



WEB3D 2015

20th International Conference on 3D Web Technology

June 18-21 –Heraklion, Crete, Greece

# COMBINING X3D WITH SEMANTIC WEB TECHNOLOGIES FOR INTERIOR DESIGN

*Konstantinos Kontakis, Malvina Steiakaki, Michael Kalochristianakis,  
Kostas Kapetanakis and Athanasios G. Malamos*



**TEI of Crete**

Technological Educational Institute of Crete





2015

# Acknowledgements



This work was funded by the Operational Program "Education and Lifelong Learning", Archimedes III, project "DECO-DECoration Ontology. Ontology and semantic search applications for support in interior architecture and decoration designs", code 32, financed by the EU Social Fund, 2007-2013.



**European Union**  
European Social Fund



MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS  
M A N A G I N G   A U T H O R I T Y

**Co-financed by Greece and the European Union**





2015

# Introduction



The vigorous advances in Virtual Reality during the last decade made possible its application in various domains, from health care and gaming industry to industrial use cases and interior decoration. In this work, it is presented DEC-O (DECoration-Ontology), a system which merges Web3D technologies, Semantic Web standards for the creation of customizable virtual interior spaces.

The ontological framework DEC-O consists of the following components:



2015

# Web-based Ontology Editor (1/2)



The ***Web-based Ontology Editor*** which is responsible for populating and manipulating the records in OWL knowledge base. Such records represent the most widely used objects in a typical room-space, along with their qualitative and quantitative characteristics. The objects in DEC-O are defined by various subclasses of four major classes which are the “*Content*”, “*DataTypesCandidates*”, “*Room*” and “*Structural*”.



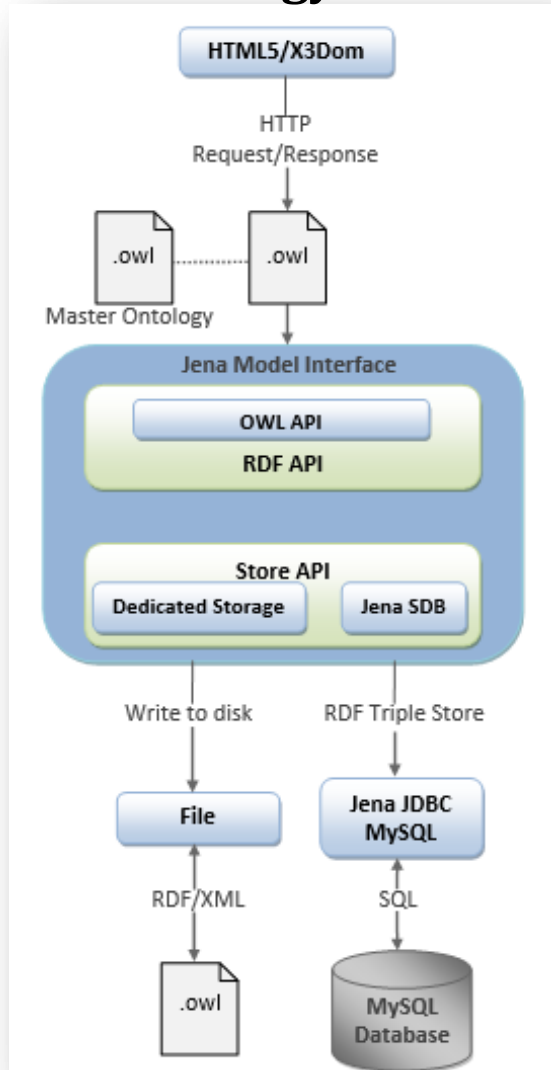


2015

# Web-based Ontology Editor (2/2)



## Web-based Ontology Editor – Back end



## Web-based Ontology Editor – Front end

The front end consists of several interactive panels:

- Create your interior space**: A panel titled "What would you like to annotate?" with a dropdown menu for **DataTypesCandidates** (options: Room, Content, Structural) and an **OK** button.
- Furniture**: A panel with a dropdown menu (options: Desk, Closet, BedSideTable, BookShelves, Bookcase, ArmChair, Table, Sofa, Chairs, Bed) and an **OK** button.
- Choose an object from a list**: A panel with a dropdown menu (options: desk3D) and a **Show Properties** button.
- Remove Individual**: A panel with a **Remove Individual** button.
- Or add your object**: A panel with a text input field for "Name for individual" (value: desk3D) and an **OK** button.

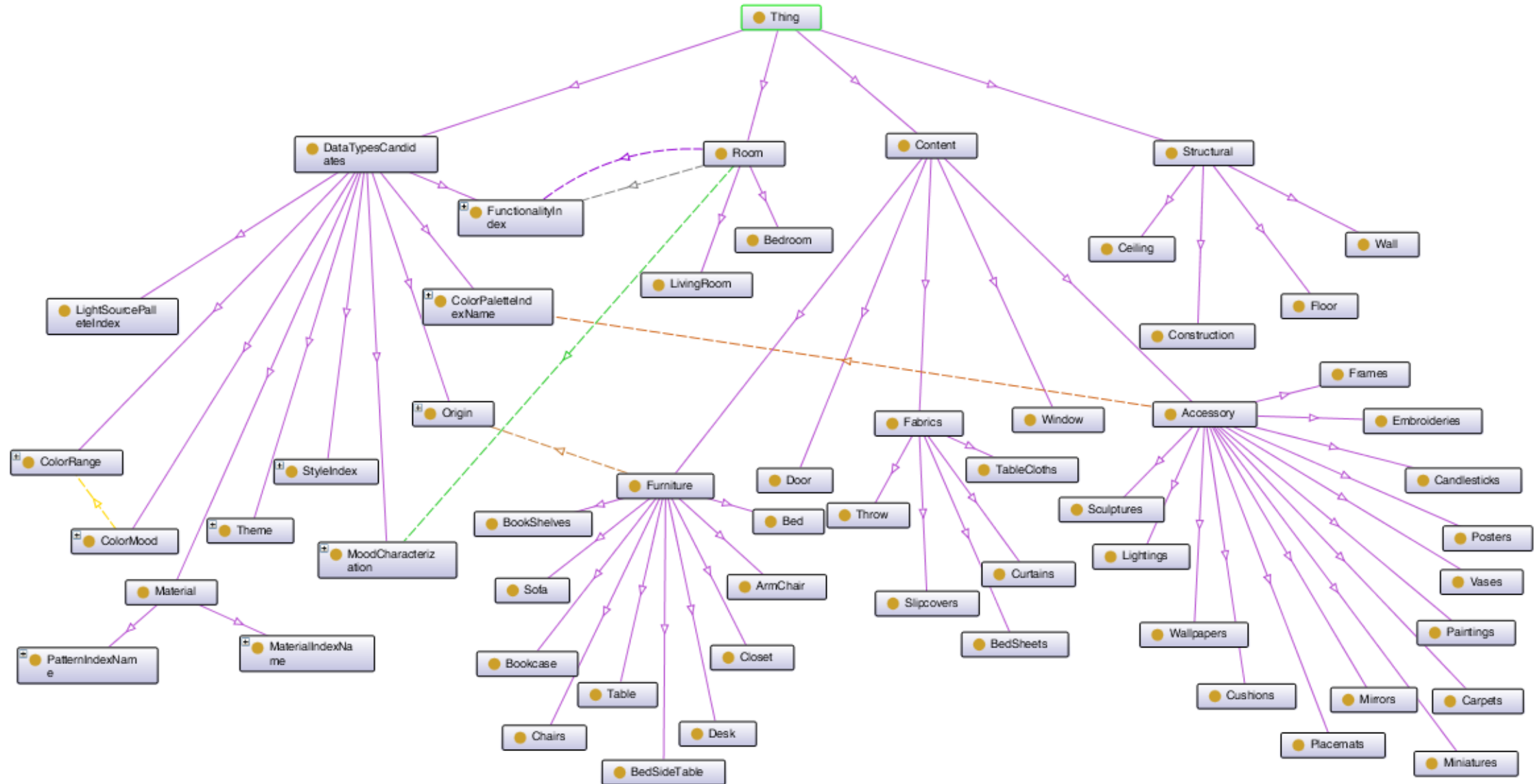
On the right side, there are four property configuration panels:

- Object Property Name:** `has_as_Material`. Value: `Suede`. A color swatch is shown next to the value.
- Datatype Property Name:** `has_as_Functional_Height_Dimension`. Value: `0.80`.
- Datatype Property Name:** `has_as_Height_Dimension`. Value: `0.60`.
- Datatype Property Name:** `has_as_Length_Dimension`. Value: `1.5`.
- Datatype Property Name:** `has_as_Functional_Width_Dimension`. Value: `1.75`.
- Datatype Property Name:** `made_From`. Radio buttons for **Industrial** (selected) and **Handmade**.



2015

# DECORATION-ONTOLOGY



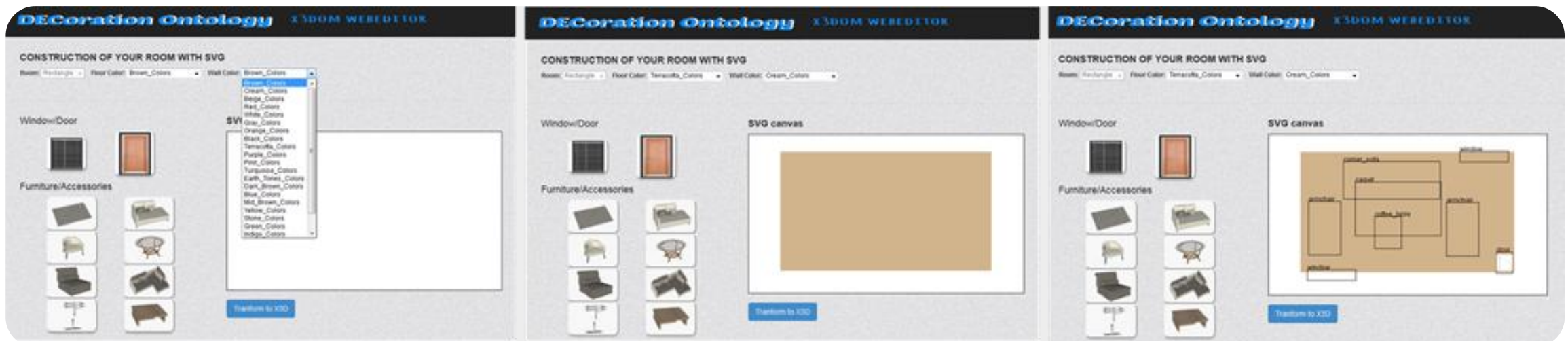


2015

# SVG tool



The **SVG tool** provides a flexible HTML-based environment for the design of a room-space, according to a predefined set of geometrical shapes, furniture and accessories. These characteristics are directly drawn from DEC-O ontology.





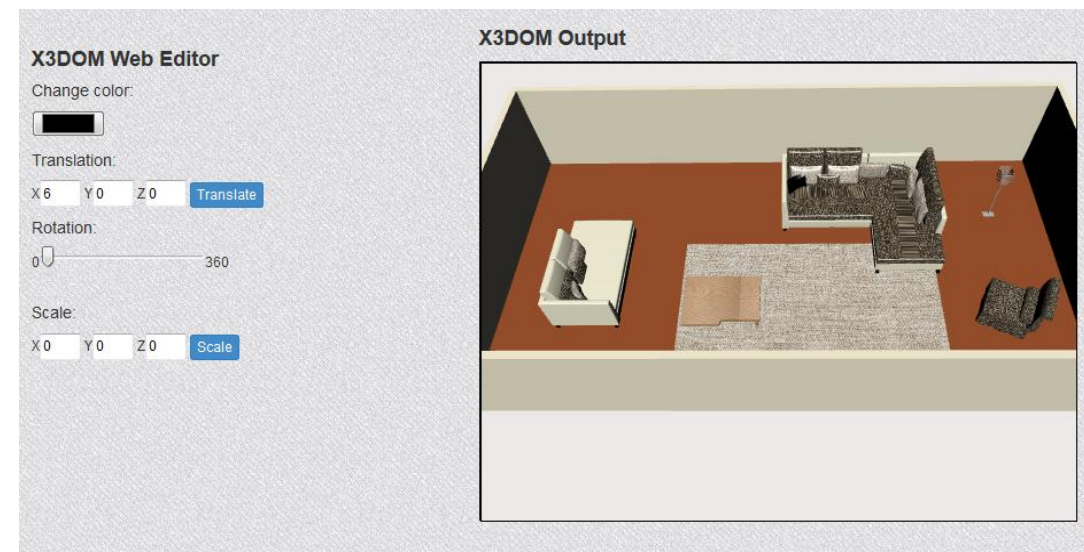
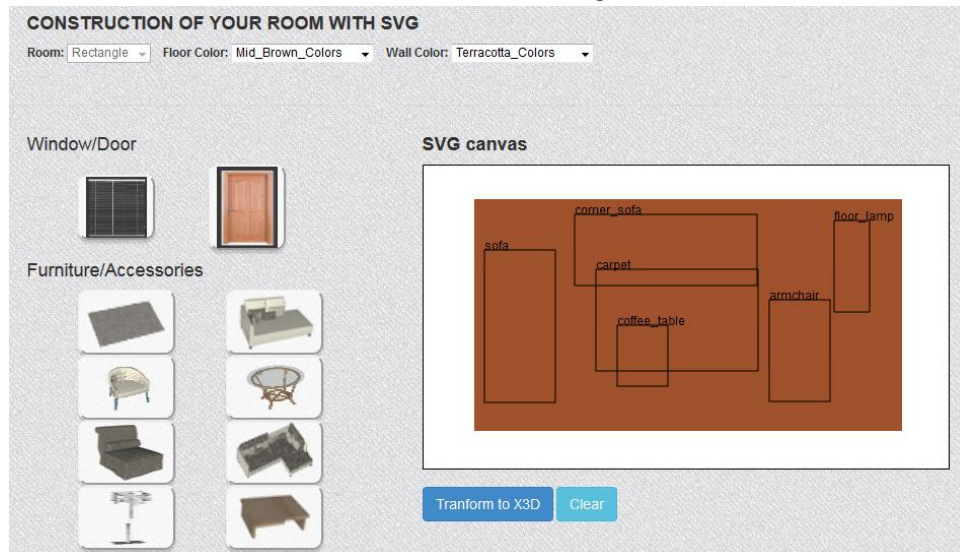


2015

# SVG to X3D converter



The **SVG to X3D converter** which makes use of an XSLT algorithm to automatically transform the scene displayed in these two standards. The first one corresponds to the room that a user has designed with SVG tool, while the second one points to its 3D representation in X3DOM framework.





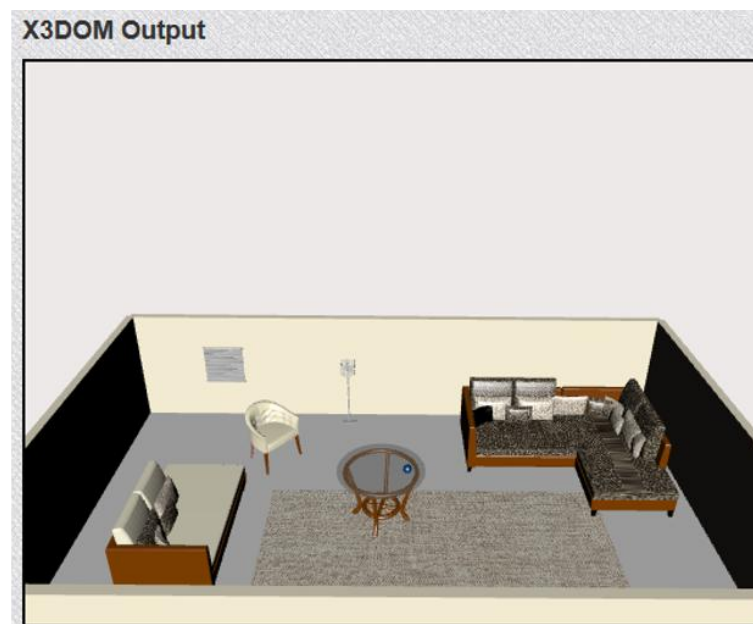
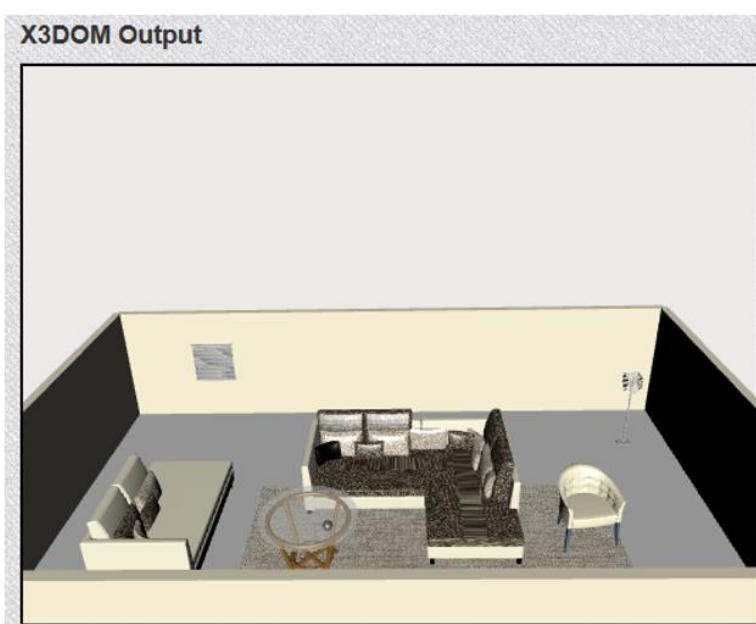


2015

# X3DOM Web Editor



The ***X3DOM Web Editor*** which allows application's users to modify the visualized scene according to their likes. Users have the ability to change the color of any object in the scene, to move it around the room-space, to rotate and scale it.





2015

# Reasoning APIs



Two Reasoning APIs that take advantage of various reasoning mechanisms for the deduction of additional OWL statements with the assistance of a rule-based system and property constructs.

## SWRL

```
DECO:Room(?x1) ^  
DECO:Furniture(?x2) ^  
DECO:Floor(?x3) ^  
DECO:has_as_Colors(?x3, ?clr) ^  
DECO:has_as_Colors(Individual(type(ColorRange), ?clr) →  
DECO:has_a_Style(?x1, Individual(type(StyleIndex)) ^  
DECO:has_as_Functionality(?x1, Individual(type(FunctionalityIndex))) ^  
DECO:has_as_Colors(?x2, Individual(type(ColorPaletteIndexName)) ^  
DECO:has_as_Material(?x2, Individual(type(MaterialIndexName)) ^  
DECO:has_as_Pattern(?x2, Individual(type(PatternIndexName)) ^  
DECO:has_as_Material(?x3, Individual(type(MaterialIndexName)) ^  
DECO:has_as_Pattern(?x3, Individual(type(PatternIndexName))
```

## SPARQL

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-  
ns#>  
PREFIX ns:  
<http://www.semanticweb.org/ontologies/2009/5/Ontology  
1244033197062.owl#>"  
SELECT ?individuals  
WHERE { ?individuals rdf:type ns:Door};
```



2015

# SWRL API (1/2)



The **SWRL API** is able to infer extra entailments from the original OWL statements by applying a finite set of SWRL rules. These rules were authored with the Protégé-OWL editor and their scope is closely related to color theory and decoration properties. DEC-O framework integrated them as its rule-based system and made feasible their execution with the utilization of an external reasoner, the Pellet. These rules comprise aesthetic proposals based on the color of the floor, which are deducted and tallied to various X3D objects. Ultimately, the users have the ability to place any of the proposed objects in the SVG canvas, which is translated into an X3DOM scene.

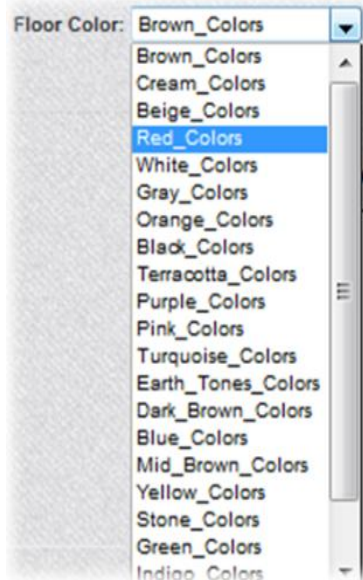


2015

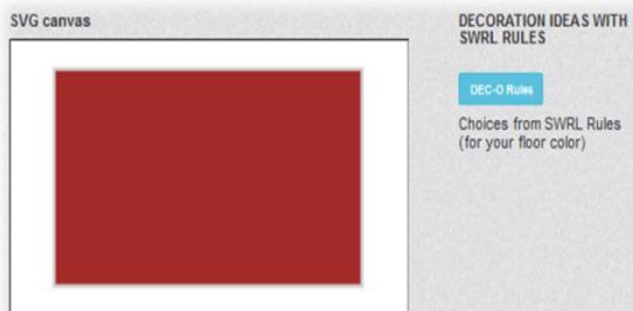
# SWRL API (2/2)



## Step 1



## Step 2



## Step 3

Choices from SWRL Rules (for your floor color)

PoolCushion



CherryCarpet



ChenilleSofa



OvalMotifChair



FloralCurtains



FeltCushion



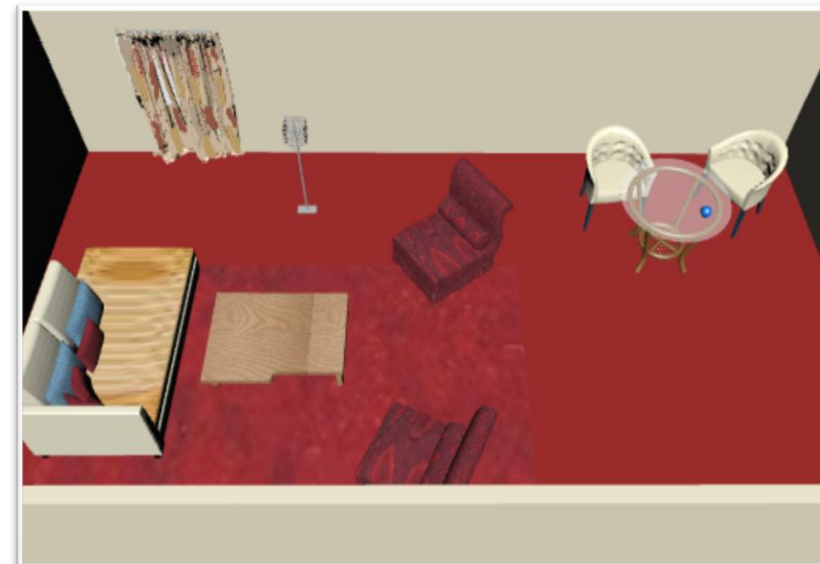
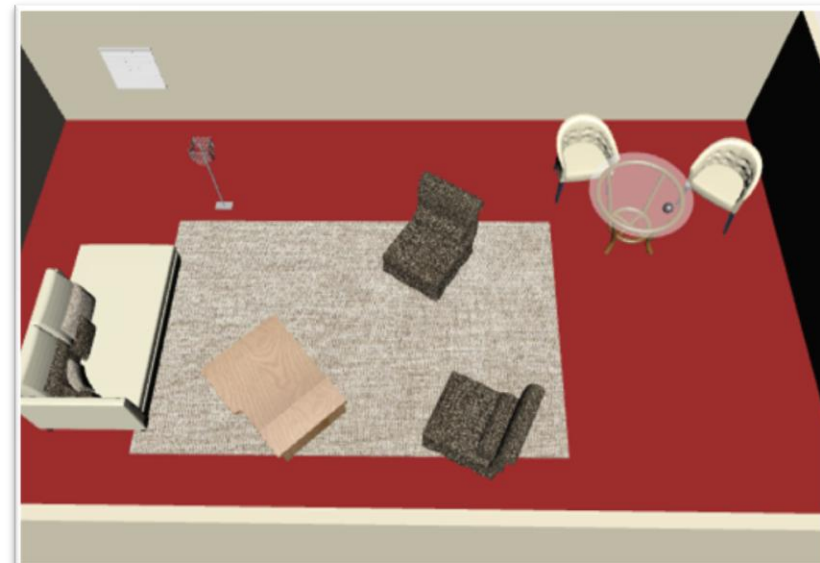
VelvetCushion



HoneySofa



MultitudeCarpet







2015

# SPARQL API (1/2)

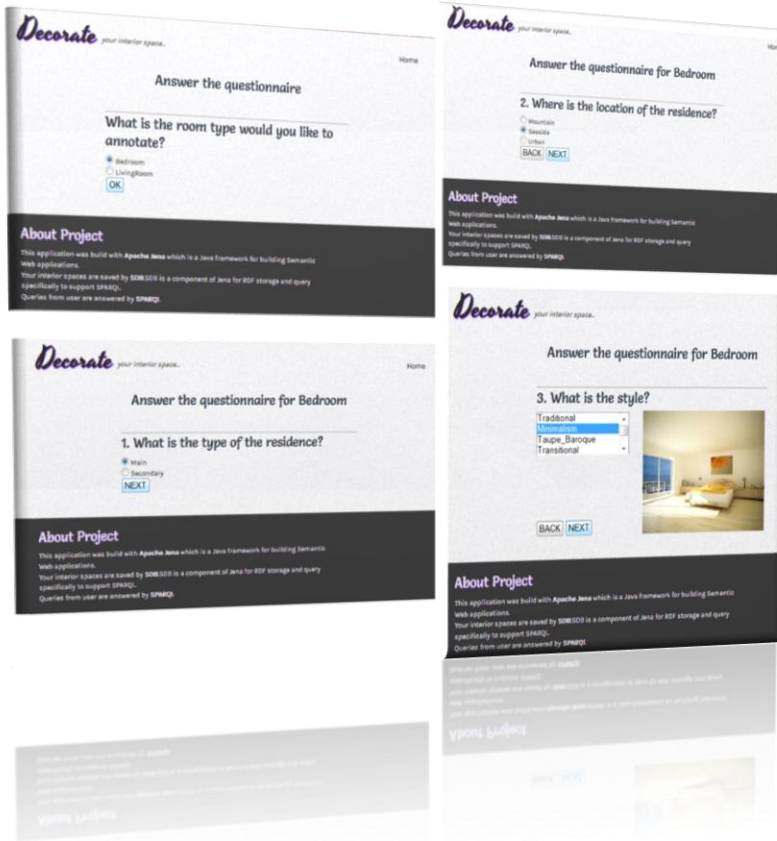


The **SPARQL API** is based on the Jena Model Interface and is responsible for the execution of queries directly upon the OWL knowledge base. Users define the desired criteria for their query -which in turn- retrieves the required information from OWL-DL ontology. By filling out a questionnaire giving as many specifications about the room as they need, the closest match is returned to them.



2015

# SPARQL API (2/2)



```
String queryString
= "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> \n"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> \n"
+ "PREFIX
ns:<http://www.semanticweb.org/ontologies/2009/5/Ontology1244033197062.owl#> \n"
+ "PREFIX
Ontology1244033197062:<http://www.semanticweb.org/ontologies/2009/5/Ontology1244
033197062.owl#> \n"
+ "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> \n"
+ "SELECT ?room \n"
+ "WHERE {\n"
+ "?room ns:has_a_Style ns:" + choice3 + " ;\n"
+ "ns:has_as_Functionality ns:" + choice6 + " ;\n"
+ "ns:has_as_Minor_Functionality ns:" + choice7 + " ;\n"
+ "Ontology1244033197062:has_as_Residence_Type "" + choice1 + ""^^xsd:string
;\n"
+ "Ontology1244033197062:has_as_Residence_Location "" + choice2 +
""^^xsd:string ;\n"
+ "Ontology1244033197062:has_as_TopViewURL "" + choice4 + ""^^xsd:string
;\n"
+ "Ontology1244033197062:has_Approximate_Size "" + choice5 + ""^^xsd:string
;\n"
+ "Ontology1244033197062:has_as_TargetGroup_for_BedRoom "" + choice8 +
""^^xsd:string ;\n"
```





2015

# Conclusion



- Annotate 3D virtual environments to OWL concepts
- Transform 2D architecture designs to corresponding 3D representations
- User questionnaire and AI systems for the efficient reasoning of the ontology based on the likes of the user



2015



**Thank you for your attention!!!**