Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 541–548

Semantic Multi-layered Design of Interactive 3D Presentations

Jakub Flotyński, Krzysztof Walczak
Poznań University of Economics, Poland
email: {flotynski, walczak}@kti.ue.poznan.pl

***Abstract*—Dependencies between interactive 3D content elements are typically more complex than dependencies between standard web pages as they may relate to different aspects of the content—spatial, temporal, structural, logical and behavioural. The Semantic Web approach helps in making data understandable and processable for both humans and computers by providing common concepts for describing web resources. However, semantic concepts may be also used to improve the process of designing content. In this paper, a new approach to semantic multi-layered design of interactive 3D content is presented. The proposed solution provides a semantic representation of 3D content in multiple layers reflecting diverse aspects of 3D content. The presented solution conforms to well-established 3D content and semantic description standards and—therefore—may facilitate creation, dissemination and reuse of 3D content in a variety of application domains on the web.**

***Index Terms*—3D web, 3D content, semantic modelling, ontology, RDF, OWL, RDFS**

I. Introduction

WIDESPREAD use of interactive 3D technologies has been recently enabled by increasing hardware performance, rapid growth in the network bandwidth as well as availability of cheap 3D accelerators and input-output devices. However, the potential of 3D technologies in everyday use may be fully exploited only if accompanied by easy-to-use methods of creating, searching and combining distributed interactive three-dimensional content.

Creating, searching and combining distributed interactive 3D content are much more complex and challenging tasks than in the case of typical web pages. Relationships between components of an interactive three-dimensional virtual scene may include, in addition to its basic meaning and presentation form, also spatial, temporal, structural, logical, and behavioural aspects. The Semantic Web approach makes the described data understandable for both humans and computers achieving a new quality in building web applications that can "understand" the meaning of particular components of content and services as well as their relationships.

Semantic Web standards may be applied to 3D content, leading to much better methods of creating, searching, reasoning, combining and presenting content. 3D content models based on commonly used semantic concepts may be independent of particular description methods and languages. The use of common concepts facilitates dissemination and reuse of individual components of the content that may be semantically assembled to create particular VR/AR presentations depending on the user location, preferences, device used, etc. Moreover, extending geometrical, structural, spatial, logical, and behavioural components with their semantic descriptions permits reasoning on complex semantically described 3D scenes that include these components.

Although a number of methods and languages for programming 3D presentations, and several solutions for creating semantic descriptions of 3D content have been proposed, they do not support layered design of interactive 3D presentations and do not enable semantic representation of components of 3D objects and scenes.

Layered design of 3D presentations provides a separation of concerns between particular semantic layers corresponding to different aspects of the presentation. It reduces the complexity and the number of connections between particular components of the content, facilitating their implementation and exchange, and simplifying the creation of the desirable final presentation.

The main contribution of this paper is an approach to semantic multi-layered representation of interactive 3D presentations. The proposed solution provides a complex representation of 3D content in multiple layers reflecting various aspects of the content—geometry, structure, appearance, scene, logic, and behaviour. The solution conforms to well-established 3D content and semantic description standards.

The remainder of this paper is structured as follows. Sections II and III provide an overview of the current state of the art in the domains of 3D content presentation and semantic description of web resources. Section IV introduces a novel semantic multi-layered model of 3D content. Section V discusses an implementation of the proposed approach. An example of the semantic design of 3D content is explained in Section VI. Finally, Section VII concludes the paper and indicates the possible directions of future research.

II. Interactive 3D Presentations

A number of technologies (including languages, libraries, frameworks, and game engines) have been devised for creating interactive 3D content presentations, in particular built into web applications.

The Virtual Reality Modeling Language (VRML) [1] is an open, textual language devised by the Web3D Consortium for describing static and animated 3D content in a declarative way. A VRML scene is represented as a graph with nodes reflecting different aspects of the described 3D content—geometry, structure, appearance, space, logic, and behaviour.

VRML also supports linking external multimedia resources—images, audio and video. In addition to the use of specific behavioural VRML nodes, the logic and behaviour of the presented 3D objects may be described by embedded imperative ECMAScript code. Several implementations of VRML browsers are available, e.g., ParallelGraphics Cortona3D [2], Bitmanagament BS Contact [3], FreeWRL [4], and InstantReality [5].

The Extensible 3D (X3D) [6] is a successor to VRML, also designed by the Web3D Consortium. X3D introduces several functional extensions to VRML, such as Humanoid Animation, NURBS and CAD geometry. Furthermore, it supports additional XML-based and binary encoding formats as well as basic means for metadata description. Depending on the set of implemented features, different X3D profiles may be selected for the presentation of particular 3D content. Currently, X3D is implemented by a few browsers, e.g., BS Contact, FreeWRL and InstantReality. VRML and X3D enable standardized presentation of 3D content on the web, accessible with additional browser plug-ins. To enable seamless integration of X3D content with web pages, X3DOM [7] has been designed. It is an open source framework intended as a potential extension to HTML5. The content encoded with X3DOM can be presented without additional plug-ins by the majority of modern web browsers.

PDF3D [8] is another approach to 3D content presentation. It utilizes the U3D [9] file format for model representation and a proprietary JavaScript API for programming its behaviour. A PDF document with 3D content may be directly embedded in a web page, and presented with the Abobe Reader plug-in.

A number of libraries have been developed for creating 3D content presentations. Such libraries usually permit programming of the logic and behaviour of the content, while 3D objects in the scene are represented by external resources, which are encoded in, e.g., JSON, COLLADA, AWD, Wavefront OBJ or 3DS. Several libraries have been implemented on the basis of JavaScript and OpenGL to enable 3D presentations built into web pages—WebGL [10], GLGE [11], JebGL [12], Oak3D, [13], and O3D [14]. Other libraries (Papervision3D [15], Alternativa3D [16], Away3D [17] or Sandy 3D [18]) have been developed for the ActionScript—an object-oriented dialect of ECMAScript that is used for web applications compatible with the Adobe Flash Player. Web presentations of 3D content can also be built using Java applets implemented with JOGL [19] or Java3D [20] libraries.

Another group of solutions incorporates game engines, which allow for the development of complex 3D web applications enriched with additional aspects, such as physics, collision detection, artificial intelligence and networking. For instance, Unity [21] and Unreal [22] permit 3D presentations accessible with the Unity Web Player and the Adobe Flash Player.

III. Semantic Descriptions of 3D Content

In this section, the state of the art in the area of semantic description of web content is presented. In particular, basic techniques for describing the semantics of web resources, metadata and ontologies for 3D multimedia content as well as methods of semantic creation of 3D content are considered.

A. Foundations for the Semantic Web

The primary technique for describing data semantics on the web is the Resource Description Framework (RDF) [23]—a standard devised by the W3C. RDF introduces general rules for making statements about resources. Each statement is comprised of three elements: *a subject* (a resource described by the statement), *a predicate* (a property of the subject) and *an object* (the value of the property).

RDF introduces classes (as types of resources), containers and lists to provide basic concepts for semantic descriptions. However, these notions are often insufficient for describing the semantics of complex resources. The RDF Schema (RDFS) [24] and the Web Ontology Language (OWL) [25] are W3C standards based on RDF providing higher expressiveness for semantic descriptions of web resources, e.g., hierarchies of classes and properties, constraints, property restrictions as well as operations on sets. OWL defines a set of profiles, which differ in complexity and decidability. Semantic Web Rule Language (SWRL) [26] is an extension to OWL devised for describing semantic Horn-like rules. While RDF and RDF-based techniques permit the creation of ontologies and knowledge bases, SPARQL [27] is a language for querying RDF data sources.

RDF and RDF-based technologies have been intended as the basis of the Semantic Web. Hence, they are applicable to any type of web resources, but they do not address specific aspects of particular content types (especially 3D). That is why application-specific ontologies are required to describe content of various types on the web.

B. Metadata and Ontologies for 3D Content

To provide a common space for the classes and properties of resources on the web, several vocabularies, metadata schemas and ontologies have been proposed for various application domains and types of resources, in particular for 3D multimedia content. The Multimedia Content Description Interface (MPEG-7) [28] is an extensive standard that defines a set of tools for creating metadata—Descriptors, Description Schemes, the Description Definition Language and Coding Schemes. There is a wide range of target media types that may be described with MPEG-7—images, audio, video and 3D objects, including multimedia content in VR applications [29].

A few ontologies have been proposed for multimedia content. The Ontology for Media Resources [30] has been devised by the W3C on the basis of RDF, RDFS and OWL, as a common solution for describing multimedia published on the web. It provides an interoperable core vocabulary that is mapped to a set of metadata formats for media content (e.g., MPEG-7). The Core Ontology for Multimedia (COMM) [31], [32] is another solution designed for describing media content such as images, audio, video and 3D objects. COMM is based on MPEG-7, but it represents knowledge with open Semantic Web solutions avoiding some interoperability problems that occur in MPEG-7, e.g., with semantically equivalent descriptors that are processed in different manners [32].

C. Semantic Creation of 3D Scenes

Several works have been devoted to the semantic creation of 3D content. In [33], an approach to creating interoperable RDF-based Semantic Virtual Environments, with system-independent and machine-readable abstract descriptions has been presented. In [34][35][36], a rule-based framework based on MPEG-7 has been proposed for the adaptation of 3D content, e.g., geometry and texture degradation or filtering of objects. The content can be described with different encoding formats (in particular X3D), and it is annotated with an indexing model. In [37], an integration of X3D and OWL using scene-independent ontologies and the concept of semantic zones are proposed to enable querying 3D scenes at different levels of semantic detail and they have been used to implement a tour through the Venetian Palace.

In [38], a method of structured design of VR content has been proposed. In [39][40][41], an approach to generating virtual words upon mappings of domain ontologies to particular 3D representation languages (e.g., X3D) has been considered. The following three content generation stages are distinguished: specification of a domain ontology, mapping of the domain ontology to a 3D description language, and generation of the final presentation. The solution stresses spatial relations (position and orientation) between objects in the scene.

Several works have been conducted on the modelling of behaviour of VR objects. In [42], the Beh-VR approach and the VR-BML language have been proposed for the dynamic creation of behaviour-rich interactive 3D content. The proposed solution aims at simplification of behaviour programming for non-professionals. Another method facilitating the modelling of content behaviour [43][44][45] provides a means for expressing primitive and complex behaviours as well as a set of temporal operators. Finally, a rule-based ontology framework for feature modelling, consistency check and feature modelling, has been explained in [46].

IV. Semantic Model of Interactive 3D Content

Although several approaches have been devised for semantic modelling of 3D content, they lack solutions for semantic representation of 3D content. Layered design of 3D presentations provides a separation of concerns between particular layers corresponding to different aspects of the designed presentation, which are described in individual, specific manners. It reduces the complexity and the number of connections between components, which are incorporated in different layers, facilitating their implementation and exchange, and simplifying the creation of the desirable content. In addition, possible implementation profiles of a structured solution may cover only layers reflecting the required aspects of the context—geometry, structure, appearance, space, logic, or behaviour.

In this section, a novel approach to the semantic design of interactive 3D presentations is proposed. The presented solution is based on a multi-layered semantic representation of 3D content. The model complies with the Semantic Web approach, and it has several important advantages in comparison to the available solutions for modelling of 3D content. First, the components of semantically described content may be searched, explored and reused by applying well-established Semantic Web standards. Second, it allows for reasoning on the content, which further enables discovering knowledge that has not been explicitly encoded. Third, 3D content described by commonly used concepts is platform- and standard-independent, and it may be transformed to final presentations encoded in different languages, depending on particular requirements, e.g., the context of interaction, client device used, user preferences.

The proposed semantic model of 3D presentations is depicted in Fig. 1. It includes six layers corresponding to distinct aspects of 3D content and different stages of the development of 3D presentations—*Geometry Layer*, *Structure Layer*, *Appearance Layer*, *Scene Layer*, *Logic Layer*, and *Behaviour Layer*. The subsequent layers are partly dependent—every layer uses only its own concepts and the concepts specified in the lower layers (gray arrows), i.e. a 3D presentation may fully utilize the components of a particular layer without referring to its higher layers. Like in OWL, the concepts defined are classes as well as data properties and object properties describing respectively the attributes and relations between class instances. The specification of the relations between class instances in the class definitions indicates optional and obligatory dependencies between components, which are specified during the modelling process while creating instances of these classes.

In the proposed approach, a 3D presentation may be created at an arbitrary layer and the development process includes the creation of components which are defined in the selected layer and its lower layers. For instance, design of a complex 3D scene with behaviour covers all of the layers of the presented model. However, presentations that consist only of lower layers are also possible, e.g., reusable structural 3D objects without appearance that are to be injected into different complex presentations can be created at layer 2.

Two types of classes are distinguished—*abstract* and *concrete*. With the presented solution, a developer designs a presentation by creating instances of the selected concrete classes. The properties in the presented model are specified as optional or obligatory with a given cardinality. A component in the resulting scene is assigned the properties of its class and the superclasses. Created objects may be described with desirable data properties and linked one to another with object properties.

The proposed semantic model has been designed with regards to concepts commonly used in well-established 3D content representation languages and libraries, such as X3D, Unity, etc. In the diagram presented in Fig. 1, several data properties which are typical for different 3D content representation standards as well as exact data types and ranges of properties have been omitted as they are not crucial for the proposed idea. The presented model contains key concepts used in designing 3D presentations, but it may be extended with new classes and properties depending on the particular use.

The proposed model is semantically complete—no classes or properties are created or removed during the content presentation, only the values of properties may change. For instance, a moving object may be stopped by turning off its motion animation, but not by removing it.

The following sections describe the semantic design of 3D presentations with regards to the particular layers of the proposed model. The design starts with the description of basic shapes included in the created presentation (the *Geometry Layer*). Second, the basic objects are assigned spatial properties to be combined into arbitrary complex structural objects—complex shapes with spatial dependencies (the *Structure Layer*). Third, appearance properties are added to the complex structural objects to create visual components, which may be illuminated and enriched with environmental effects (the *Appearance Layer*). Next, the components with appearance are included in a scene with a viewpoint and navigation modes (the *Scene Layer*). Finally, logic and behaviour may be added to the scene and all its components (the *Logic* and *Behaviour* layers).

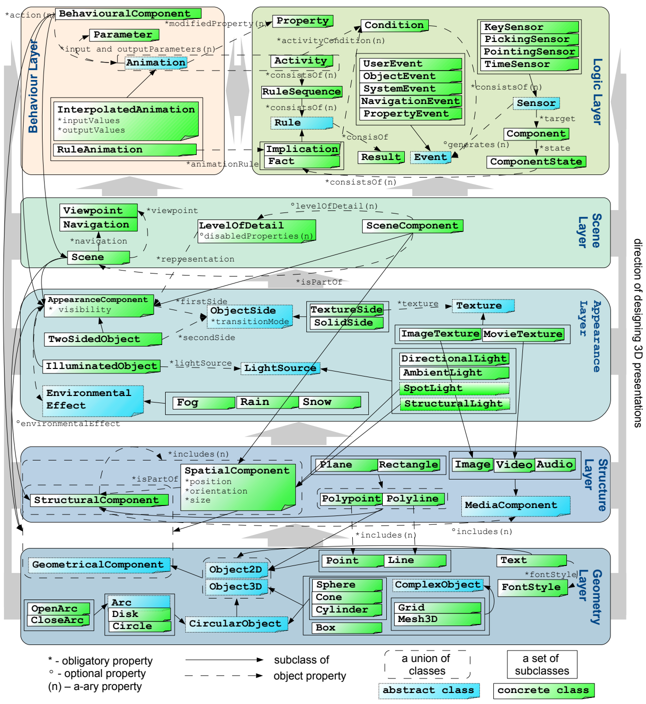

Fig. 1. The semantic model of interactive 3D content

A. Geometry Layer

The *Geometry Layer* is the base layer of the proposed semantic model and it includes concepts of basic 2D and 3D geometry shapes, which physically form the presentation. In the proposed model, this layer is aimed at the low-level manipulation of the geometry of scene components. The primary `GeometricalComponent` class is abstract, thus all shapes created are instances of its descendants. The classes of the *Geometry Layer* are relatively simple, in comparison to the classes defined in the other layers, as they are mainly determined by using data properties specific to 2D and 3D content. Since the components of this layer have no common point of spatial reference, their spatial properties (exact size, position and orientation) cannot be given. As this layer does not specify other aspects of 3D content (space, appearance, logic and behaviour), it allows only for modelling simple separated objects that do not combine into complex models and scenes. Hence, this is not sufficient for building practical 3D presentations. `GeometricalComponents` have limited expressiveness in 3D presentations, as they can only represent isolated integral objects, e.g., 3D models of sculptures or shapes of buildings.

B. Structure Layer

The *Structure Layer* is the second layer of the model and it depends only on the *Geometry Layer*. While the `GeometricalComponents` describe basic shapes included in a scene, `StructuralComponents` enable creating logical and spatial combinations of them into complex objects. `StructuralComponents` may recursively include other `StructuralComponents` as well as `Media` (`Images`, `Audio` and `Video`) and `Spatial` components. `SpatialComponents` are `Geometrical` and `Structural` components with spatial properties (`position`, `orientation` and `size`) set relatively to the parent `StructuralComponent`.

Thanks to the specific meaning of the `include` inverse functional property, `StructuralComponents` may be considered as a whole while assigning some properties (sub-properties of the `structurallyTransitive` property) in higher layers, e.g., appearance or spatial properties are automatically set for all the subcomponents of a complex `StructuralComponent` when the property is set to it.

Although the components of this layer have no appearance and thereby they are not sufficient for practical 3D presentations, they describe its structure and are managed from within higher layers, e.g., when determining logic and behaviour.

C. Appearance Layer

The *Appearance Layer* is aimed at adding appearance to `Geometrical` and `Structural` components that are defined in the previous layers. The primary `AppearanceComponent` may be either a single- or a two-sided object. Each side can be covered with textures (images or movies), or described by typical appearance properties (`colour`, `transparency`, etc.). The same appearance properties may be set for a whole `StructuralComponent` with all its subcomponents by specifying `transitionMode`—to ignore or respect the individual settings of subcomponents. In addition, `AppearanceComponents` may be illuminated by `LightSources` of several types and enriched with `EnvironmentalEffects`. At this layer, a 3D presentation consists of a set of logically structured components with appearance that have no logic and behaviour, e.g., static museum artefacts, furniture, buildings, etc.

D. Scene Layer

The primary class of the *Scene Layer* is the `Scene` that is an `AppearanceComponent` or a `StructuralComponent`. It has assigned a list of `Viewpoints` and a list of `Navigation` modes. A background and `EnvironmentalEffects` of the `Scene` may be specified at the lower *Appearance Layer*. The subcomponents of `Scene` are `SceneComponents` inheriting from either `Spatial` or `Appearance` components. Complex `SpatialComponents` may be presented with different `LevelsOfDetail` depending on the current distance between the object and the observer. Each `LevelOfDetail` indicates a set of `AppearanceComponents` that are the ingredients of a particular target object, to be visible, and a set of appearance properties to be disabled. At this layer, designing complex navigable 3D presentations (without logic and behaviour) is feasible, e.g., static virtual museum exhibitions, models of cities, etc.

E. Logic Layer

The *Logic Layer* is intended as a framework providing concepts for describing the logic of components that are defined in other layers. This layer does not introduce apparent effects to 3D presentations, as opposed to the previous layers. As the logic may be related to different aspects of the created 3D content, the *Logic Layer* links all of the previous layers. This layer has been designed according to the rule-based approach that enables both complex declarative descriptions and reasoning on the created 3D presentations. The primary entity of logic description is a `Rule` that may be either a `Fact` or an `Implication` (if a `Condition`—a conjunction—is satisfied then a `Result` is also satisfied). While `Facts` describe the `ComponentStates` of any object defined in any layer of the presented semantic model, `Implications` are mainly used for describing complex `RuleSequences`. In a `RuleSequence`, the result of an `Implication` is the `Condition` of its following `Implication`. Such a chain permits an ordered performance of consecutive steps, like in typical imperative programming. In turn, several independent sequences may create an `Activity` that is initiated when a common required alternative of `Conditions` is satisfied. A `Condition` may refer to `Events` (generated either by a user interaction, an object, the navigation, the system, or a change of a property value). Events are generated by `Sensors` (e.g., `KeySensor` or `PointingSensor` for user interactions, `PickingSensor` for object interactions, `TimeSensor` for system interactions). `Events` and `Sensors`, as well as the relations between them defined in the model are similar to the corresponding concepts widely-used in other technologies for describing 3D, thus they are not described in detail in Fig. 1.

F. Behaviour Layer

The *Behaviour Layer* provides concepts that introduce apparent behavioural effects to 3D presentations built upon the previous layers. Like the *Logic Layer*, the *Behaviour Layer* leverages all the lower layers, as the behaviour of a component may concern different aspects of 3D content. In particular, selected classes of the *Logic Layer* are especially important for defining behaviour. The primary `BehaviouralComponent` class extends the `Geometrical`, `Structural`, `Appearance` components, or `Scenes` with an arbitrary number of `actions`—`Activities` or `Animations`. `Activities`, which are conditionally dependent on `Events`, enable programming interactions. While `Activities` are entirely defined in the *Logic Layer*, `Animations` are behavioural objects with sets of input and output `Parameters` used to control `modifiedProperties`. Two types of `Animations` may be distinguished. `RuleAnimations` utilize `Implications` to bind input and output `Parameters` in a functional manner. `InterpolatedAnimations` specify sets of values of input and output `Parameters`. The changes of classification attributes are gradual. For numeric attributes with continuous domains, intermediate input and output values that are not specified explicitly, are calculated from their neighbouring values. At this layer, the created presentations may be dynamic 3D scenes with components including all the aspects of 3D content. The presentation may change in time due to interactions between objects in the scene, user interactions, system interactions, etc.

V. Implementation of the Proposed Semantic Model

The proposed semantic model of interactive 3D content has been implemented as an ontology with well-established Semantic Web standards (RDF, RDFS, OWL, SWRL and SPARQL), using the Protege editor [47]. The implementation rules of the model are explained below.

A. Class Definitions

The classes of the model are described by the `owl:Class` type. Inheritance between classes is described by the `rdfs:subClassOf` property. In some cases, the subclasses of a class union have been defined, e.g., `BehaviouralComponents` may inherit either from `Geometrical`, `Appearance`, `StructuralComponents`, or `Scenes`; behavioural `Actions` may be either `Activities` or `Animations`.

B. Disjointness of Classes

In the presented model, the subclasses of a common superclass (surrounded by solid rectangles in the diagram) are mutually disjoint classes described with the `owl:disjointWith` property, e.g., 2D and 3D objects, media components, events.

C. Property Definitions

Data and object properties are reflected by the `owl:DatatypeProperty` and the `owl:ObjectProperty`, respectively. In some cases, properties defined for a superclass need to be implemented individually for different subclasses, e.g., the `size` specifies two values (two dimensions) for 2D components, and three values (three dimensions) for 3D objects. Such dependencies have been encoded as hierarchies of properties using the `rdfs:subPropertyOf`.

D. Optionality and Cardinality of Properties

Some classes of the model may allow or require their instances to have selected properties specified with a particular cardinality, e.g., a `SpatialComponent` included in a `StructuralComponent` must be assigned a single `position`, a single `orientation` and a single `size`. Classes with such requirements have been indicated as subclasses or equivalent classes (`owl:equivalentClass`) of restricted superclasses (the `owl:Restriction` property). Optional unary, optional n-ary, obligatory unary, and obligatory n-ary properties of components are described, respectively, by the `owl:maxQualifiedCardinality`, `owl:someValuesFrom`, `owl:qualifiedCardinality`, and `owl:minQualifiedCardinality` properties set to 1. However, to check if the exact and maximal cardinality are satisfied, the closed world assumption should be made. This requirement can be met for 3D presentations designed strictly for the particular use, whose components come from limited or well-known sources.

E. Domains and Ranges of Properties

To unambiguously connect properties to the corresponding classes, domains and ranges have been determined for properties. In the vast majority of cases, domains and ranges enclose single classes, e.g., the sides of an `AppearanceObject`, the `Viewpoint` of a `Scene`. However, a few domains and ranges are unions of classes, e.g., a `consistOf` property may be indicated for an `Activity`, a `RuleSequence` or a `Rule`; an `includes` property can indicate a `Geometrical` or a `StructuralComponent`.

F. Structurally Transitive Properties

Structurally transitive properties influence not only the component they are specified for, but also the subcomponents it `includes`. Structurally transitive properties have been defined with SPARQL rules (Listing 1)—if a component has a property (line 2) that is a structurally transitive property (3), and the component has a subcomponent (4), then this subcomponent is also assigned the property with the same value (1).

In the presented semantic model, structurally transitive properties are related to appearance and spatial aspects of the content, e.g., colour, transparency, light sources (the parts of an illuminated object are also illuminated).

Listing 1. A SPARQL rule spreading structurally transitive properties

```
1 construct {?subcomponent prop ?value.}
2 where { ?component ?prop ?value.
3         ?prop rdfs:subPropertyOf STproperty.
4         ?component includes ?subcomponent. }
```

G. Logic Definitions

To enable complex descriptions of logic and reasoning on the content, components of the *Logic Layer* are implemented in SWRL. An `Implication` with a `Condition` and a `Result` is encoded as a `ruleml:imp` element containing a body (`ruleml:_body`) and a head (`ruleml:_head`). A `RuleSequence` is mapped to a list of `ruleml:imp` elements, in which the head of an implication is the body of the next one. Several independent sequences which are attainable by a common `Condition` (their initial rules have the same body) form an `Activity`.

VI. Example Layered Design of a 3D Artefact Model

In this section, an example of the use of the implemented semantic model is presented in the context of designing a 3D presentation of a virtual museum artefact, e.g., in development of educational games, creating commercial presentations, etc. The artefact is a complex static reusable 3D component that represents a bronze statue with a hat. The example focuses on modelling activities performed by a developer in the consecutive layers of the proposed model.

The design process is illustrated and based on an instance of the model (a knowledge base indicated by the `museum` prefix, Listing 2), which is compliant with the implemented model ontology (the `sm` prefix) and includes descriptions of the created objects, their properties and logic rules. The knowledge base is manually transformed to a concrete 3D scene. However, a tool for automatic transformation could be developed. In the presented example, the resulting 3D content description is encoded in X3D (Listing 3), however, other formats could be used as well.

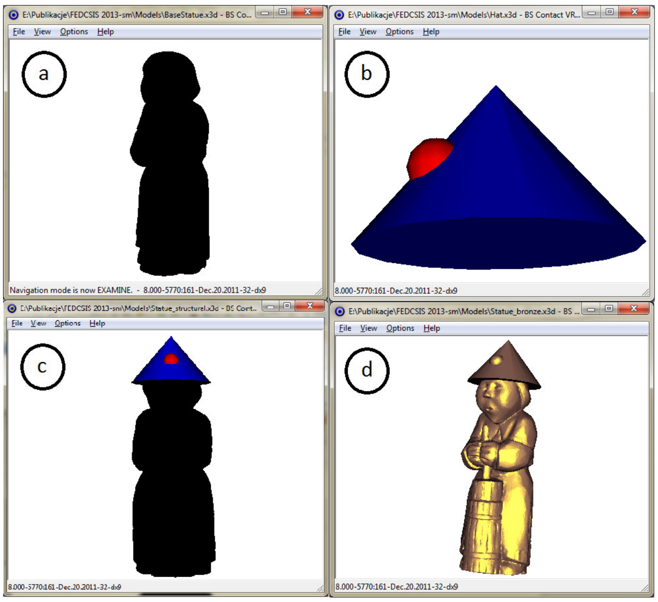

Fig. 2. The results of the consecutive stages of the example layered design of the 3D artefact: the geometry of the artefact (a), the hat element (b), the artefact as a structural component (c), the artefact as a component with appearance (d)

Fig. 2 depicts particular steps of the design process. During the first stage of the process, a `Mesh3D` geometrical object (Listing 2, line 17) is created by a scanner or a modelling tool (e.g., Blender or 3ds Max) to describe the shape of the body of the statue—its coordinates and coordinate indexes (2/20-21). The generated X3D code (3/18-19) leads to the view depicted in 2a. This is not a full 3D presentation due to the lack of properties describing the appearance—the created geometrical object is theoretically invisible.

In the second stage, a desirable `Hat` is found in another existing knowledge base and retrieved—e.g., by a SPARQL query with given conditions (Fig. 2b). In contrast to the body of the statue, the `Hat` is not limited only to geometry, but it has also structure, appearance, and spatial properties applied to its subcomponents (a brim and a decoration, Listing 2, lines 4,5). In the resulting X3D document the `Hat` is represented by a `Transformation` element with nested `Shape` elements (3/7-16).

Listing 2. An example of a semantically designed virtual 3D artefact

```
1  museum:Hat rdf:type owl:NamedIndividual ;
2    rdf:type sm:SpatialComponent ,
3    sm:SceneComponent ;
4    sm:includes museum:HatBrim ;
5    sm:includes museum:HatDecoration ;
6    sm:size ”50 50 50” ;
7    sm:position ”−5 100 −205” ;
8    sm:orientation ”0 0 0” .
9  museum:BronzeStatue rdf:type owl:NamedIndividual ,
10   sm:StructuralComponent , sm:AppearanceComponent ,
11   sm:Scene .
12   sm:includes museum:BronzeStatueBody ;
13   sm:includes museum:Hat ;
14   sm:firstSide museum:BronzeStatueSide ;
15   sm:viewpoint museum:Viewpoint;
16   sm:navigation museum:Navigation .
17 museum:BronzeStatueBody rdf:type sm:Mesh3D ,
18   owl:NamedIndividual , sm:SpatialComponent ,
19   sm:SceneComponent ;
20   sm:coordinateIndex ”...” ;
21   sm:coordinates ”...” ;
22   sm:size ”35 30 150” ;
23   sm:orientation ”0 0 0” ;
24   sm:position ”−5 100 0” .
25 museum:BronzeStatueSide rdf:type sm:SolidSide ,
26   owl:NamedIndividual ;
27   sm:color ”0.65 0.45 0.4” ;
28   sm:transitionMode ”override” .
29 museum:Viewpoint rdf:type sm:Viewpoint ,
30   owl:NamedIndividual ;
31   sm:position ”−11.5858 −6.00235 1038.03” ;
32   sm:orientation ”1.0 0.0 0.0 0.0” .
33 museum:Navigation rdf:type sm:Viewpoint ,
34   owl:NamedIndividual ;
35   sm:mode ”examine” ;
36   sm:transitionType ”linear” .
```

At the third stage, a structure is added to the designed model to represent its complexity. A new `StructuralComponent BronzeStatue` is created, and it includes both the `Hat` and the `BronzeStatueBody` (2/9-13). Since the `Hat` and the `Body` are parts of the `Statue` their `positions`, `orientations` and `sizes` need to be specified relatively to the parent `BronzeStatue` component (2/6-8,22-24). Then the subcomponents are inferred to be `SpatialComponents` (2/2,18), as they are defined as parts of a `StructuralComponent`. This structuralization leads to the X3D code (3/7-20), excluding an instruction describing the appearance of the statue body (3/17). At this stage of the design, the complex `BronzeStatue` may be illustrated as a complex component with spatial relations between its subcomponents (Fig. 2c).

At the next stage, appearance is specified for the whole `BronzeStatue` (2/14). The `BronzeStatueSide` (2/25-28) is a `SolidSide` component that defines a colour (2/27) used for the entire complex `BronzeStatue` overriding colours set for its subcomponents (2/28). The corresponding X3D `diffuseColor` attribute is set appropriately for both the `StatueBody` and the parts of the `Hat` (3/10,14,17). Since the appearance is specified for the whole statue (and not only for its `Hat`), it is inferred to be an `AppearanceComponent` (2/10), as it is a `StructuralComponent` with an `ObjectSide` assigned. The resulting X3D scene is depicted in Fig. 2d and Listing 3, lines 7-20—it is a reusable complex component with the appearance and the spatial relations between its subcomponents.

The last stage encloses activities performed in the *Scene Layer* and it extends the previous `AppearanceComponent` statue with a `Viewpoint` (2/29-32) and a descriptor of `Navigation` (2/33-36). With a `Viewpoint` and a `Navigation` mode specified, the `BronzeStatue` starts to be classified as a `Scene` (2/11), and its subcomponents—the `Hat` and the `Body`—as `SceneComponents` (2/3,19). The corresponding X3D description contains all the instructions in Listing 3, including lines 5,6.

Listing 3. The final representation of the virtual 3D artefact in X3D

```
1  <?xml version=”1.0” encoding=”UTF−8” standalone=”no”?>
2  <!DOCTYPE X3D SYSTEM ”x3d−3.0.dtd”>
3  <X3D profile=”Immersive” version=”3.0”>
4   <Scene>
5    <Viewpoint position=”−11.5858 −6.00235 1038.03” orientation=”1 0 0 0” />
6    <NavigationInfo type=”EXAMINE” transitionType=”LINEAR” />
7    <Transform><Transform>
8     <Transform rotation=”−1.0 0.0 0.0 1.5708” translation=”−5 100 −205 ” scale=”
        50 50 50”>
9      <Shape><Cone height=’1.2’ />
10      <Appearance><Material diffuseColor=”0.65 0.45 0.4” /></Appearance>
11    </Shape></Transform>
12    <Transform rotation=”−1.0 0.0 0.0 1.5708” translation=”−5 125 −200 ” scale=”
        50 50 50”>
13     <Shape><Sphere radius=’0.2’ />
14      <Appearance><Material diffuseColor=”0.65 0.45 0.4”/></Appearance>
15    </Shape></Transform>
16    </Transform>
17    <Shape><Appearance><Material diffuseColor=”0.65 0.45 0.4” />
18     </Appearance><IndexedFaceSet coordIndex=”...”>
19     <Coordinate point=”...” /></IndexedFaceSet>
20 </Shape></Transform></Scene></X3D>
```

VII. Conclusions and Future Works

In this paper, a novel approach to the semantic multi-layered design of interactive 3D presentations has been proposed. The presented division of the structure of 3D content into several distinct semantic layers facilitates the content creation process at the level of 3D model representation. However, the considered model does not address 3D content creation at an arbitrary high level of semantic abstraction, in particular by the use of domain concepts and ontologies. In addition, the presentations created with the model require explicit specification of all the components and relationships between them, which need to be presented in the resulting scene. Methods of semantic modelling and composition of 3D content at an arbitrarily high level of abstraction may be proposed to permit implicit conditional query-based assembly of complex 3D scenes.

Other possible directions of future research incorporate several facets. First, although the presented approach is independent of any modelling tools, the use of semantic editors (e.g., Protege [47]) is highly recommended as it significantly facilitates working with the utilized Semantic Web standards. A specific development environment may be devised to support designing 3D presentations with regard to the consecutive layers of the presented model. Second, the implementation of the model should be evaluated and compared to other platforms in terms of the simplicity of 3D content creation. Furthermore, translators for selected target languages and technologies might be implemented, e.g., Java3D or Unity. To permit semantic exploration of 3D content in real-time, a persistent mapping between the primary semantic representations and the generated final scenes encoded in particular 3D content representation languages should be elaborated. Finally, the context of user–system interaction (e.g., user location, preferences, client device, etc.) can be semantically modelled to enable multi-platform 3D content presentations.

VIII. Acknowledgements

This research work has been partially funded by the Polish National Science Centre grant No. DEC-2012/07/B/ST6/01523.

References

[1] ISO/IEC 14772-1:1997. The Virtual Reality Modeling Language *http://www.web3d.org/x3d/specifications/*. Retrieved May 17, 2013.
[2] Cortona3D, *http://www.cortona3d.com/*. Retr. May 17, 2013.
[3] Bitmanagement, BS Contact, *http://www.bitmanagement.com/products/interactive-3d-clients/bs-contact*. Retrieved May 17, 2013.
[4] FreeWRL, *http://freewrl.sourceforge.net/*. Retrieved May 17, 2013.
[5] Instantreality Framework 2.0. Fraunhofer IGD (2011) *http://instantreality.de/home/*. Retrieved May 17, 2013.
[6] ISO/IEC 19775-1:2008. Extensible 3D (X3D) (2008) *http://web3d.org/x3d/specifications/*. Retrieved May 17, 2013.
[7] X3DOM, *http://www.x3dom.org/*. Retrieved May 17, 2013.
[8] Visual Technology Services, PDF3D—3D Visualization and Technical Publishing Technology, *http://www.pdf3d.com/*. Retrieved May 17, 2013.
[9] U3D File Format, *http://ecma-international.org/*. Retrieved May 17, 2013.
[10] WebGL Specification, *https://www.khronos.org/registry/webgl/specs/1.0/*. Retrieved May 17, 2013.
[11] GLGE, *http://www.glge.org/*. Retrieved May 17, 2013.
[12] JebGL, *http://http://jebgl.com/*. Retrieved May 17, 2013.
[13] Oak3D, *http://www.oak3d.com/*. Retrieved May 17, 2013.
[14] O3D, *http://code.google.com/intl/pl/apis/*. Retrieved May 17, 2013.
[15] Papervision3D, *http://papervision3d.org/*. Retrieved May 17, 2013.
[16] Alternativa, *http://alternativaplatform.com/en/*. Retr. May 17, 2013.
[17] Away3D, *http://away3d.com/*. Retrieved May 17, 2013.
[18] Sandy 3D, *http://www.flashsandy.org/*. Retrieved May 17, 2013.
[19] JOGL, *https://jogamp.org/jogl/www/*. Retrieved May 13, 2013.
[20] Java3D, *http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html*. Retrieved May 13, 2013.
[21] Unity. *http://unity3d.com/*. Retrieved May 13, 2013.
[22] Unreal GE. *http://www.unrealengine.com/*. Retr. May 13, 2013.
[23] Resource Description Framework (RDF). *http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/*. Retrieved May 17, 2013.
[24] Resource Description Framework Schema. *http://www.w3.org/TR/2000/CR-rdf-schema-20000327/*. Retrieved May 17, 2013.
[25] OWL Web Ontology Language Reference. *http://www.w3.org/TR/owl-ref/*. Retrieved May 17, 2013.
[26] SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *http://www.w3.org/Submission/SWRL/*. Retrieved May 17, 2013.
[27] SPARQL Query Language for RDF. *http://www.w3.org/TR/rdf-sparql-query/*. Retrieved May 17, 2013.
[28] Inf. tech. Mult. content desc. interface Part 10: Schema def. *http://www.chiariglione.org/ mpeg/working_documents/mpeg-07/schema_def/cd.zip*. Retrieved May 17, 2013.
[29] Walczak K., Chmielewski J., Stawniak M., Strykowski S., Extensible Metadata Framework for Describing Virtual Reality and Multimedia Contents. In: *Proc. of the 7th IASTED Int. Conference on Databases and Applications*, Innsbruck, Austria, February 14-16, 2006, pp. 168-175.
[30] Ont. for Media Resources 1.0. *http://www.w3.org/TR/mediaont-10/*. Retrieved May 17, 2013.
[31] Core Ont. for Mult.. *http://comm.semanticweb.org/*. Retrieved May 17, 2013.
[32] Arndt R., Troncy R., Staab S., Hardman L., Vacura M., COMM: designing a well-founded multimedia ontology for the web. In: *Proc. of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*, Busan, Korea, November 11-15, 2007, pp. 30-43.
[33] Otto K. A., Semantic Virtual Environments. In: *Special interest tracks and posters of the 14th international conference on World Wide Web*, Japan, May 10-14, 2005, pp. 1036-1037.
[34] Bilasco I. M., Villanova-Oliver M., Gensel J., Martin H., Semantic-based Rules for 3D Scene Adaptation. In: *Proc. of the twelfth int. conf. on 3D web technology*, Umbria, Italy, April 15-18, 2007, pp. 97-100.
[35] Bilasco I. M., Gensel J., Villanova-Oliver M., Martin H., On Indexing of 3D Scenes Using MPEG-7, In: *Proc. of the 13th annual ACM int. conf. on Multimedia*, Singapore, November 06-12, 2005, pp. 471-474.
[36] Bilasco I. M., Gensel J., Villanova-Oliver M., Martin H., 3DSEAM: a model for annotating 3D scenes using MPEG-7, In: *Proceedings of the eleventh international conference on 3D web technology*, Columbia, MD, USA, April 18-21, 2006, pp. 65-74.
[37] Pittarello F., Faveri A., Semantic description of 3D environments: a proposal based on web standards, In: *Proc. of the 11th Int. Conf. on 3D web Techn.*, Columbia, USA, April 18-21, 2006, pp. 85-95.
[38] Walczak K., Flex-VR: Configurable 3D Web Applications. In: *Proceedings of the International Conference on Human System Interaction HSI 2008*, Krakw, May 25-27, 2008, pp. 135-140, ISBN: 1-4244-1543-8.
[39] De Troyer O., Bille W., Romero R., Stuer P., On Generating Virtual Worlds from Domain Ontologies, In: *Proc. of the 9th Int. Conf. on Multi-Media Modeling*,Taipei,Taiwan, 2003, pp. 279-294, ISBN 957-9078-57-2.
[40] Bille W., Pellens B., Kleinermann F., De Troyer O., Intelligent Modelling of Virtual Worlds Using Domain Ontologies, In: *Proc. of the Workshop of Intelligent Computing (WIC), held in conjunction with the MICAI 2004 conference*,Mexico City,Mexico,2004, pp. 272-279, ISBN 968-489-024-9.
[41] De Troyer O., Kleinermann F., Pellens B., Bille W., Conceptual modeling for virtual reality, In: *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling - Vol. 83*, Australian Computer Society, Inc. Darlinghurst, Australia, 2007, pp. 3-18, ISBN: 978-1-920682-64-4.
[42] Walczak K., Beh-VR: Modeling Behavior of Dynamic Virtual Reality Contents. In: *Proc. of the 12th Int. Conf. on Virtual Systems and Multimedia VSMM 2006*, in: *H. Zha et al. (Eds.): VSMM 2006, LNCS 4270*, Springer Verlag Heidelberg 2006, pp. 40-51.
[43] Pellens B., De Troyer O., Bille W., Kleinermann F., Romero R., An Ontology-Driven Approach for Modeling Behavior in Virtual Environments, In: *Proceedings of Ontology Mining and Engineering and its Use for Virtual Reality (WOMEUVR 2005)*, Springer-Verlag, Agia Napa, Cyprus, 2005, pp. 1215-1224, ISBN 3-540-29739-1.
[44] Pellens B., De Troyer O., Bille W., Kleinermann F., Conceptual Modeling of Object Behavior in a Virtual Environment, In: *Proc. of Virtual Concept 2005*, Springer-Verlag, Biarritz, France, 2005, pp. 93-94.
[45] Pellens B., Kleinermann F., De Troyer O., A Development Environment using Behavior Patterns to Facilitate Building 3D/VR Applications, In: *In Proceedings of the 6th Australasian Conference on Interactive Entertainment*, ACM, Sydney,Australia 2009, ISBN 978-1-4503-0010-0.
[46] Zaid L. A., Kleinermann F., De Troyer O., Applying semantic web technology to feature modeling, In: *Proceedings of the 2009 ACM symposium on Applied Computing*, Honolulu, Hawaii, USA, 2009, pp. 1252-1256, ISBN: 978-1-60558-166-8.
[47] Protege, *http://protege.stanford.edu/*. Retrieved May 18, 2013.