X3D Ontology for Semantic Web

X3D Semantic Web Working Group

Don Brutzman

brutzman@nps.edu

9 December 2019
The X3D Semantic Web Working Group mission is to publish models to the Web using X3D in order to best gain Web interoperability and enable intelligent 3D applications, feature-based 3D model querying, and reasoning over 3D scenes.

Motivations

• Establish best practices for metadata and semantic information/relationships using X3D as a Web-based presentation layer.

• Enable authors to utilize the power of X3Dv4 and HTML5/DOM together in any Web page utilizing a family of specifications and practices provided by the Semantic Web, such as HTML5 Microdata (microformats) and Linked Open Data, (possibly) MPEG-7 and related references.

• Align the X3Dv4 specification with these standards as best possible to further enable the Digital Publishing industry and communities.

• Describe value proposition for utilizing semantic information in concert with archival export, publishing, visualization, and printing of any 3D model as X3D.
Goals

• Enable more effective indexing, search, comparison, and analysis of X3D models through the advanced use of metadata, shape geometry, etc.

• Create and autogenerate X3Dv4 OWL ontology from X3D Unified Object Model (X3DUOM) using best-practice design patterns, starting with those shown by prior published work.

• Support various Web3D Working Groups including Computer-Aided Design (CAD), 3D printing/scanning, Medical, Cultural and Natural Heritage, Humanoid Animation (HAnim) design work may consider other potential domains such as Building Information Models (BIM), etc.

• Build list of domain-specific controlled vocabularies and taxonomies suitable for use with X3D Ontology.

• Maximize interoperability with Semantic Web standards for greatest possible reuse, integration with Web.

Outcomes

• Create appropriate specifications and recommended practices.

• Build suite of exemplars and tools exposed through various portals.

• Publish X3D Ontology for semantic queries, products in multiple formats: RDF/OWL, Terse RDF Triple Language (Turtle), JSON-LD.
The X3D Ontology for Semantic Web provides terms of reference for semantic query of X3D models.

**Motivation**

The X3D Semantic Web Working Group mission is to publish models to the Web using X3D in order to best gain Web interoperability and enable intelligent 3D applications, feature-based 3D model querying, and reasoning over 3D scenes.

Motivating insights:

"The answer to your question is the response to the query." Jim Hendler and Dean Allemang

"Trying to use the Semantic Web without SPARQL is like trying to use a relational database without SQL." Tim Berners-Lee

"The proof of the pudding is in the eating." Wiktionary

[https://www.web3d.org/x3d/content/semantics/semantics.html](https://www.web3d.org/x3d/content/semantics/semantics.html)
What is the X3D Ontology?

• A thorough representation of X3D Architecture using RDF and OWL.
  • RDF: Resource Description Framework
  • OWL: Web Ontology Language
  • Expressed in Turtle form, the Terse Triple Language (.ttl)

• A strictly typed definition of all X3D nodes, fields and data types.

• A way to “explode” an X3D model into subject-predicate-object triples.

• Basis for model search and query reasoning when
  • Example X3D scene is also converted into .ttl form
  • Queries are written in SPARQL, the SPARQL Protocol and RDF Query Language

• Experimental, but design examples are converging satisfactorily.
What is the X3D Ontology *not*?

- X3D Ontology is not suitable for scene graph rendering in real time.
- X3D Ontology is not suitable for programmatic model creation.
- Also not suitable for most tasks familiar to 3D graphics modelers.

This is because triple form of an X3D model is instead optimized for search queries and logical inference, using Semantic Web technologies.

Current activity includes the possibility of round-trip conversion, but only as a way to confirm complete mappings of all model information. Equivalent information models are possible, but for different purposes.
Assets available for X3D Ontology testing

- **X3dOntology4.0.ttl** is our primary work.
- **OWLDoc documentation** is generated via **Protégé** ontology editor.
- **X3dToTurtle.xslt** conversion stylesheet for converting an .x3d model from XML into .ttl triples.
- **X3dSemanticWebDevelopmentArchive.zip** collects current assets for easier download and use.
- Build process: **build.xml** uses **Apache Ant** with associated output logs: **build.all.log.txt** and **build.SPARQL.query.tests.log.txt**

This Web3D work is **Open Source** and maintained in **version control**.
X3D Ontology Production Chain

Creation of the X3D Ontology is thorough and autogenerated for each version of X3D, based on **X3D Unified Object Model (X3DUOM)**
XML Schema decorated with full X3DUOM annotations
X3DUOM in Netbeans
X3duomToX3dOntology.xlsl stylesheet
X3d Ontology source
X3D Ontology excerpt

```
:WorldInfo a owl:Class ;
    rdfs:subClassOf :X3DInfoNode ;
    rdfs:label "WorldInfo contains a title and simple persistent metadata information about an X3D scene. This node is strictly for documentation purposes and has no effect on the visual appearance or behaviour of the world." .

:info a owl:DatatypeProperty ;
    rdfs:domain :WorldInfo ;
    rdfs:range :MFString .
:title a owl:DatatypeProperty ;
    rdfs:domain :WorldInfo ;
    rdfs:range :SFString .

# DEF field inheritedFrom=X3DNode with accessType=inputOutput, type=SFString, baseType=xs:ID
# USE field inheritedFrom=X3DNode with accessType=inputOutput, type=SFString, baseType=xs:IDREF
# class field inheritedFrom=X3DNode with accessType=inputOutput, type=SFString, baseType=xs:NMTOKENS
# IS field inheritedFrom=X3DNode with accessType=inputOutput, type=SFNode, default=NULL
# metadata field inheritedFrom=X3DNode with accessType=inputOutput, type=SFNode, default=NULL
```
X3D Ontology in Protégé
X3D model representation in Turtle .ttl form

- Goal: design X3D Ontology to enable precise, in-depth SPARQL queries
- Unique naming pattern is necessary for each X3D node in a model:
  - DEF names, if available, are prefixed by a colon. Example: :MySpecialMaterial.
  - USE names, if available, are adapted by prefixing with a colon and appending -USE- and the number of current occurrence. Example: :MySpecialMaterial-USE-4.
  - X3D, head and Scene are always singletons and so are eponymous and always get the same name, e.g. :X3D, :head and :Scene.
  - Unnamed X3D nodes and statements are identified with the following pattern: node name, followed by tree hierarchy below the root X3D node.
    - Example labels: :meta_1_5 had 1 for head and 5 for fifth child; :Shape_2_3_2_1 has initial 2 for Scene followed by second/second/first children nodes, and so on.
- Automate conversion .x3d to .ttl using X3dToTurtle.xslt stylesheet
- Build thousands of models from X3D Examples Archives and test them!
Full set of examples

• Several thousand XML-based .x3d models in the X3D Example Archives have been converted into .ttl Turtle triples, using the X3dToTurtle.xslt conversion stylesheet.

• Both .ttl and SPARQL query results are available for each model, as illustrated in the screenshot for the HelloWorld example catalog entry.

• Comprehensive conversion and test-query results for all models in X3D Resources, Examples: Scene Archives for X3D are listed in build.turtle.all.log.txt. This is a comprehensive set of unit tests for X3D ontology, xslt model conversion into .ttl, and SPARQL query.
Example model in X3D Examples Archive: Hello World

X3D Example Archives: X3D for Web Authors, Chapter 01 Technical Overview, Hello World

Simple X3D scene example: Hello World!

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.3//EN" "http://www.web3d.org/specifications/x3d-3.3.dtd">
<X3D profile='immersive' version='3.3' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.3.xsd'>
  <head>
    <meta name='title' content='HelloWorld.x3d'/>
    <meta name='description' content='Simple X3D scene example: Hello World!'/>
    <meta name='created' content='30 October 2000'/>
    <meta name='modified' content='28 July 2015'/>
    <meta name='creator' content='Don Brutzman'/>
    <meta name='Image' content='HelloWorld.tall.png'/>
    <meta name='reference' content='http://en.wikipedia.org/wiki/Hello_world'/>
  </head>
</X3D>
```
Example excerpt, **HelloWorld.ttl**  
(converted from **HelloWorld.x3d**)

```turtle
:Shape_2_3_2_1 a owl:NamedIndividual, x3do:Shape ;
  x3do:hasParent :TestWhitespaceCommas ;
  x3do:hasGeometry :Sphere_2_3_2_1_1 ;
  x3do:hasAppearance :Appearance_2_3_2_1_2 .

:Sphere_2_3_2_1_1 a owl:NamedIndividual, x3do:Sphere ;
  x3do:hasParent :Shape_2_3_2_1 .

:Appearance_2_3_2_1_2 a owl:NamedIndividual, x3do:Appearance ;
  x3do:hasParent :Shape_2_3_2_1 ;
  x3do:hasMaterial :MaterialLightBlue ;
  x3do:hasTexture :ImageCloudlessEarth .

:MaterialLightBlue a owl:NamedIndividual, x3do:Material ;
  x3do:hasParent :Appearance_2_3_2_1_2 ;
  x3do:diffuseColor ( 0.1 0.5 1 ) .

:ImageCloudlessEarth a owl:NamedIndividual, x3do:ImageTexture ;
  x3do:hasParent :Appearance_2_3_2_1_2 ;
  x3do:url "earth-topo.png" "earth-topo.jpg" "earth-topo-small.gif" .
```
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#>
PREFIX x3d: <http://www.web3d.org/specifications/x3d-4.0.xsd#>
PREFIX x3do: <http://www.web3d.org/semantics/ontologies/X3dOntology4.0#>

# X3dHelloWorldQuery_03.rq    Query HelloWorld.ttl to show title given to model within a WorldInfo node.

####################################################
WHERE
{
  ?WorldInfo rdf:type x3do:WorldInfo ;
  x3do:title       ?title      ;
  x3do:hasParent   ?parent    .

  BIND (strafter(xsd:string(?WorldInfo),"#") AS ?WorldInfoNode)
  BIND (strafter(xsd:string(?parent),"#") AS ?parentNode)
}
####################################################
### Query response: X3dHelloWorldQuery_03.rq.txt

```sql
WHERE
{
?WorldInfo rdf:type x3do:WorldInfo ;
x3do:title     ?title ;
x3do:hasParent ?parent .

BIND (strafter(xsd:string(?WorldInfo),"#") AS ?WorldInfoNode)
BIND (strafter(xsd:string(?parent),"#")  AS ?parentNode)
}
```

<table>
<thead>
<tr>
<th>WorldInfoNode</th>
<th>title</th>
<th>parentNode</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;WorldInfo_2_1&quot;</td>
<td>&quot;Hello World!&quot;</td>
<td>&quot;Scene&quot;</td>
</tr>
</tbody>
</table>
Future Work

• Many more SPARQL queries! For example, in recent presentations.
  • TODO: query to check all animation chains (source/ROUTE/destination nodes)
• Capturing, mapping metadata vocabularies within X3D models
• Investigate existing vocabularies and tools for visual 3D shape search including potential analysis and annotation of models
• Define similar/corresponding triple constructs for different 3D formats, building upon Leslie's 3dmo.ttl exemplars.
• Experiment with Data Format Description Language (DFDL) using Apache Daffodil as a way to actually parse and subsequently query those other formats coherently.
Recommended Reading: Semantic Web, X3D


• Don Brutzman and Leonard Daly, *X3D for Web Authors*, Morgan, 2007.


• Wikibooks *SPARQL* edition covers both language and also Web-based GUI: Wikidata Query Service SPARQL endpoint.
Contact

Don Brutzman, Ph.D.

brutzman@nps.edu
http://faculty.nps.edu/brutzman

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA