Semantic Web3D: Towards Comprehensive Representation of 3D Content on the Semantic Web

Part 2: Query Examples

International Conference on 3D Immersion (IC3D)
in Brussels Belgium, 11-12 December 2019.

Jakub Flotyński, Don Brutzman, Felix G. Hamza-Lup, Athanasios Malamos, Nicholas Polys, Leslie F. Sikos, and Krzysztof Walczak

semantics-public@web3d.org

Presentation Outline

- X3D Semantic Web Working Group
- Motivations for Semantic 3D content
- The Semantic Web3D Approach
- X3D Ontology and Knowledge Bases
- Examples
- Conclusions and Future Work

3D/VR/AR on the Web

- Integration of 3D/VR/AR with Web browsers
- Wide accessibility and collaborative environments
- Content formats, e.g., Extensible 3D (X3D)
- Programming libraries, e.g. WebGL
- Interfaces, e.g. WebXR

The Semantic Web

- Global database linking structured content with semantic descriptions
- Ontologies and knowledge bases
- Applicable to any domain
- Enables
  - Content description at arbitrary specification level
  - Reasoning
  - Queries
- W3C Standards: RDF, RDFS, OWL, SPARQL
- No integration between 3D/VR/AR and the Semantic Web

X3D Semantic Web Working Group

- Maximize interoperability with Semantic Web standards for greatest possible reuse and integration of 3D with the web
- Efficient indexing, search, comparison, and analysis of X3D models through the advanced use of metadata and semantics
- Create, partially autogenerate X3Dv4 OWL Ontology from the X3D Unified Object Model (X3DUOM) using best-practice design patterns
- 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 and maintain a list of domain-specific ontologies that are suitable for use in concert with the X3D Ontology.

Motivations for Semantic 3D Content

- Compliant with current Web evolution (Semantic Web)
- Facilitates
  - Management (indexing, searching)
  - Exploration (reasoning, queries)
  - Modeling (non-IT-specialists)

  of 3D content
- Independent of particular 3D formats and presentation platforms
- Different levels of specificity (3D and application/domain)
- Declarative content representation

Example Semantics of 3D Content

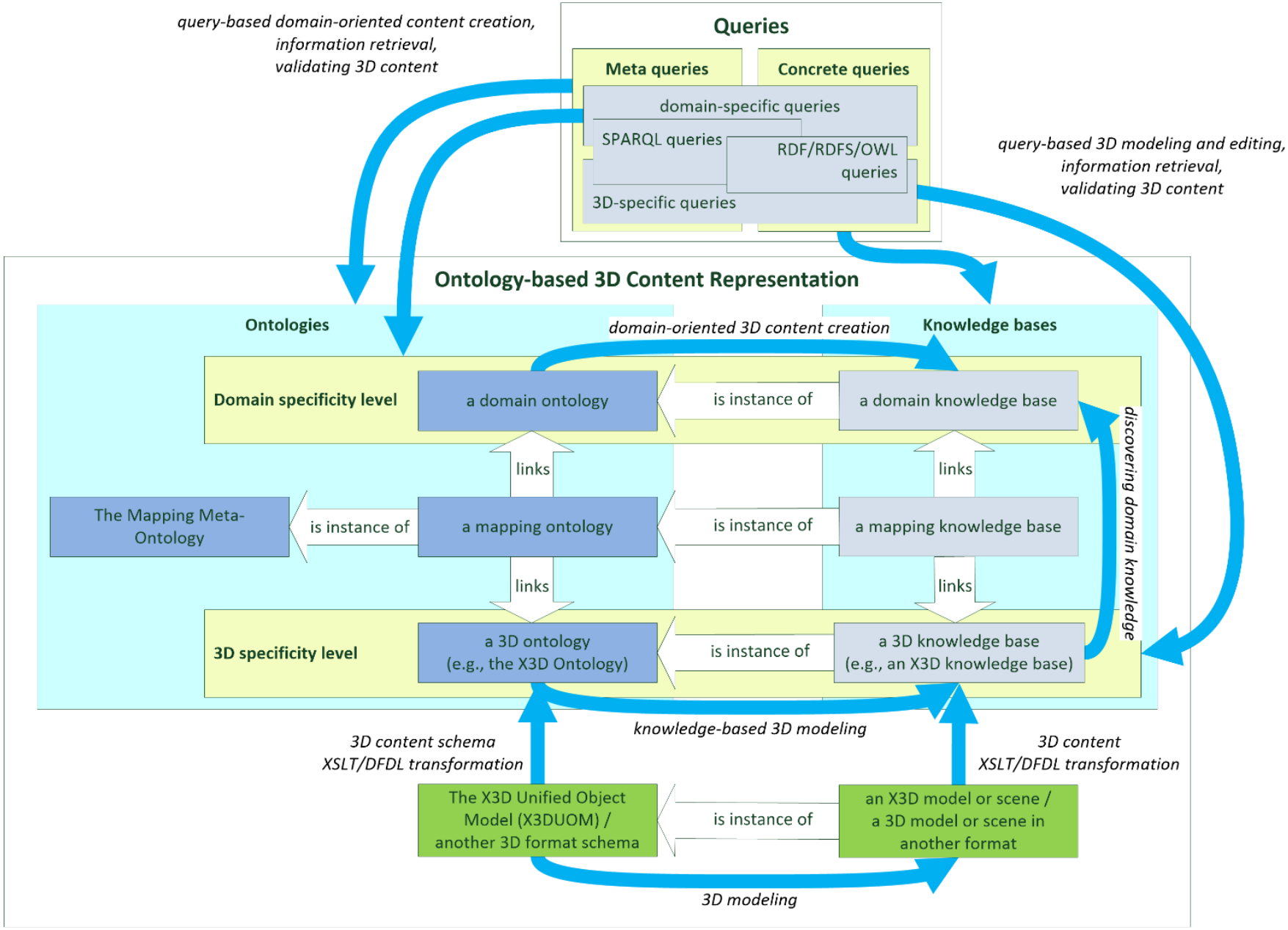
- Semantic description of 3D scene enables answers to semantic reasoning and queries about it
- Reasoning and queries may cover properties of 3D objects
  - At both 3D and domain levels of specificity
  - Related to different content features
    - Geometry, e.g.,
      - What is the type of a shape? (3D-specific)
      - What is the category of a car based on its shape? (domain-specific)
    - Structure, e.g.,
      - How many polygons does a 3D model have? (3D-specific)
      - What are components of a virtual car? (domain-specific)
    - Presentation, e.g.,
      - Which objects in a scene use a common texture? (3D-specific)
      - Which objects in a scene are made of wood? (domain-specific)
    - Behavior, e.g.,
      - What scripts describe the behavior of an object? (3D-specific)
      - What is the exercise performed by an avatar? (domain-specific)
- Combining specificity levels by ontology mapping, e.g., virtual museum ontology to 3D ontology

The Semantic Web3D

- Current efforts of the working group
- Application of the semantic web to 3D technologies intended to support:
  - Development,
  - Management, and
  - Usage

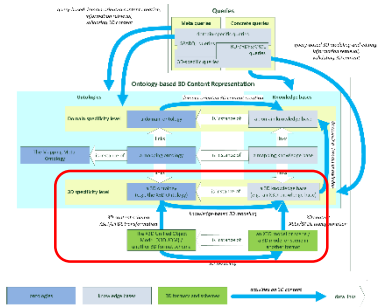
… of 3D content on the web

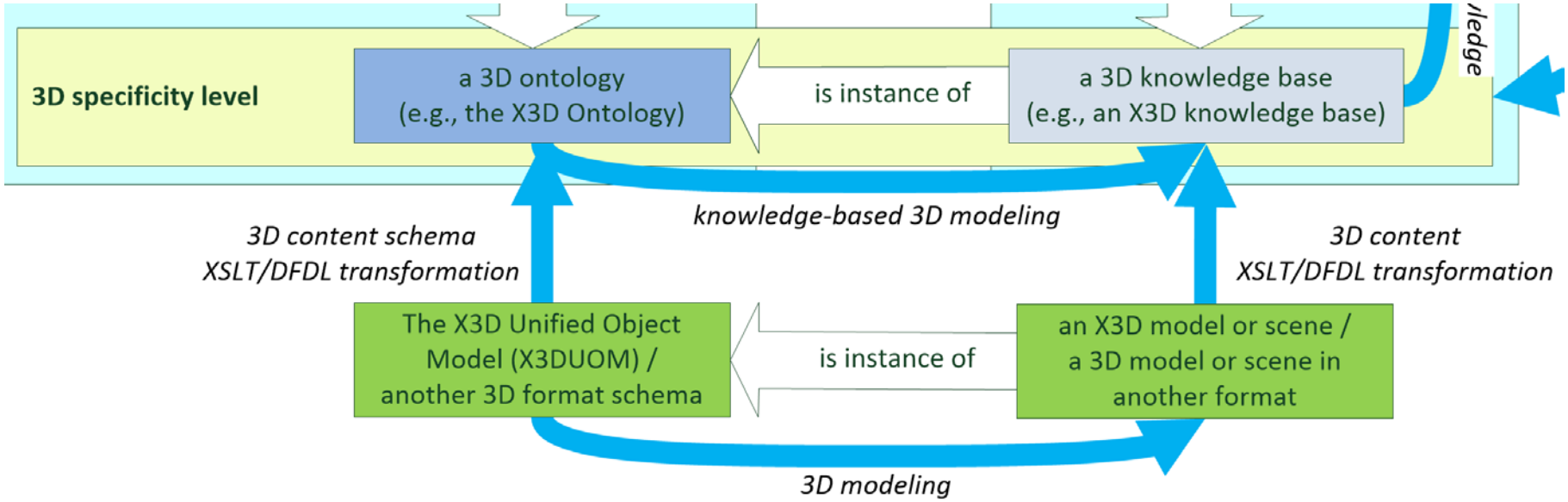
Semantic Web3D



ontologies | knowledge bases | 3D formats and schemes | activities on 3D content | data links

3D Content and Transformations





X3D Ontology

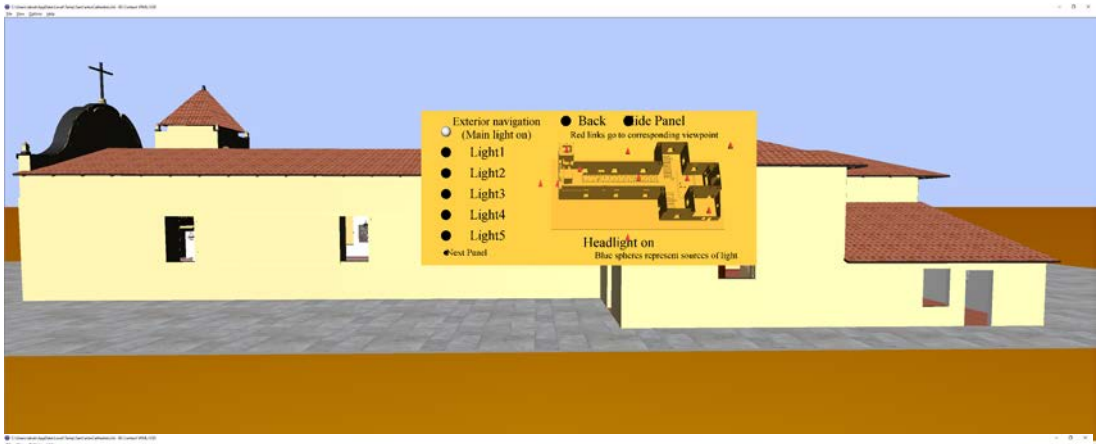
- Semantic representation of the X3D format
- Automatically generated from the X3D Unified Object Model (X3DUOM) via XSLT
- Encoded in RDF, RDFS, OWL
- Queryable with SPARQL
- Enables reasoning

Advantages

- Up-to-date representation of various 3D features
  - Geometry
  - Structure
  - Presentation
  - Animation
- Automatic generation of semantic X3D repositories based on already-available 3D models
- Reasoning and querying over transcribed versions of available X3D content

Example: San Carlos Cathedral

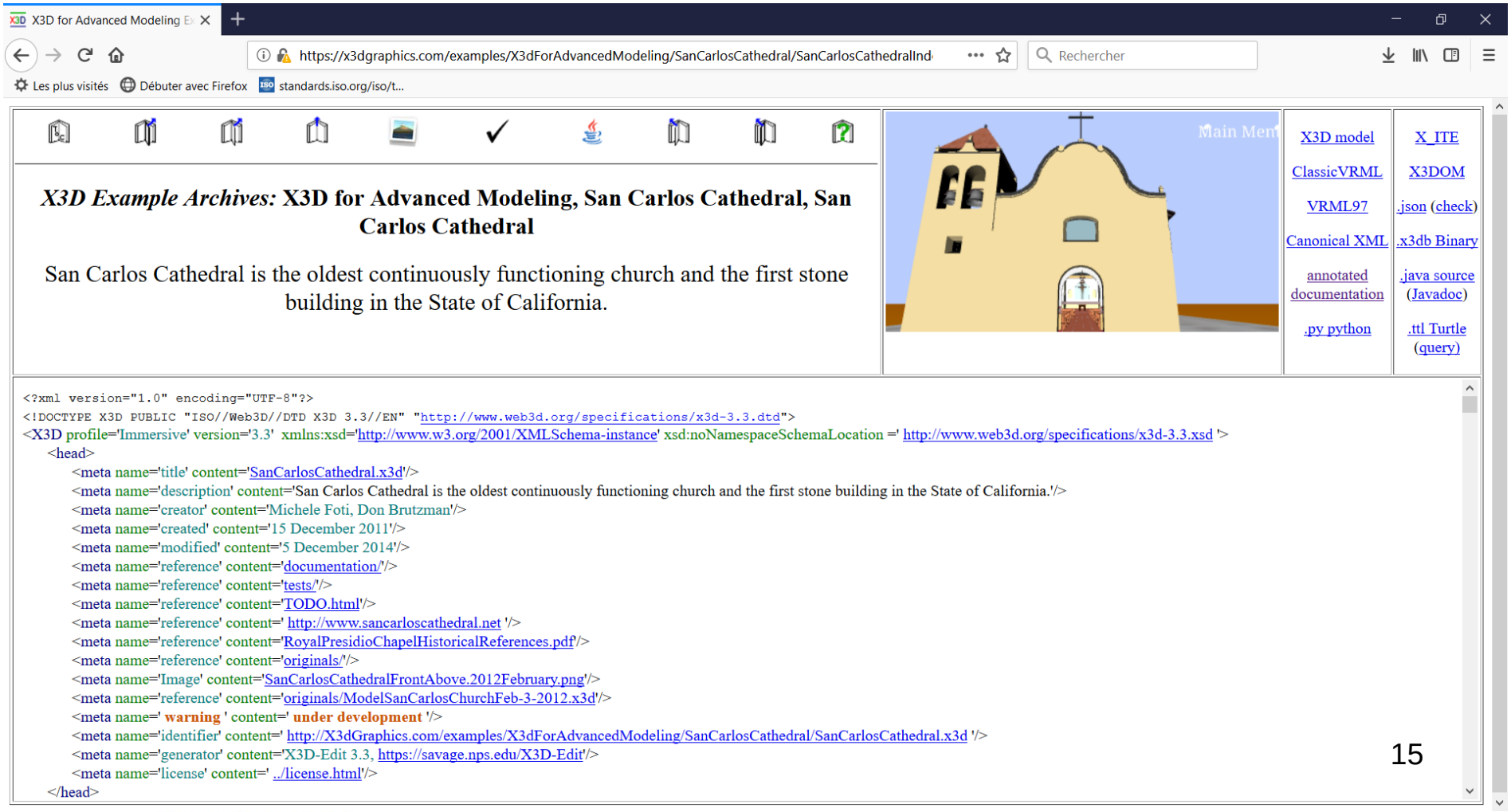
https://x3dgraphics.com/examples/X3dForAdvancedModeling/SanCarlosCathedral



https://upload.wikimedia.org/wikipedia/commons/c/c1/Monterey%2C_California_-_Cathedral_of_San_Carlos_Borromeo_%28Royal_Presidio_Chapel%29_-_panoramio.jpg

Example: San Carlos Cathedral

https://x3dgraphics.com/examples/X3dForAdvancedModeling/SanCarlosCathedral



Cathedral: semantic representation

```
1  # Prefixes: 'x3do', ':', 'rdf' and 'owl' indicate: the X3D
       Ontology and knowledge base as well as RDF and OWL.
2
3  :scene rdf:type owl:NamedIndividual , x3do:Scene .
4  :scene x3do:hasBackground :background .
5  :background rdf:type owl:NamedIndividual, x3do:Background;
6    x3do:skyColor (0.7216 0.8 0.9922).
7  :scene x3do:hasTransform :Colonna1 .
8  :Colonna1 rdf:type owl:NamedIndividual , x3do:Transform ;
9    x3do:translation (0.7 0 -0.7) .
10 :Colonna1 x3do:hasShape :woodenElement1 .
11 :woodenElement1 rdf:type owl:NamedIndividual , x3do:Shape.
12 :woodenElement1 x3do:hasBox :woodenElement1Box .
13 :woodenElement1Box rdf:type owl:NamedIndividual, x3do:Box;
14   x3do:size (0.4 1.2 0.4) .
15 :woodenElement1 x3do:hasAppearance :WoodAppearance .
16 :WoodAppearance rdf:type owl:NamedIndividual , x3do:
       Appearance .
17 :WoodAppearance x3do:hasTexture :Wood .
18 :Wood rdf:type owl:NamedIndividual , x3do:ImageTexture ;
       x3do:url ".../Wood.jpg" .
```

web3D CONSORTIUM

Turtle (.ttl)

WorldInfo Query

X3D 3D ANYWHERE

Cathedral: SPARQL semantic query 1

```
1  # Prefixes: 'x3do', ':', 'rdf' and 'owl' indicate: the X3D
       Ontology and knowledge base as well as RDF and OWL.
2
3  :scene rdf:type owl:NamedIndividual , x3do:Scene .
4  :scene x3do:hasBackground :background .
5  :background rdf:type owl:NamedIndividual, x3do:Background;
6    x3do:skyColor (0.7216 0.8 0.9922).
7  :scene x3do:hasTransform :Colonna1 .
8  :Colonna1 rdf:type owl:NamedIndividual , x3do:Transform ;
9    x3do:translation (0.7 0 -0.7) .
10 :Colonna1 x3do:hasShape :woodenElement1 .
11 :woodenElement1 rdf:type owl:NamedIndividual , x3do:Shape.
12 :woodenElement1 x3do:hasBox :woodenElement1Box .
13 :woodenElement1Box rdf:type owl:NamedIndividual, x3do:Box;
14   x3do:size (0.4 1.2 0.4) .
15 :woodenElement1 x3do:hasAppearance :WoodAppearance .
16 :WoodAppearance rdf:type owl
       Appearance
```

How many shapes together compose the altar?

```
SELECT (count(distinct ?shape) as ?num) WHERE {
  ?shape rdf:type x3do:Shape . }
```

Query and result

X3dSanCarlosCathedralAltarQuery_01.rq.txt

```
Perform X3D Ontology query X3dSanCarlosCathedralAltarQuery_01.rq using examples/Altar.ttl to produce
output file X3dSanCarlosCathedralAltarQuery_01.rq.txt:
============================
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#>

# X3dSanCarlosCathedralAltarQuery_01.rq    Query Altar.ttl to count numberShapes

################################################
# Every X3D knowledge base can be subject to semantic queries.

# The following SPARQL query provides the number of shapes composing the altar.
# The result of the query is: 14.

SELECT (count(distinct ?shape) as ?numberShapes)

WHERE
{
    ?shape rdf:type x3do:Shape .
}
################################################
---------------
| numberShapes |
===============
| 14           |
---------------
```

Cathedral: SPARQL semantic query 2

```
1   # Prefixes: 'x3do', ':', 'rdf' and 'owl' indicate: the X3D
        Ontology and knowledge base as well as RDF and OWL.
2
3   :scene rdf:type owl:NamedIndividual , x3do:Scene .
4   :scene x3do:hasBackground :background .
5   :background rdf:type owl:NamedIndividual, x3do:Background;
6     x3do:skyColor (0.7216 0.8 0.9922).
7   :scene x3do:hasTransform :Colonna1 .
8   :Colonna1 rdf:type owl:NamedIndividual , x3do:Transform ;
9     x3do:translation (0.7 0 -0.7) .
10  :Colonna1 x3do:hasShape :woodenElement1 .
11  :woodenElement1 rdf:type owl:NamedIndividual , x3do:Shape.
12  :woodenElement1 x3do:hasBox :wo
13  :woodenElement1Box rdf:type owl
14    x3do:size (0.4 1.2 0.4) .
15  :woodenElement1 x3do:hasAppeara
16  :WoodAppearance rdf:type owl:Na
        Appearance .
17  :WoodAppearance x3do:hasTexture
18  :Wood rdf:type owl:NamedIndividual , x3do:ImageTexture ;
        x3do:url ".../Wood.jpg" .
```

What textures are used for the 3D model?

```
SELECT ?textureUrl WHERE {
  ?x x3do:hasTexture ?texture .
  ?texture x3do:url ?textureUrl . }
ORDER by ASC(?textureUrl)
```

web3D CONSORTIUM

Query and result

X3D 3D ANYWHERE

X3dSanCarlosCathedralAltarQuery_02.rq.txt

```
Perform X3D Ontology query X3dSanCarlosCathedralAltarQuery_02.rq using examples/Altar.ttl to produce
output file X3dSanCarlosCathedralAltarQuery_02.rq.txt:
============================
# (PREFIX headers omitted)

# X3dSanCarlosCathedralAltarQuery_02.rq    Query Altar.ttl for texture url values.
################################################

# Every X3D knowledge base can be subject to semantic queries.

# The following query provides the url addresses of all textures used within the scene.
# The result is the wood texture: ../Wood.jpg (cf. Listing 3, line 18)

SELECT ?appearanceNode ?textureUrl
WHERE
{
    ?appearance x3do:hasTexture ?texture .
    ?texture    x3do:url        ?textureUrl .

    BIND (strafter(xsd:string(?appearance),"#") AS ?appearanceNode)
}
ORDER by ASC(?textureUrl)
################################################

-------------------------------------------------------------------------------------------------------
| appearanceNode   | textureUrl                                                                       |
=======================================================================================================
| "WoodAppearance" | "\"images/Wood.jpg\"
  \“https://x3dgraphics.com/examples/X3dForAdvancedModeling/SanCarlosCathedral/images/Wood.jpg\""      |
-------------------------------------------------------------------------------------------------------
```

Cathedral: SPARQL semantic query 3

```
1  # Prefixes: 'x3do', ':', 'rdf' and 'owl' indicate: the X3D
       Ontology and knowledge base as well as RDF and OWL.
2
3  :scene rdf:type owl:NamedIndividual , x3do:Scene .
4  :scene x3do:hasBackground :background .
5  :background rdf:type owl:NamedIndividual, x3do:Background;
6    x3do:skyColor (0.7216 0.8 0.9922).
7  :scene x3do:hasTransform :Colonna1 .
8  :Colonna1 rdf:type owl:NamedIndividual , x3do:Transform ;
9    x3do:translation (0.7 0 -0.7) .
10 :Colonna1 x3do:hasShape :woodenElement1 .
11 :woodenElement1 rdf:type owl:NamedIndividual , x3do:Shape.
12 :woodenElement1 x3do:hasBox :woodenElement1Box .
13 :woodenElement1Box rdf:type owl:NamedIndividual, x3do:Box;
14   x3do:size (0.4 1.2 0.4) .
15 :woodenElement1 x3do:hasAppearance :WoodAppearance .
16 :WoodAppearance rdf:type owl:NamedIndividual , x3do:
       Appearance .
17 :WoodAppearance x3do:hasTexture :Wood .
18 :Wood rdf:type owl:NamedIndividual , x3do:ImageTexture ;
       x3do:url "../Wood.jpg" .
```

What is the sky color?

Query and result

```
SELECT ?skyColorListVal WHERE {
  ?background rdf:type x3do:Background ;
    x3do:skyColor/rdf:rest*/rdf:first ?skyColorListVal . }
```

web 3D CONSORTIUM

X3dSanCarlosCathedralAltarQuery_03.rq.txt

```
Perform X3D Ontology query X3dSanCarlosCathedralAltarQuery_03.rq using examples/Altar.ttl to produce
output file X3dSanCarlosCathedralAltarQuery_03.rq.txt:
# (PREFIX headers omitted)

# X3dSanCarlosCathedralAltarQuery_03.rq    Query Altar.ttl to determine Background skyColor values.

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

# Every X3D knowledge base can be subject to semantic queries.

# The following query retrieves the Background skyColor used in the scene.
# The result is the following list of RGB values: (0.7216 0.8 0.9922) (cf. Listing 3, line 6).

# Note special handling of RDF lists:
# Bob DuCharme's weblog, 21 April 2014, "RDF lists and SPARQL"
# http://www.snee.com/bobdc.blog/2014/04/rdf-lists-and-sparql.html

SELECT ?backgroundNode ?skyColorListValues
WHERE
{
    ?background rdf:type                          x3do:Background ;
                x3do:skyColor/rdf:rest*/rdf:first ?skyColorListValues .
    BIND (strafter(xsd:string(?background),"#") AS ?backgroundNode)
    # TODO re-aggregate skyColor list values into list of tuples
}
################################################
-----------------------------------------
| backgroundNode   | skyColorListValues |
=========================================
| "Background_2_2" | 0.7216             |
| "Background_2_2" | 0.8                |
| "Background_2_2" | 0.9922             |
-----------------------------------------
```

Conclusions and Future Work

- Advantages of the presented approach
  - Integration of the Semantic Web and 3D
  - Up-to-date with all versions of X3D
  - Automatic generation of ontology eliminates potential errors
  - Queries and reasoning become feasible, consistent
  - Platform-independent
- Future work
  - Integration with achievements of other Web3D Working Groups
  - Integration with metadata and semantics in X3D metadata nodes
  - Mapping to diverse domain ontologies

The **Semantic Web3D**: Towards Comprehensive Representation of 3D Content on the Semantic Web

Thank you for your attention

Jakub Flotyński, Don Brutzman, Felix G. Hamza-Lup,
Athanasios Malamos, Nicholas Polys, Leslie F. Sikos, Krzysztof Walczak
semantics-public@web3d.org