics enable efficient indexing and retrieval of 3D models and scenes, however, until now no controlled vocabulary or ontology has been released with the capability to provide complex structured descriptors in this domain. The vocabulary of one of the most comprehensive 3D standards, X3D, was no exception, because it was designed in XML Schema as a semi-structured vocabulary.

Multiple attempts have been made to bridge, or at least narrow, the semantic gap in 3D model representation via mapping the XSD vocabulary of X3D to a structured RDFS or OWL ontology. However, previous approaches did not employ formal grounding, inherited inconsistency and conceptual ambiguity issues from the original vocabulary of the standard, and most of them did not provide a full coverage of the X3D terminology, which has also been evolved and significantly extended since their release. The 3D Modelling Ontology presented in this paper overcomes the previous limitations, and features full coverage for X3D 3.3 and X3DOM terms, complemented by 3D modelling terms of the industry. The 3D Modelling Ontology exploits all the mathematical constructors available in OWL 2, which makes it the most expressive 3D ontology to date. This feature enables advanced reasoning tasks that have been unavailable with previous ontologies. The integrity of the proposed ontology has been tested with industry-leading reasoners, and the correctness verified by the developers of the X3D standard. The implementation potential has been evaluated through the semantic enrichment and retrieval of
3D models. The case studies have confirmed that the exploitation of the 3D Modelling Ontology outperforms traditional keyword-based, metadata-based, and previous 3D ontology-based 3D model indexing techniques.

The semantics of 3D scenes can be detailed further by complementing the 3DMO annotations with highly specialised domain ontology terms, and interlinking them with concepts of the LOD Cloud to obtain sophisticated high-level descriptors, rather than just the parameters of vertices and edges of the 3D polygons and basic metadata.

Ongoing discussions with the Web3D Consortium indicated mutual interest for a potential collaboration to develop the next version of the 3D Modelling Ontology based on X3D 4.0, with improved conceptual modelling and role hierarchy, and X3D versioning for each class and property.

Acknowledgement

I would like to express my gratitude to the Web3D Consortium, in particular Don Brutzman, Christophe Mouton, and Vincent Marchetti of the Web3D Board of Directors for their invaluable comments, and the co-chair of the X3D Working Group, Roy Walmsley, for reviewing the OWL 2 implementation of the 3D Modelling Ontology.

References


A novel ontology for 3D semantics


Notes

1 Low-poly polygon meshes have a relatively small number of polygons to be used for high frame rate applications, such as computer games, and high-poly meshes contain a large number of polygons to provide superb surface detail, regardless of the time requirement of rendering (e.g. characters of computer-animated films from Hollywood).

2 .max and .3ds are used by Autodesk 3ds Max, .obj is an open 3D geometry definition file format developed by Wavefront Technologies, .blend is created in Blender, .fbx can be exported from many popular 3D applications, .stl (STereoLithography) is a file format native to CAD software created by 3D Systems and is widely used in 3D printing, and .ztl is the extension of the ZBrush ZTool Native File.

3 A machine-readable property-value pair does not necessarily provide semantics (e.g. a controlled vocabulary written in XML defines the property value of Property X to be the floating-point number 0.8). In contrast, a machine-interpretable property-value pair can be processed by software agents (e.g. an OWL ontology defines the property value of the directional light intensity property to be the floating-point number 0.8, along with the permissible values of that property to be normalised floating-point numbers).

4 https://www.w3.org/TR/rdf11-concepts/

5 https://www.w3.org/TR/rdf-schema/

6 https://www.w3.org/TR/ow2-overview/

7 http://lod-cloud.net

8 http://users.abo.fi/rowikstr/KinetOntology/KinetOntology.owl

9 http://cui.unige.ch/isi/onto/2010/12/furniture.owl

10 http://cui.unige.ch/isi/onto/2010/12/furniture.owl

11 http://rhizomik.net/ontologies/2005/03/Mpeg7-2001.owl

12 http://multimedia.semanticweb.org/COMM/visual.owl

13 http://mpeg7.org/mpeg7.ttl

14 http://cui.unige.ch/isi/onto/citygml2.0.owl

15 http://people.cs.umass.edu/~kalo/papers/graphicsOntologies/ OntologyX3D.zip

16 http://www.cs.swan.ac.uk/~csowen/SVGtoX3D/examples/ X3D_OntologyRDFS.htm

17 http://mafra-toolkit.sourceforge.net

18 A scalable HTML/X3D integration model.
The VRML-based X3D data types starting with SF correspond to one permissible value of the declared type, while MF indicates that multiple values are allowed for the corresponding property, whether they are floating-point numbers, integers, string literals, or URIs.

A product of Tendyne Holdings.

A trademark of Mardil Medical.