Joint Virtual Reality Conference of EGVE - EuroVR (2013)
B. Mohler, B. Raffin, H. Saito, and O. Staadt (Editors)

Semantic Modelling of Interactive 3D Content

Jakub Flotyński[1], Krzysztof Walczak[1]

[1]Poznań University of Economics, Poland
{flotynski, walczak}@kti.ue.poznan.pl
http://www.kti.ue.poznan.pl

Abstract

Interactive three-dimensional content is the primary element of virtual reality (VR) and augmented reality (AR) systems. The increasing complexity and the use of VR/AR systems in various application domains requires efficient methods of creating, searching and combining interactive 3D content, which could be used by people with different specialities, who are not required to be IT-experts. The Semantic Web approach enables description of web resources with common semantic concepts. However, the use of semantic concepts may also facilitate creation of 3D content. The main contribution of this paper is a method of semantic modelling of interactive 3D content. The method leverages semantic constraints between different components of 3D content as well as representations of 3D content at different levels of abstraction. It can be used with a multitude of domain-specific ontologies and knowledge bases to simplify creating and searching of reusable semantic 3D content components and assembling complex 3D scenes from independent distributed elements.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information interfaces and presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1. Introduction

Widespread use of interactive 3D technologies and multimedia systems, including virtual reality (VR) and augmented reality (AR), has been recently enabled by the significant progress in hardware performance, the rapid growth in the available network bandwidth as well as the availability of versatile input-output devices. The primary element of VR/AR systems, apart from interface technologies, is interactive three-dimensional content. Dependencies between components of interactive 3D content may include, in addition to its basic meaning and presentation form, also spatial, temporal, structural, logical and behavioural aspects. Hence, creating, searching and combining distributed interactive 3D content are much more complex and challenging tasks than in the case of typical web resources.

The potential of VR/AR applications in everyday use can be fully exploited only if accompanied by efficient methods of creating, searching and combining distributed interactive 3D content, which would be easy to use by domain experts, who are not required to have advanced technical skills. However, currently creation of 3D content is accessible mainly to professional developers equipped with specific software tools (e.g., for 3D modelling) and hardware devices (e.g., for 3D scanning), and it is hardly accessible to non-IT-specialists. Several modelling tools, such as Autodesk 123d [Aut], 3DVIA shape [Das] and SketchUp [Tri], have been developed to simplify 3D content creation. Nevertheless, the process is still slow and laborious, and the tools refer to specific technical concepts, which may be incomprehensible to users without IT expertise. It is desirable that domain experts create advanced presentations on the basis of domain knowledge and concepts, without studying technical issues. Moreover, the available solutions provide only limited capabilities to search independent components that could be further reused and combined, with respect to their mutual semantic dependencies and relations, into complex 3D scenes.

The Semantic Web standards enable description of various types of web resources, such as typical web pages, images, audio, video and 3D content with commonly used concepts. However, the use of semantic concepts may also facilitate modelling of content at an arbitrarily chosen (high) level of abstraction, with regards to the meaning of particular objects reflected in the created VR/AR application as well as dependencies and relations between them. Referring to semantic domain-specific concepts, which may be abstract in

DOI: 10.2312/EGVE.JVRC13.041-048

the sense of their final presentation, can improve efficiency and reduce complexity of creation, dissemination and reuse of 3D content. New semantic methods of content creation can be used by domain experts in a variety of applications that need to share and reuse a number of common elements with respect to specific conditions and rules, e.g., engineering, e-commerce, cultural heritage and education.

The main contribution of this paper is an approach to semantic modelling of interactive 3D content. The proposed method is based on multi-layered semantic representation of 3D content, and it enables creation of 3D content at different levels of abstraction. The solution is intended to simplify designing 3D content by non-IT-specialists equipped with various domain-specific ontologies and knowledge bases. The method is platform- and standard-independent, but it conceptually complies with well-established 3D content and semantic description standards.

2. Related Works

Numerous works have been devoted to semantic modelling of 3D content. In [Ott05], an approach to designing interoperable RDF-based Semantic Virtual Environments, with system-independent and machine-readable abstract descriptions has been presented. In [BGVOM06, BVOGM07], a rule-based framework using MPEG-7 has been proposed for the adaptation of 3D content, e.g., geometry and texture degradation, and filtering of objects. The content can be described with different encoding formats (in particular X3D), and it is annotated with an indexing model. In [PF06], 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. In [KCM06], an ontology for X3D as well as semantic properties for coupling VR scenes with domain knowledge have been proposed. In [Wal08], a method of creating VR content on the basis on reusable elements with specific roles has been proposed to facilitate 3D content design by non-IT-specialists. In [BPKT04, TKPB07], an approach to generating virtual words upon mappings of domain ontologies to particular 3D content 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 content representation 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 modelling of behaviour of VR objects. In [Wal06], 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-IT-specialists. Another method facilitating modelling of content behaviour [PTBK05, PKT09] provides a means of expressing primitive and complex behaviours as well as temporal operators. Tool-supported design approach to defining object behaviour in X3D scenes has been presented in [PTK08]. Finally, a rule-based ontology framework for feature modelling and consistency checking has been explained in [ZKT09].

Other works have been devoted to the use of semantic descriptions of 3D content in artificial intelligence systems. The idea of semantic description of 3D worlds has been summarized in [LB08]. In [AL00], diverse issues arising from combining AI and virtual environments have been reviewed. In [CP00, LC07], abstract semantic representations of events and actions in AI simulators have been presented. In [LBW05, WL12, LF07], a technique of integration of knowledge into VR applications, a framework for decoupling components in real-time intelligent interactive systems with ontologies and a concept of semantic entities in VR applications have been discussed.

The aforementioned approaches address different aspects of semantic description and semantic creation of 3D content, but they lack general solutions for comprehensive modelling of 3D content, its components, properties and relations, at an arbitrarily high level of semantic abstraction, by domain experts.

3. Multi-layered Semantic Model of 3D Content

Although existing Semantic Web ontologies are sufficient for describing a multitude of types of real objects, they are generally insufficient for creating complex semantic representations of virtual objects (3D content). To permit accurate representation of 3D content, more comprehensive models enabling additional levels of detail and reflecting the different concepts used by different components of 3D content are required. For instance, stating that one car outruns another car is sufficient for a typical semantic analysis, but, to be presented, it requires an animation that changes the positions of both objects, which represent the cars.

In [FW13a, FW13b], a multi-layered semantic model of 3D content has been proposed. It has been designed as a set of ontologies that enable representation of 3D content at different levels of semantic abstraction—low-level *concrete semantic representation* of 3D content and arbitrarily high-level *conceptual semantic representation* of 3D content (Figure 1). Both representations are based on semantic concepts (classes and properties), which are used for defining semantic individuals, their properties and relations. The representations are linked by semantic mapping.

3.1. Concrete Semantic Representation of 3D Content

A concrete semantic representation of 3D content is a knowledge base compliant with the semantic model that has been proposed in [FW13b]. It introduces the separation of concerns between several layers corresponding to distinct aspects directly related to 3D content—*geometry layer*, *structure and space layer*, *appearance layer*, *scene layer*, *logic layer* and *behaviour layer*.

The layers include concepts (classes and properties), which are based on concepts commonly used in well-established 3D content representation languages and programming libraries, such as X3D, VRML and Away3D. The *geometry layer* introduces basic uniform individual geometrical components, e.g., planes and meshes. The *structure and space layer* introduces complex structural components, which assemble geometrical components, allowing for definition of spatial dependencies between them, e.g., position, orientation and size. The *appearance layer* adds appearance to geometrical and structural components, e.g., color, transparency and texture. The *scene layer* extends structural components to navigable scenes with viewpoints. The *logic* and *behaviour layers* enrich components that have been defined in the previous layers, with logic and behaviour, in particular animations and interactions. The subsequent layers are partly dependent—every layer uses its concepts and concepts specified in its lower layers.

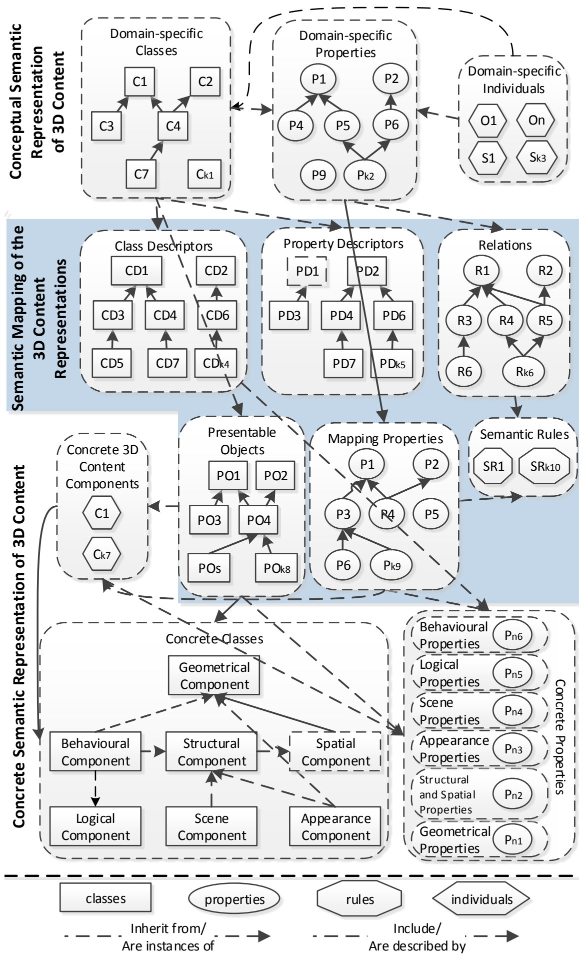

Figure 1: *Multi-layered representation of 3D content*

3.2. Conceptual Semantic Representation of 3D Content

A conceptual semantic representation of 3D content is a knowledge base compliant with domain-specific ontologies. Domain-specific concepts permit representation of 3D content at an arbitrarily chosen level of semantic abstraction. The concepts are abstract in the sense of their final presentation, as—in general—they can be presented in different manners (e.g., 2D graphics, 3D models, voice). The concepts do not need to cover any aspects that are typical to 3D content, or such aspects do not need to be indicated directly. For instance, an abstract (conceptual) car does not have to be specified as a particular 3D shape though it may be implicitly considered as such in terms of its final presentation, e.g., by belonging to a particular sub-class of cars (delivery van, limousine, etc.). Various dependencies, which are typical for Semantic Web standards (RDF, RDFS, OWL and SWRL [W3C]), may be defined for and between individual concepts, in particular multiple inheritance, restrictions on members of classes as well as domains and ranges of properties can be used.

3.3. Semantic Mapping of 3D Content Representations

A semantic mapping is a knowledge base that links a concrete representation to a conceptual representation of 3D content. The goal of mapping is to make the domain-specific concepts presentable. A mapping complies with the semantic mapping model, which has been explained in detail in [FW13a]. Each mapping assigns concrete semantic components and properties of 3D content to domain-specific concepts. Linking domain-specific classes and properties used in a conceptual (high-level) representation with particular components and properties of a concrete (low-level) representation improves efficient modelling and reusability of the domain-specific concepts in contrast to defining individual concrete representations for particular domain-specific objects and scenes (high-level semantic individuals). The following mapping concepts are distinguished: *presentable objects*, *mapping properties*, *descriptors* and *relations*.

Every class from a domain-specific ontology whose individuals are primary entities to be presented in the created scene, is specified as a *presentable object* (PO) class, e.g., artefacts in a virtual museum exhibition, avatars in an RPG game, etc. For each PO class, various concrete properties related to geometry, structure, space, appearance, scene, logic and behaviour can be specified.

POs may be described by various *mapping properties* (MPs), which indicate features of these objects, e.g., shape, color, texture, etc.

Descriptors are a functional extension of MPs, as they gather multiple properties to describe POs. Unlike POs, *descriptors* do not have individual 3D representations. *Class descriptors* (CDs) are assigned to objects as classes defining various properties, e.g., a class of interactive rotating objects includes POs that rotate after being touched. *Property descriptors* (PDs) are individuals that are linked to the described POs by properties. For example, furniture (described objects) can be made of (property) different types of wood (descriptive individuals–PD), each of which is described by a few properties such as color, texture, etc.

A *relation* (RL) is a domain-specific property that links different POs occurring in the created domain-specific scene. Every RL has at least two parts (participants), which are connected one to another by mutual dependencies related to some concrete properties of 3D content, such as geometry, structure, space, appearance, scene, logic or behaviour, e.g., a relation that defines the relative position between three POs indicates these objects and also specifies their relative orientations and distances between them.

The description of complex dependencies that occur in definitions of *mapping properties*, *descriptors* and *relations* is built in a declarative way by using auxiliary semantic rules, which are specified in a rule description language.

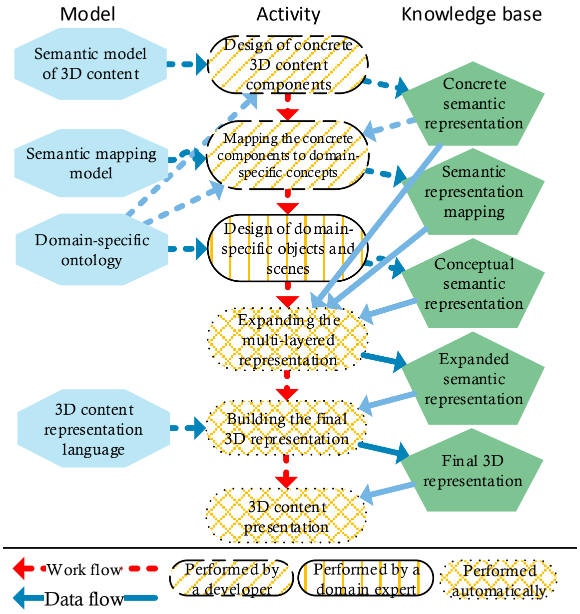

Figure 2: *Creation of 3D content based on the semantic modelling approach*

4. Method of Semantic Modelling of 3D Content

In this paper, a method of semantic modelling of interactive 3D content is proposed. The method enables flexible creation of 3D content at an arbitrarily chosen level of abstraction by leveraging the multi-layered semantic model, which has been explained in the previous section. Creation of 3D content with the presented approach is a sequence of steps, which are activities corresponding to different levels of semantic abstraction of the created content—design of a concrete semantic representation of 3D content, mapping the concrete representation to domain-specific concepts, design of a conceptual semantic representation of 3D content, expanding the multi-layered representation and building a final representation of 3D content (Figure 2). Consecutive activities depend on their preceding activities. The first three activities leverage the particular layers of the model presented and they are performed by a developer or a domain expert. These activities produce knowledge bases that conform to the parts of the model. The other two steps may be accomplished automatically. They precede the presentation of 3D content, which may be performed using different 3D content browsers and presentation tools.

The following sections describe subsequent activities of the modelling process. The activities may be performed using a typical semantic editor (e.g., Protégé), however, development of a specific 3D visual semantic modelling tool is also possible.

4.1. Design of a Concrete Representation of 3D Content

The design of a concrete semantic representation provides particular components of 3D content to enable presentation of domain-specific concepts that will be further used by a domain expert in the modelling process. This activity is typically performed by a developer (an IT-specialist with technical skills in 3D modelling) according to the semantic model of 3D content (presented in [FW13b] and in Sec. 3). The result of this modelling activity is a knowledge base including concrete semantic 3D content components.

In this activity, 3D content design may cover an arbitrary number of layers (starting from the bottom *geometrical layer*) of the 3D content model. For instance, the design of a complex 3D scene with behaviour involves components from all of the layers, while reusable 3D objects that are to be injected into different presentations may be designed at the *structural layer*.

Since the concrete components are to represent domain-specific individuals (that will be designed in activity 3), they must be created with respect to the domain-specific concepts that are defined in the selected domain-specific ontology, e.g., a particular 3D mesh represents cars of a particular domain-specific class.

4.2. Mapping the Representations of 3D Content

The second modelling activity—mapping of a concrete 3D content representation to domain-specific semantic concepts—enables 3D presentation of particular domain-specific concepts by linking them to particular concrete components, which have been designed in activity 1. The mapping does not preclude (and does not address) representations of domain-specific concepts by combinations of other domain-specific concepts. Such representations can be encoded during the design of a conceptual representation (activity 3) by typical means provided by the Semantic Web standards (such as class inheritance or restrictions on members of classes). The mapping is performed once for a particular domain-specific ontology and a concrete representation (a set of concrete components). Like a concrete representation, a mapping may be used for presenting various domain-specific individuals that conform to the domain-specific ontology selected. This may be done by a developer or a technician using a semantic modelling tool (depending on the number and complexity of desirable mapping links).

The following aspects of the mapping are distinguished. For each domain-specific class whose individuals need to have independent presentations in the final 3D scene, a PO class is created. This class is described by restrictions that indicate desirable concrete properties (with values) and concrete components, which are included in the concrete representation. If domain-specific classes form a hierarchy, the ascendant domain-specific classes are described first. Additional presentational effects (including behaviour), which are not covered by the ascendant classes, are described in their sub-classes.

Domain-specific classes, which may be assigned to POs to specify their features, but which do not identify independent entities to be presented, are expressed as CDs. Like POs, CDs may be described with desirable concrete properties.

Domain-specific data properties are specified as MPs, which indicate target concrete properties. If a dependency between a conceptual and a concrete property is not trivial (equivalent), an auxiliary semantic rule must be given. Like for POs, the mapping starts with ascendant properties and only these domain-specific sub-properties that introduce additional presentational effects (in comparison to their super-properties) are additionally described.

Domain-specific classes whose individuals determine a number of features of POs, and which are linked to POs by domain-specific object properties, are specified as PDs with optional auxiliary semantic rules.

Domain-specific object properties whose domains and ranges are POs, and domain-specific classes that are not POs, but refer to multiple POs by properties, are specified as RLs. Auxiliary semantic rules of RLs describe the dependencies between the parts of the RLs.

4.3. Design of a Conceptual Representation of 3D Content

The design of a conceptual semantic representation enables creation of 3D content at a high level of abstraction. This activity can be performed many times for a particular domain-specific ontology, a concrete representation and a mapping. This is typically performed by a domain expert (a non IT-specialist) every time when new 3D content is required for a particular specific 3D/VR/AR application. A domain expert uses a domain-specific ontology to focus only on high-level semantic concepts and does not touch low-level concrete components of 3D content, which can be confusing for people without advanced technical skills, and which are hidden from the expert behind the mapping.

In general, this modelling activity is independent of the activities described previously, and a conceptual representation may be created before a concrete representation and a mapping are created, e.g., when a domain expert designs an accurate digital equivalent to a known fragment of the reality. However, when designing non-existing objects or scenes (e.g., a planned virtual museum exhibition) the availability of the concrete representation and the mapping may be desirable to enable preview of the results during the modelling.

The result of this activity is a knowledge base that complies with the domain-specific ontology and includes the required domain-specific individuals. A domain-specific scene incorporates various domain-specific objects, which can be described and linked by domain-specific properties and relations. In this modelling activity, 3D content may be created at such levels of semantic abstraction that are permitted by the domain-specific ontology used.

4.4. Expanding the Representation of 3D Content

The three previous activities of the modelling process provide a comprehensive multi-layered semantic representation of 3D content at different levels of semantic abstraction. This representation covers all modelling elements that must be resolved by a human. The remaining activities of the content creation process may be completed automatically.

The concrete semantic components of 3D content (designed in activity 1) have been assigned to domain-specific concepts, but they are not linked to domain-specific individuals (designed in activity 3). To enable presentation of the domain-specific individuals, the multi-layered representation is expanded with respect to the mapping (designed in activity 2), and the concrete components and properties are linked to the domain-specific individuals (designed in activity 3) in the following steps.

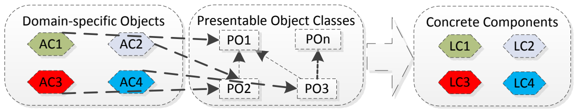

Figure 3: *Expanding the multi-layered representation*

1. For every instance of a PO class, a new concrete (*geometrical* or *structural*) component is created according to the definition of this PO class and its particular concrete properties are assigned (Figure 3).
2. For every PO, each MP that is assigned to it directly or by a CD or a PD, is converted to its corresponding concrete property (a set of concrete properties) according to the auxiliary semantic rules associated with this MP. The resulting concrete properties are assigned to the concrete component that primarily represents the PO.
3. For every RL, concrete properties and components that are specified in its auxiliary semantic rules are directly assigned to the concrete components that primarily represent the parts (POs) of the RL.

As the result of the above steps, every domain-specific individual is represented by a concrete component and with a set of concrete properties.

4.5. Building a Final Representation of 3D Content

The last step in the content creation process is a transformation of the concrete semantic components created in the expanding activity, to their final 3D counterparts encoded in a particular 3D content representation language. This part of the content creation process can be performed automatically with a possible support of an ontology that describes the target 3D content representation language and a mapping ontology that links the concrete concepts with their corresponding final counterparts, which are defined in the target language. The transformation can cover a wide range of target presentation platforms, as proposed in [FDW12].

5. Semantic Modelling of an Example 3D Scene

In this section, an example of the semantic modelling of a 3D scene is explained. The example demonstrates the consecutive activities of the modelling process that are performed by a human, with the focus on the main goal of the proposed approach—an easy-to-use method of creating 3D content by non-IT-specialists. The scene is described with the Semantic Web standards: RDF, RDFS, OWL and SWRL.

The example scene presents a room, e.g., for interior design systems (Figure 4). In the scene, 3D models of a table, a chair and a flowerpot have been used [Blo]. The domain-specific ontology that is selected for semantic modelling of the scene, defines classes of individuals (table, chair, flowerpot, decoration and room) as well as properties and relations for describing these individuals.

In the presented example, all modelling activities have been performed using the Protégé semantic editor. However, specific graphical tools can be developed to support visual modelling in particular activities of the modelling process. Such tools can be implemented, e.g., as plug-ins to widely used modelling tools, such as Blender or 3ds Max.

Figure 4: *An example interactive 3D scene*

Listing 1: *The concrete representation of the scene*

```
1   cr:Chair rdfs:subClassOf cr:Mesh3D
2     [ rdf:type owl:Restriction ; owl:onProperty cr:
          coordinates ; owl:hasValue "..." ].
3
4   cr:woodSide rdf:type cr:TextureSide ;
5     cr:texture "wood_tex.jpg".
6
7   cr:orientationAnimation rdf:type cr:InterpolatedAnimation.
8   cr:orientationAnimation cr:timePoint "0␣2␣4" ;
9     cr:value "0␣0␣11␣0␣0␣20␣0␣0␣11" ;
10    cr:enabled "false".
11
12  cr:orientationActivity rdf:type cr:Activity.
13  cr:orientationActivity: cr:clicked(?x), cr:animation(?x,?
        anim) -> cr:enabled(?anim,"true").
```

The first activity of the modelling is the design of concrete semantic components of 3D content, which is performed by a developer with technical skills in 3D modelling. To provide a 3D representation of domain-specific concepts, the following components are created (Listing 1, the `cr` prefix). The `cr:Chair` is a subclass of meshes (specific geometrical components) with the given coordinates (1-2). The `cr:woodSide` indicates a texture for wooden objects (4-5). The `cr:orientationAnimation` component rotates an object to present it (7-10). It is activated by the `cr:orientationActivity` component when the associated object is touched (12-13). Other representational concrete components are skipped in this example, as they are created analogously.

The second activity of the modelling is mapping the concrete components and properties to appropriate domain-specific concepts (classes and properties), which is performed by a developer or a technician. To enable 3D presentation of a domain-specific scene (which will be designed by a domain expert), the following mapping concepts are created (Listing 2, the `mp` prefix) to link the concrete components to a conceptual representation (the `cp` prefix). Since the individuals of the domain-specific `cp:Chair` class need to have independent representations in the final 3D scene, the mapping `mp:Chair` class is created and it is specified as a super-class of the `cp:Chair`, as a sub-class of the `mp:PresentableObject` and as a sub-class of the `cr:Chair` class (1-2), which specifies the required coordinates of the objects.

Listing 2: *The mapping of the scene representations*

```
cp:Chair rdfs:subClassOf mp:Chair.                                  1
mp:Chair rdfs:subClassOf mp:PresentableObject , cr:Chair            2
                                                                    3
cp:material rdfs:subPropertyOf mp:material.                         4
mp:material rdfs:subPropertyOf mp:MappingProperty ;                 5
  mp:rule mp:material-rule1 ;                                       6
  mp:rule mp:material-rule2 ;                                       7
  mp:rule mp:material-rule3.                                        8
mp:material-rule1: mp:material(?x,"metal") -> cr:color(?x,"         9
     0.4␣0.5␣0.6"), cr:specularColor(?x,"0.8␣0.7␣0.5").
mp:material-rule2: mp:material(?x,"wood") -> cr:firstSide          10
     (?x,cr:woodSide), cr:specularColor(?x,"0.2␣0.2␣0.2").
mp:material-rule2: mp:material(?x,"leather") -> cr:color(?x        11
     ,"0.7␣0.7␣0.45").
                                                                   12
cp:RotatingObject rdfs:subClassOf mp:RotatingObject.               13
mp:RotatingObject rdfs:subClassOf mp:ClassDescriptor ,             14
  [ rdf:type owl:Restriction ; owl:onProperty cr:action ;          15
       owl:hasValue cr:orientationActivity ] ;
  [ rdf:type owl:Restriction ; owl:onProperty cr:animation         16
       ; owl:hasValue cr:orientationAnimation ].
                                                                   17
cp:AppearanceDesc rdfs:subClassOf mp:AppearanceDesc.               18
mp:AppearanceDesc rdfs:subClassOf mp:PropertyDescriptor.           19
cp:shininess owl:equivalentProperty cr:shininess.                  20
cp:appearance rdfs:subPropertyOf mp:appearance.                    21
mp:appearance rdfs:subPropertyOf mp:DescriptorProperty.            22
                                                                   23
cp:standsOn rdfs:subPropertyOf mp:standsOn.                        24
mp:standsOn rdfs:subPropertyOf mp:Relation.                        25
mp:standsOn mp:rule mp:standsOn-rule.                              26
mp:standsOn-rule: mp:standsOn(?A,?B), cr:position(?B,?BPos)        27
     , cr:size(?B,?BSize), cr:position(?A,?APos), cr:size(?
     A,?ASize) -> swrlb:equal(?APos[Z],?BPos[Z]+0.5*?BSize[
     Z]+0.5*?ASize[Z]), swrlb:greaterThan(?APos[X,Y],?BPos[
     X,Y]), swrlb:lessThan(?APos[X,Y],?BPos[X,Y]+?BSize[X,Y
     ]).
                                                                   28
cp:standsInTheMiddleOf rdfs:subPropertyOf mp:                      29
     standsInTheMiddleOf.
mp:standsInTheMiddleOf rdfs:subPropertyOf mp:Relation.             30
mp:standsInTheMiddleOf mp:rule mp:standsInTheMiddleOf-rule.        31
mp:standsInTheMiddleOf-rule: mp:standsInTheMiddleOf(?A,?B),        32
     cr:position(?B,?BPos), cr:size(?B,?BSize), cr:
     position(?A,?APos) -> swrlb:equal(?APos[X,Y],(?BPos[X,
     Y]+?BSize[X,Y]) / 2).
```

The domain-specific `cp:material` property determines the material of objects, thus influencing their appearance. It becomes a sub-property of the `mp:material` MP (4-5). The three associated auxiliary semantic rules allow for the specification of a desirable color or texture of the primary described object (6-11). The domain-specific `cp:RotatingObject` class indicates objects that perform a rotation after being touched. It is determined as a CD (13-14) that defines interactivity and animation for all its individuals (15-16). The domain-specific `cp:appearance` property and the domain-specific `cp:AppearanceDesc` class will be used by a domain expert for describing appearance of objects. The `cp:appearance` property is defined as a sub-property of a *descriptor property* (21-22) whose range is the `mp:AppearanceDesc` PD (18-19). This PD contains the two properties—the `cp:material` and the `cp:shininess`. While the first one has already been defined, the second one requires a mapping. It is specified as an equivalent to the concrete `cr:shininess` property (20). Although the properties are assigned to an instance of the descriptor, they always influence the described PO. Finally, the domain-specific `cp:standOn` property, which describes the relative position of objects, is defined as an RL that is described by a semantic rule (24-27)—if an object A stands on an object B, the *X*, *Y* and *Z* coordinates of A are set with regard to the coordinates and the size of A and B. For the descendant domain-specific `cp:standsInTheMiddleOf` property an additional relational condition is given—the A object is placed exactly in the middle of B (29-32). The mapping of the other domain-specific concepts is omitted, as it is created analogously.

Listing 3: *The conceptual representation of the scene*

```
1  cp:room rdf:type cp:Room.
2  cp:floor rdf:type cp:Floor.
3  cp:table rdf:type cp:Table.
4  cp:chair1 rdf:type cp:Chair.
5  cp:chair2 rdf:type cp:Chair.
6  cp:flowerpot rdf:type cp:Flowerpot.
7  cp:decoration rdf:type cp:Decoration.
8  cp:table cp:material "wood".
9  cp:decoration cp:material "metal".
10 cp:flowerpot rdf:type cp:RotatingObject.
11 cp:chairAppearance rdf:type cp:AppearanceDesc.
12 cp:chair2 cp:appearance cp:chairAppearance.
13 cp:chairAppearance cp:shininess "0.2".
14 cp:chairAppearance cp:material "leather".
15 cp:room cp:hasFloor cp:floor.
16 cp:decoration cp:standsOn cp:table.
17 cp:standsInTheMiddleOf rdfs:subPropertyOf cp:standsOn.
18 cp:flowerpot cp:standsInTheMiddleOf cp:table.
```

Although the conceptual modelling is independent of the previous activities, and it could be performed first, in the presented example it is assumed that the domain-expert needs to preview the results during the modelling, so it becomes the last activity of the modelling process. The conceptual representation (Listing 3, the `cp` prefix) is created with a domain-specific ontology. This representation does not cover any aspects that are directly related to 3D content. Such aspects have been addressed by a developer in activities 1 and 2. Lines 1-7 are used to create the scene (a room) and some objects, which are incorporated in it. The `cp:material` property (8) determines the material of the `cp:table`. The `cp:flowerpot` is specified as an `cp:RotatingObject` (10). Appearance of the `cp:chair2` object is specified in lines 11-14. The `cp:decoration` may be placed in an arbitrary point on the `cp:table` as it is described by the generalized `cp:standsOn` property (16). The `cp:flowerpot` stands in the middle of the `cp:table`, as indicated by the sub-property (17-18).

In the presented example, the last two activities of the modelling process—expanding the representation and building its final equivalent—have been performed manually. However, a tool for automatic conversion is currently being developed.

6. Conclusions and Future Works

The approach proposed in this paper has several important advantages in comparison to the available solutions for modelling 3D content. First, it can significantly reduce design effort when 3D content is formed based on reusable components, which need to be designed and mapped only once and can be then frequently reused. The conformance to well-established Semantic Web standards and tools improves the interoperability of the approach, simplifies the dissemination and reuse of semantic objects and scenes in different applications, and fosters domain-specific ontologies and knowledge bases leading to common repositories of open linked data. In addition, referring to the domain-specific meaning of the particular objects may improve the possibilities of creating, searching, exploring and reasoning on 3D content by domain experts who no longer need to go into details specific to diverse aspects of 3D content. Semantic description of properties and relations of the modelled objects can define applicability and compatibility between different objects, which are used in the designed scene. Moreover, since the representation of 3D content includes multiple layers, the presented approach facilitates the creation and analysis of 3D content at different (arbitrarily chosen) levels of semantic abstraction. Finally, 3D content described by commonly used concepts is platform- and standard-independent, and it may be transformed to final 3D representations encoded in different languages, depending on, e.g., the context of interaction, user preferences as well as hardware and software used.

However, the proposed approach has also some limitations. First, it is inconvenient for representing simple objects and scenes that are neither reused nor shared, and that are accessed mainly by authors, who know their semantics. In such cases, referring to the high-level semantics of the modelled content is not necessary. Second, the solution can be improper for building 3D objects and scenes that have no common contexts of presentation, which require completely disjoint representations of the components. In such cases, the modelling effort to create different semantic representations is not beneficial. Moreover, designing 3D content with the presented method requires explicit specification of all the objects, their properties and relations that need to be presented in the resulting scene. Methods of dynamic semantic composition and presentation of 3D content may be proposed to permit implicit conditional query-based assembly of 3D scenes from different reusable components.

Possible directions of future research incorporate several facets. First, the method needs to be fully implemented including the automatic expanding of multi-layered representations and building final 3D representations (encoded in, e.g., VRML, X3D, Java3D, etc.). Second, the implementation should be evaluated and compared to other platforms in terms of the simplicity and efficiency of 3D content creation. Third, the encoding of auxiliary semantic mapping rules is verbose and inconvenient for specifying complex dependencies. A specific rule description language can be elaborated to overcome this limitation and to provide a flexible solution for semantic modelling of complex behavioural VR scenarios. Next, a visual semantic modelling tool supporting the proposed method could be developed. Finally, the presented example of the semantic modelling assumes a unidirectional transformation of the conceptual scene to its final 3D representation. To synchronize a semantic representation with its final equivalent and to permit semantic management and exploration of 3D content in real-time, a persistent link between the semantic individuals and the components of the generated final 3D representation should be maintained.

7. Acknowledgements

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

References

[AL00] AYLETT R., LUCK M.: Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence 14* (2000), 3–32. 2

[Aut] AUTODESK:. URL: http://www.123dapp.com/. 1

[BGVOM06] BILASCO I. M., GENSEL J., VILLANOVA-OLIVER M., MARTIN H.: 3dseam: a model for annotating 3d scenes using mpeg-7. In *Proc. of the 11th int. conference on 3D web technology* (Columbia, MD, USA, April 18-21, 2006), pp. 65–74. 2

[Blo] BLOGSCOPIA: *URL: http://resources.blogscopia.com/2011 [/05/19/bar-table-and-chairs], [2010/05/11/armchair-2-download], [2010/04/22/flowerpot]*. 6

[BPKT04] BILLE W., PELLENS B., KLEINERMANN F., TROYER O. D.: 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. 2

[BVOGM07] BILASCO I. M., VILLANOVA-OLIVER M., GENSEL J., MARTIN H.: Semantic-based rules for 3d scene adaptation. In *Proc. of the 12th int. conference on 3D Web technology* (Umbria, Italy, April 15-18, 2007), pp. 97–100. 2

[CP00] CAVAZZA M., PALMER I.: High-level interpretation in virtual environments. *Applied AI 14* (2000), 125–144. 2

[Das] DASSAULT SYSTEMES:. URL: http://www.3dvia.com/products/3dvia-shape/. 1

[FDW12] FLOTYŃSKI J., DALKOWSKI J., WALCZAK K.: Building multi-platform 3d virtual museum exhibitions with flex-vr. In *The 18th International Conference on Virtual Systems and Multimedia* (Milan, Italy, September 2-5, 2012), pp. 391–398. 5

[FW13a] FLOTYŃSKI J., WALCZAK K.: Conceptual semantic representation of 3d content. *Lecture Notes in Business Information Processing: 16th Int. Conf. on Business Information Systems, Poznań, Poland, 19 - 20 June, 2013 160* (2013), 244–257. 2, 3

[FW13b] FLOTYŃSKI J., WALCZAK K.: Semantic multi-layered design of interactive 3d presentations. In *Proc. of the Federated Conf. on Computer Science and Information Systems* (Kraków, Poland, September 8-11, 2013), IEEE, pp. 541–548. 2, 4

[KCM06] KALOGERAKIS E., CHRISTODOULAKIS S., MOUMOUTZIS N.: Coupling ontologies with graphics content for knowledge driven visualization. In *VR '06 Proceedings of the IEEE conference on Virtual Reality* (Alexandria, Virginia, USA, March 25-29, 2006), pp. 43–50. 2

[LB08] LATOSCHIK M. E., BLACH R.: Semantic modelling for virtual worlds - a novel paradigm for realtime interactive systems? In *Proc. of the 2008 ACM symp. on VR software and technology* (Bordeaux, France, October 27-29, 2008), pp. 17–20. 2

[LBW05] LATOSCHIK M. E., BIERMANN P., WACHSMUTH I.: Knowledge in the loop: Semantics representation for multimodal simulative environments. In *Lecture Notes in Computer Science* (Springer Berlin Heidelberg, 2005), pp. 25–39. 2

[LC07] LUGRIN J., CAVAZZA M.: Making sense of virtual environments: action representation, grounding and common sense. In *Proc. of the 12th int. conference on Intelligent user interfaces* (Honolulu, HI, USA, January 28-31, 2007), pp. 225–234. 2

[LF07] LATOSCHIK M. E., FROHLICH C.: Semantic reflection for intelligent virtual environments. In *IEEE Virtual Reality Conference 2007* (Charlotte, USA, March 10-14, 2007). 2

[Ott05] 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. 2

[PF06] 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, MD, USA, April 18-21, 2006), pp. 85–95. 2

[PKT09] PELLENS B., KLEINERMANN F., TROYER O. D.: A development environment using behavior patterns to facilitate building 3d/vr applications. In *Proc. of the 6th Australasian Conf. on Interactive Entertainment* (Sydney, Australia, 2009), ACM. 2

[PTBK05] PELLENS B., TROYER O. D., BILLE W., KLEINERMANN F.: Conceptual modeling of object behavior in a virtual environment. In *Proceedings of Virtual Concept 2005* (Biarritz, France, 2005), Springer-Verlag, pp. 93–94. 2

[PTK08] PELLENS B., TROYER O. D., KLEINERMANN F.: Codepa: a conceptual design pattern approach to model behavior for x3d worlds. In *Proc. of the 13th int. symp. on 3D web technology* (Los Angeles, August 09-10, 2008), pp. 91–99. 2

[TKPB07] TROYER O. D., KLEINERMANN F., PELLENS B., BILLE W.: Conceptual modeling for virtual reality. In *Tutorials, posters, panels and industrial contributions at the 26th int. conf. on Conceptual modeling* (Darlinghurst, 2007), pp. 3–18. 2

[Tri] TRIMBLE:. URL: http://www.sketchup.com/. 1

[W3C] W3C: *World Wide Web Consortium*. URL: http://www.w3.org. 3

[Wal06] 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* (2006), Lecture Notes in Computer Sciences, Springer Verlag Heidelberg, pp. 40–51. 2

[Wal08] WALCZAK K.: Flex-vr: Configurable 3d web applications. In *Proc. of the Int. Conf. on Human System Interaction HSI 2008* (Kraków, Poland, May 25-27, 2008), pp. 135–140. 2

[WL12] WIEBUSCH D., LATOSCHIK M. E.: Enhanced decoupling of components in intelligent realtime interactive systems using ontologies. In *2012 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)* (Orange County, CA, USA, 2012), pp. 43–51. 2

[ZKT09] ZAID L. A., KLEINERMANN F., TROYER O. D.: Applying semantic web technology to feature modeling. In *Proceedings of the 2009 ACM symposium on Applied Computing* (Honolulu, Hawaii, USA, 2009), pp. 1252–1256. 2