X3D Ontology for Semantic Web

X3D Semantic Web Working Group

Don Brutzman and Jakub Flotyński

brutzman@nps.edu flotynski@kti.ue.poznan.pl

29 July 2020

Semantic Web

- W3C is helping to build a technology stack to support a “Web of data” supported by Semantic Web standards.

- Wikipedia: “The **Semantic Web** is an extension of the World Wide Web through standards set by the World Wide Web Consortium (W3C). The goal of the Semantic Web is to make Internet data machine-readable. To enable the encoding of semantics with the data, technologies such as Resource Description Framework (RDF) and Web Ontology Language (OWL) are used. These technologies are used to formally represent metadata. For example, ontology can describe concepts, relationships between entities, and categories of things. These embedded semantics offer significant advantages such as reasoning over data and operating with heterogeneous data sources.”

X3D Semantic Web Working Group 1

- 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, 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.

X3D Semantic Web Working Group 2

Goals

- Enable more effective indexing, search, comparison, and analysis of X3D models through the advanced use of metadata, shape geometry, etc.
- Create, 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 in multiple formats: Terse RDF Triple Language (Turtle), RDF/OWL, (TODO) JSON-LD.

X3D Ontology for Semantic Web

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

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.

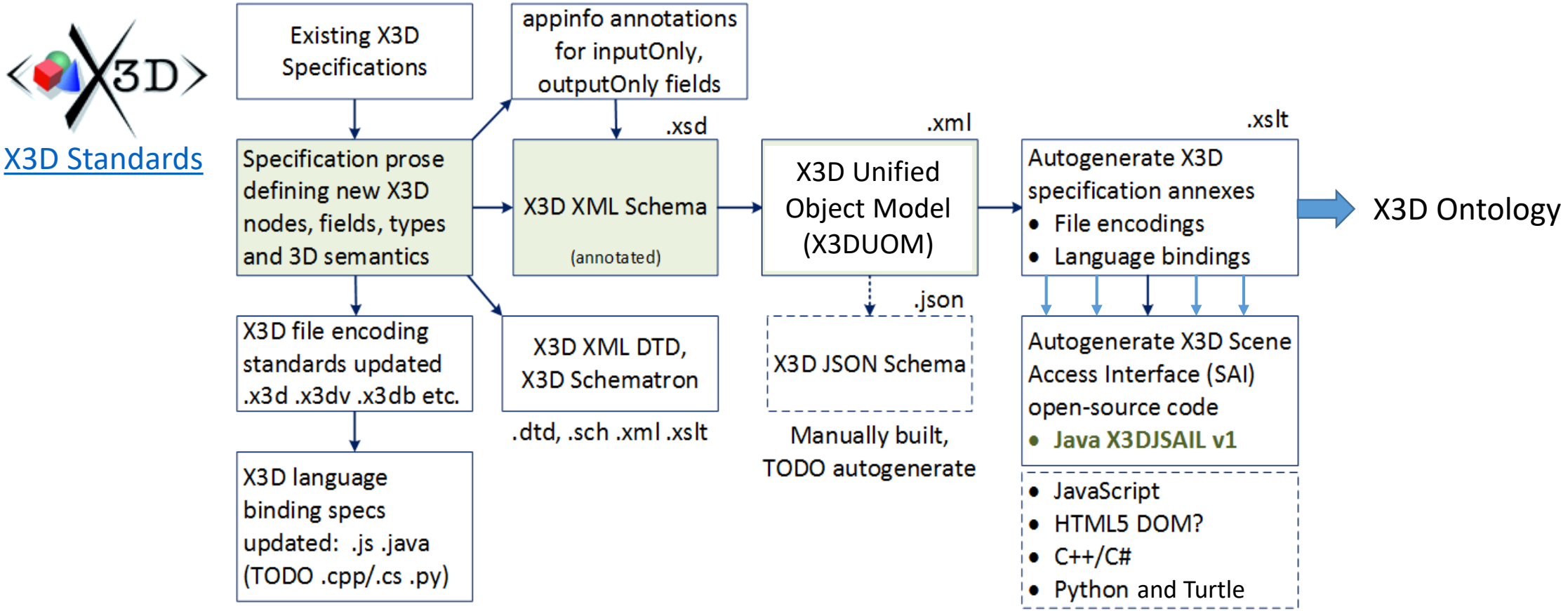
Assets available for X3D Ontology testing

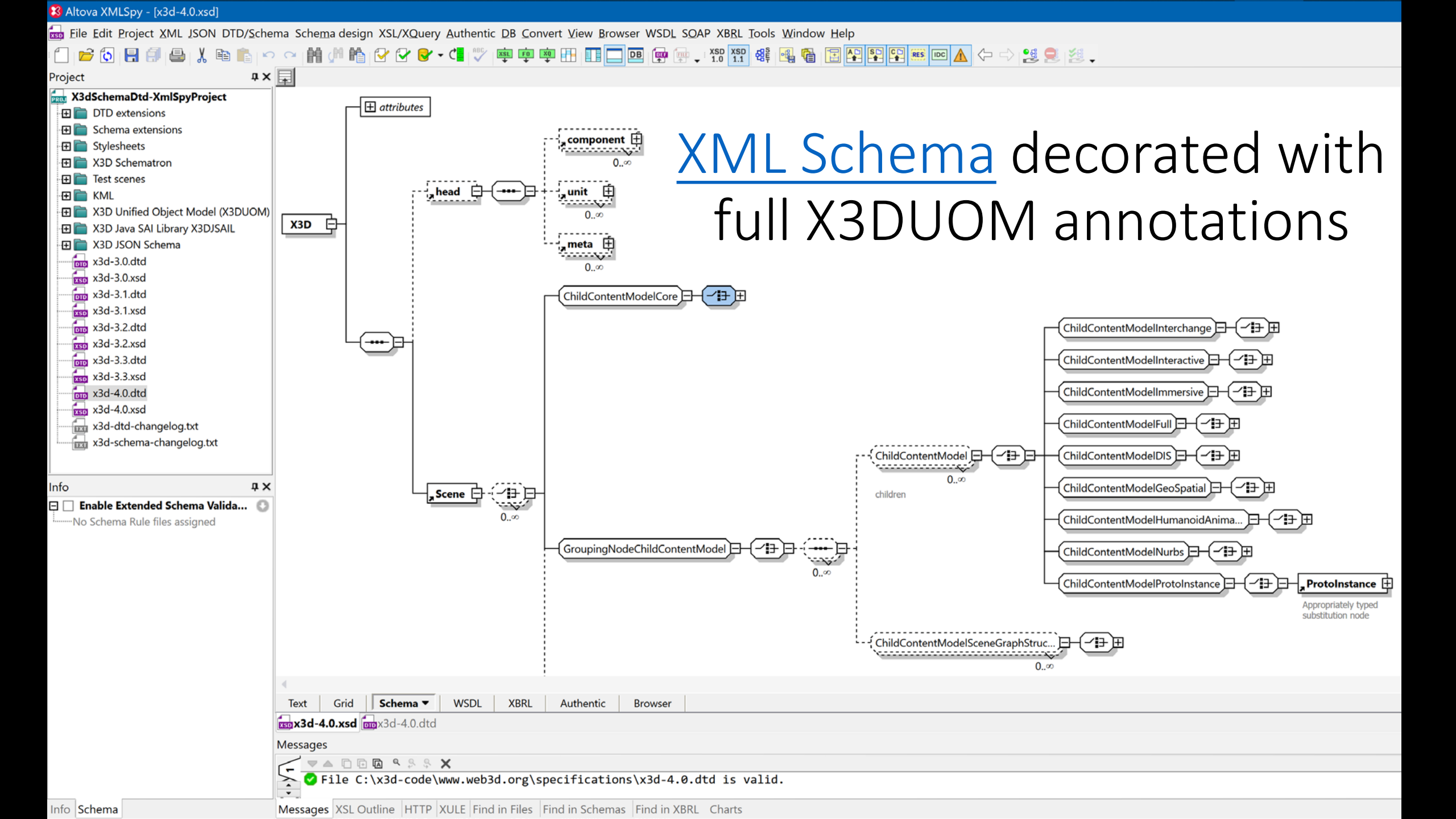
- X3dOntology4.0.ttl is our primary work.
- OWLDoc documentation is generated via Protégé ontology editor.
- X3dToTurtle.xslt conversion stylesheet for converting an .x3d model from XML into .ttl triples.
- X3dSemanticWebDevelopmentArchive.zip collects current assets for easier download and use.
- Build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

This Web3D work is Open Source and maintained in version control.

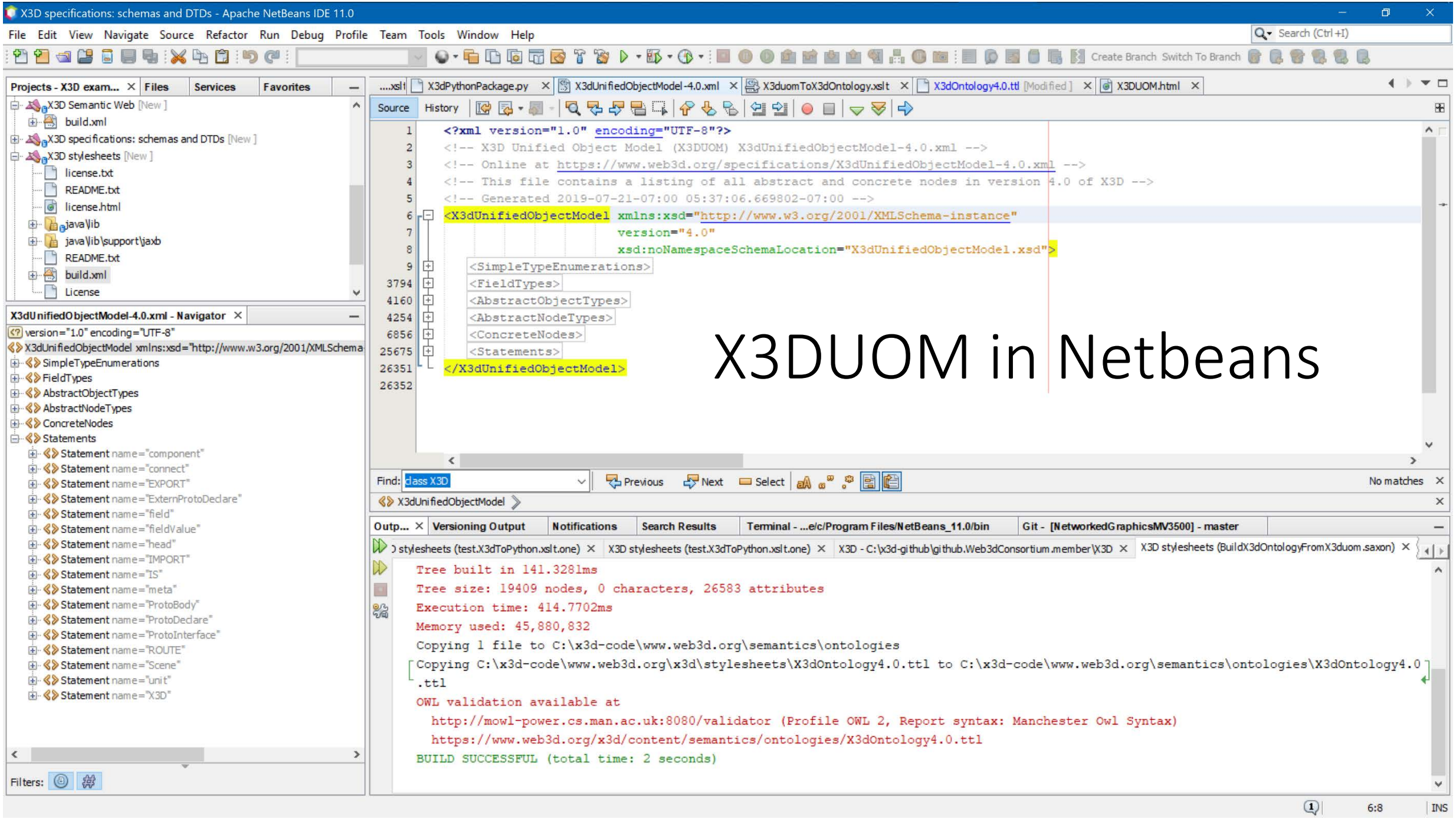
X3D Ontology Production Chain

Creation of the X3D Ontology is thorough and autogenerated for each version of X3D, based on **X3D Unified Object Model (X3DUOM)**

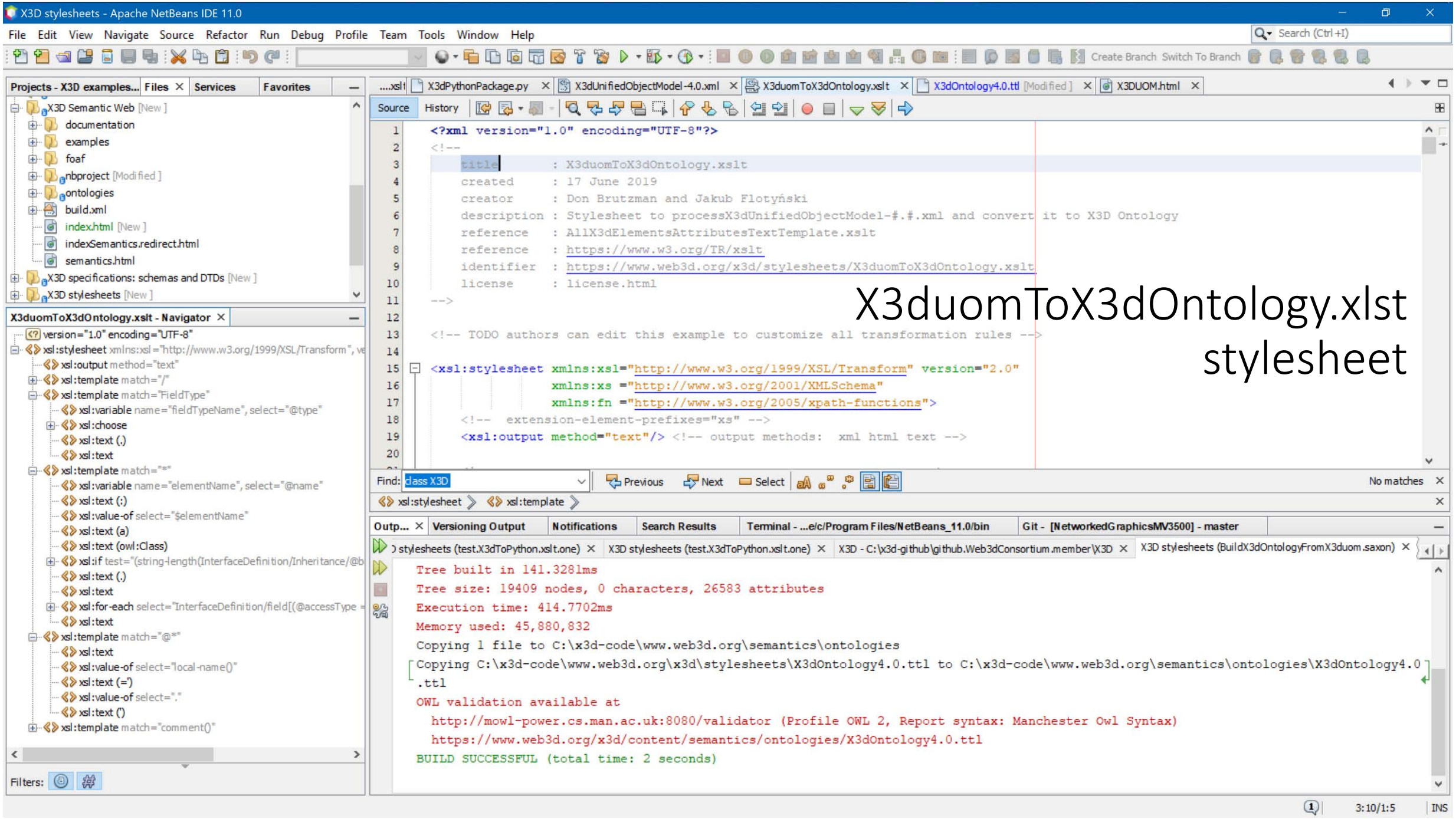


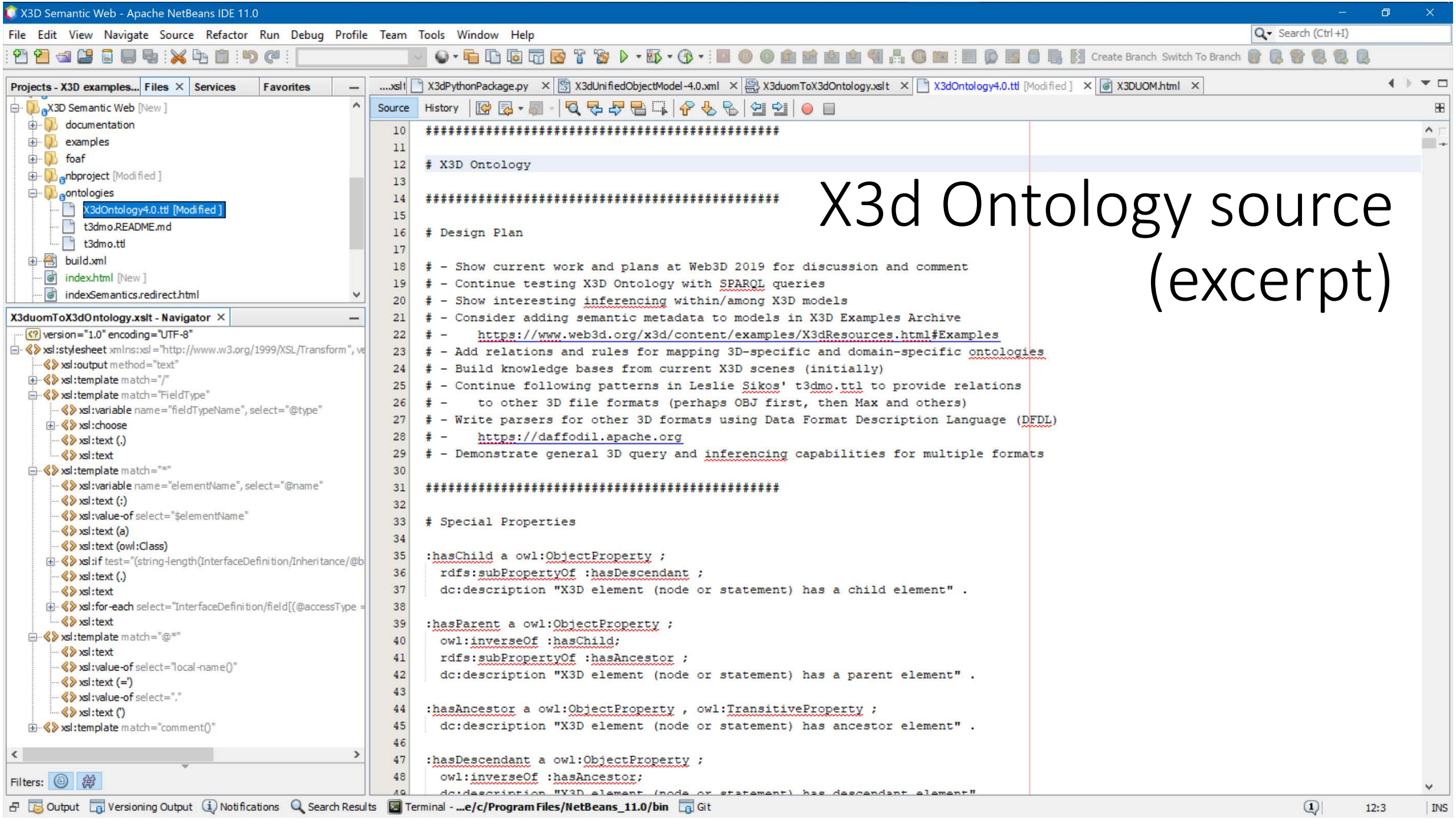


XML Schema decorated with full X3DUOM annotations



X3DUOM in Netbeans


X3D stylesheets - Apache NetBeans IDE 11.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects - X3D examples... Files Services Favorites
X3D Semantic Web [New]
documentation
examples
foaf
nbproject [Modified]
ontologies
build.xml
index.html [New]
indexSemantics.redirect.html
semantics.html
X3D specifications: schemas and DTDs [New]
X3D stylesheets [New]
X3duomToX3dOntology.xslt - Navigator
version="1.0" encoding="UTF-8"
xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform", ve
xsl:output method="text"
xsl:template match="/"
xsl:template match="FieldType"
xsl:variable name="fieldTypeName", select="@type"
xsl:choose
xsl:text (,)
xsl:text
xsl:template match="*"
xsl:variable name="elementName", select="@name"
xsl:text (:)
xsl:value-of select="$elementName"
xsl:text (a)
xsl:text (owl:Class)
xsl:if test="(string-length(InterfaceDefinition/Inheritance/@b
xsl:text (,)
xsl:text
xsl:for-each select="InterfaceDefinition/field[(@accessType =
xsl:text
xsl:template match="@*"
xsl:text
xsl:value-of select="local-name()"
xsl:text (=')
xsl:value-of select="."
xsl:text (')
xsl:template match="comment()"
Filters:
....xslt X3dPythonPackage.py X3dUnifiedObjectModel-4.0.xml X3duomToX3dOntology.xslt X3dOntology4.0.ttl [Modified] X3DUOM.html
Source History
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 title : X3duomToX3dOntology.xslt
4 created : 17 June 2019
5 creator : Don Brutzman and Jakub Flotyński
6 description : Stylesheet to processX3dUnifiedObjectModel-#.#.xml and convert it to X3D Ontology
7 reference : AllX3dElementsAttributesTextTemplate.xslt
8 reference : https://www.w3.org/TR/xslt
9 identifier : https://www.web3d.org/x3d/stylesheets/X3duomToX3dOntology.xslt
10 license : license.html
11 -->
12
13 <!-- TODO authors can edit this example to customize all transformation rules -->
14
15 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
16 xmlns:xs ="http://www.w3.org/2001/XMLSchema"
17 xmlns:fn ="http://www.w3.org/2005/xpath-functions">
18 <!-- extension-element-prefixes="xs" -->
19 <xsl:output method="text"/> <!-- output methods: xml html text -->
20
Find: class X3D
Previous Next Select
No matches
xsl:stylesheet xsl:template
X3duomToX3dOntology.xslt stylesheet
Outp... Versioning Output Notifications Search Results Terminal - ...e/c/Program Files/NetBeans_11.0/bin Git - [NetworkedGraphicsMV3500] - master
stylesheets (test.X3dToPython.xslt.one) X3D stylesheets (test.X3dToPython.xslt.one) X3D - C:\x3d-github\github.Web3dConsortium.member\X3D X3D stylesheets (BuildX3dOntologyFromX3duom.saxon)
Tree built in 141.328lms
Tree size: 19409 nodes, 0 characters, 26583 attributes
Execution time: 414.7702ms
Memory used: 45,880,832
Copying 1 file to C:\x3d-code\www.web3d.org\semantics\ontologies
Copying C:\x3d-code\www.web3d.org\x3d\stylesheets\X3dOntology4.0.ttl to C:\x3d-code\www.web3d.org\semantics\ontologies\X3dOntology4.0.ttl
OWL validation available at
http://mowl-power.cs.man.ac.uk:8080/validator (Profile OWL 2, Report syntax: Manchester Owl Syntax)
https://www.web3d.org/x3d/content/semantics/ontologies/X3dOntology4.0.ttl
BUILD SUCCESSFUL (total time: 2 seconds)
3:10/1:5
INS



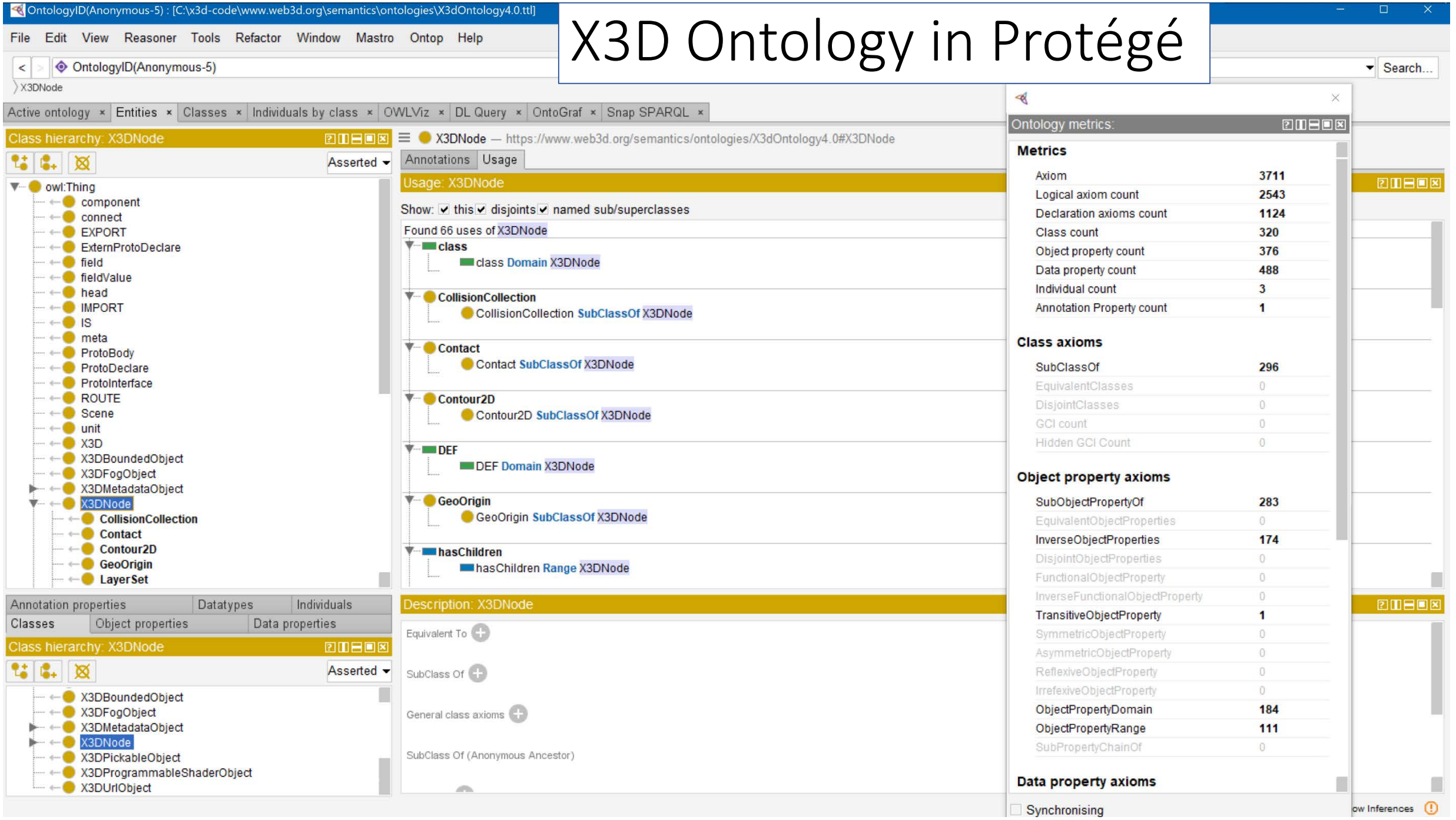
X3d Ontology source (excerpt)

X3D Ontology excerpt

Full set of data typing and scene graph parent-child relationships available.

Enables precise query of any X3D model.

```
: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: broadly apply 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
- Convert all models and metadata in X3D Examples Archives for testing!

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

https://x3dgraphics.com/examples/X3dForWebAuthors/Chapter 120%

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

Simple X3D scene example: Hello World!

X3D model | X_ITE
ClassicVRML | X3DOM
VRML97 | .json (check)
Canonical XML | .x3db Binary
annotated documentation | .java source (Javadoc)
.py python | .ttl Turtle (query)

```
<?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 '/>
```

Example excerpt, HelloWorld.ttl (converted from HelloWorld.x3d)

```
: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" ' .
```

SPARQL Query: X3dHelloWorldQuery_03.rq

```
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.

################################################

SELECT ?WorldInfoNode ?title ?parentNode
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

```
##############################################
SELECT ?WorldInfoNode ?title ?parentNode
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)
}
##############################################
```

SPARQL query

```
-------------------------------------------------
| WorldInfoNode   | title           | parentNode |
=================================================
| "WorldInfo_2_1" | "Hello World!" | "Scene"    |
-------------------------------------------------
```

Simple results

Asking, answering more-difficult 3D questions

Quality Assurance (QA) of sophisticated behavior is challenging. Can we query?

- Example: do connections exist for intended user interactions, model animation?
- Can we automate such error-detection queries for massive X3D model archives?

parentNode	RouteFound	fromNodeDEF	fromNodeTypeFound	fromField	toNodeDEF	toField
"Scene"	"ROUTE_2_5"	"OrbitalTimeInterval"	x3do:TimeSensor	"fraction_changed"	"SpinThoseThings"	"set_fraction"
"EarthCoordinateSystem"	"ROUTE_2_6_1"	"SpinThoseThings"	x3do:OrientationInterpolator	"value_changed"	"EarthCoordinateSystem"	"set_rotation"
"EarthCoordinateSystem"	"ROUTE_2_6_5"	"ClickTriggerTouchSensor"	x3do:TouchSensor	"touchTime"	"OrbitalTimeInterval"	"startTime"

- Complementary compatible domains: 3D printing and 3D scanning
- Future queries: document metadata, embedded-in-model metadata, topology and geometric structure, size and shape relationships, intended purpose, etc.
- Future horizon: Hybrid queries across linked models in multiple 3D formats

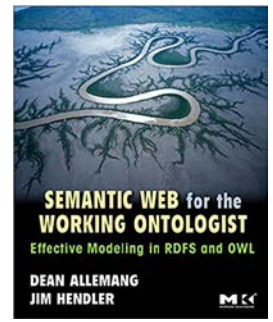
Future Work Agenda

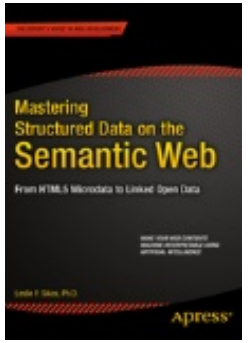
X3D Semantic Web Working Group
is essential forum for group progress

- Many more SPARQL queries! *If we can precisely state it, we can search for it…*
  - Current: query to check all animation chains (source/ROUTE/destination nodes)
- Capturing, mapping metadata vocabularies within X3D models
- Rigorous geometric terms: ISO draft *Ontology for Geometry and Topology*
- Investigate existing vocabularies and tools for visual 3D shape search including potential analysis and annotation of models. Can we align?
- Define similar/corresponding triple constructs for different 3D formats, building upon design patterns in Leslie Sikos' *3dmo.ttl* exemplars.
- Experiment with Data Format Description Language (DFDL) 2-way mappings using Apache Daffodil as a way to actually parse and subsequently perform queries on various other 3D formats coherently and comprehensively.

Recommended Reading: Semantic Web, X3D

- Dean Allemang and Jim Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, second edition, 2012.
- Don Brutzman and Leonard Daly, X3D for Web Authors, Morgan, 2007.
- Bob DuCharme, Learning SPARQL, second edition, 2013.
- Leslie F. Sikos, Mastering Structured Data on the Semantic Web: from HTML5 Microdata to Linked Open Data, 2015.
- Wikibooks SPARQL edition covers both language and also Web-based GUI: Wikidata Query Service SPARQL endpoint.





Contact

Don Brutzman, Ph.D.

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

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

Contact

Jakub Flotyński, Ph.D.

flotynski@kti.ue.poznan.pl
https://www.kti.ue.poznan.pl/en/jakub_flotynski.html

Department of Information Technology
Poznan University of Economics and Business
Poznan Poland