Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 35 (2014) 531 – 540
Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

# Semantic modelling of interactive 3D content with domain-specific ontologies

Jakub Flotyński

*Poznań University of Economics, Niepodległości 10, 61-875 Poznań, Poland*

**Abstract**

The creation of interactive 3D presentations is typically a complex process involving activities related to various aspects of the content such as geometry, structure, space, appearance, logic and behaviour. However, widespread dissemination of interactive 3D content on the web requires flexible and efficient methods of content creation. In this paper, an approach to semantic modelling of 3D content is proposed. The proposed solution enables creation of content components and properties - reflecting different aspects of the content - with domain-specific ontologies and knowledge bases. The use of domain-specific knowledge liberates authors from going into details that are specific to 3D modelling, allows for content representation at different levels of abstraction and permits content creation by domain experts, who are not required to be IT-professionals.

Peer-review under responsibility of KES International.
*Keywords:* semantic web; semantic modelling; 3D content; domain-specific ontologies; virtual and augmented reality

## 1. Introduction

Widespread use of interactive 3D technologies in virtual (VR) and augmented (AR) reality applications 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. VR/AR applications become increasingly popular in various domains, such as eduction, training, tourism, medicine, entertainment, social media and cultural heritage, significantly enhancing possibilities of presentation and interaction with complex data and objects. The primary element of VR/AR applications, apart from the interface technologies, is interactive 3D 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 interactive 3D content are much more complex and challenging tasks than in the case of typical web resources.

The potential of VR/AR applications accessible on the web can be fully exploited only if 3D content presentation techniques are accompanied by efficient methods of content creation. A number of solutions have been devised for 3D content creation, including declarative programming languages (e.g., VRML[1], X3D[2] and COLLADA[3]), imperative programming languages (e.g., Java and ActionScript) with programming libraries (e.g., Java3D[4] and Away3D[5]) as

* Corresponding author.
*E-mail address:* flotynski@kti.ue.poznan.pl


Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.134

well as visual environments. Advanced visual environments, which are intended for professional users (e.g., Blender[6] and 3ds Max[7]) offer rich capabilities for designing various content elements, but their complexity requires author's expertise in 3D modelling. User-friendly visual environments (e.g., SketchUp[8] and 3DVIA[9]), which have been designed for domain experts (e.g., architects, engineers and interior designers), provide tools for relatively fast and efficient modelling, without requiring much users' experience in content creation, but they narrow the domain of application and the set of available operations.

The available approaches to 3D content creation have some important limitations, as they are oriented on the modelling of content instead of the modelling of knowledge. First, they demand users' knowledge of issues related to computer graphics and, therefore, they are difficult to use for domain experts, who are not required to be IT-professionals. Second, they require the modelling of all content details to be presented, and they do not support content parametrization and discovery of hidden knowledge that is relevant to the desirable presentational effects, which could reduce the effort required for content creation. Finally, they have not been intended to facilitate content exploration with common concepts and the use of content in content repositories, which is one of the key issues for widespread dissemination of content on the web.

The main contribution of this paper is a method of mapping 3D content representations to domain-specific ontologies. The method is a key part of the generic approach to Semantic Modelling of Interactive 3D Content (SEMIC), which has been partially described in the previous works[10,11,12,13]. The method enables mapping of components and properties, which are specific to 3D content, to concepts included in domain-specific ontologies, which may be abstract in the sense of their presentation. Hence, the method enables conceptual content creation at different levels of abstraction, which may be determined by common, reusable concepts. The conformance to the semantic web standards and the possible use of various domain-specific ontologies and knowledge-bases can facilitate content creation and management (indexing, searching and analysing) by domain-experts in diverse application domains.

## 2. Related works

Several works have been devoted to semantic creation and description of 3D content. In[14], an ontology providing elements and properties that are equivalent to elements and properties specified in X3D has been proposed. Moreover, a set of semantic properties have been proposed to enable description of 3D scenes with domain knowledge. In[15,16,17,18], an approach to creating parametrized VR content based on reusable elements with specific roles has been proposed. The approach has been intended to enable 3D content design by non-IT-specialists. In[19,20], an approach to generating virtual environments upon mappings of domain ontologies has been proposed. The solution stresses modelling of spatial relations between objects in the scene. In[21], a semantic model of virtual environments based on the MPEG-7 and MPEG-21 standards has been proposed to enable dynamic scaling and adapting the geometry and functions of virtual objects. In[22], an approach facilitating modelling of content behaviour with temporal operators has been proposed. In[23,24,25,26], an approach to building semantic descriptions embedded in 3D web content and a method of harvesting semantic metadata from 3D web content have been proposed.

Several works provide an overview of the use of semantic descriptions of 3D content in artificial intelligence systems. The idea of semantic description of 3D worlds has been summarized in[27]. In[28], a review of the main aspects related to the use of 3D content in connection with the semantic web technologies has been provided. In[29], diverse issues arising from combining AI and virtual environments have been reviewed. In[30], abstract semantic representations of events and actions in AI simulators have been presented. In[31,32], a technique of incorporating knowledge in 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.

## 3. The SEMIC approach

Although several approaches have been proposed for semantic modelling of 3D content, they lack general and comprehensive solutions for content creation with various domain-specific ontologies and knowledge bases, which are flexible and efficient in use by domain experts. Recent trends in the development of the web provide new opportunities for efficient and flexible 3D content creation, which go beyond the current state of the art by enabling: declarative knowledge-based creation of generalized content representations with discovery of hidden knowledge that influences the created content, conceptual modelling of content with different ontologies, which represent the content at different levels of abstraction, separation of concerns in content creation between different modelling users with different skills and equipped with different modelling tools as well as multi-platform 3D content presentation. These aspects of modelling content have been discussed in [10,11,12,13]. This paper addresses content creation with different domain-specific ontologies. In comparison to the available approaches, opportunities for modelling 3D content with domain-specific ontologies can be extended by:

1) the use of both concepts that are specific to 3D content and concepts that are not specific to 3D content (abstract in the sense of final presentation),
2) the use of hierarchies of classes and properties,
3) the reflection of complex 3D content features by classes, properties, individuals and rules,
4) the reflection of complex 3D content features by combinations of classes, combinations of properties and combinations of individuals,
5) the reflection of relations (mutual dependencies) between content objects.

This section provides an outline of the approach to Semantic Modelling of Interactive 3D Content (SEMIC)—proposed in [11], which enables 3D content creation with domain-specific ontologies.

### *3.1. Separation of concerns in content creation*

SEMIC supports separation of concerns between users with different responsibilities and capabilities. Creation of 3D content consists of a sequence of partly dependent steps, which use different content models and modelling tools, and produce content representations compliant with the content models. Some of the steps are performed by a content developer, a domain expert and a content consumer, while the other are performed automatically—by specific software. The steps are outlined below.

Step 1—the design of a concrete semantic representation of 3D content (CrR) provides particular elements of 3D content (content components and content properties) to enable representation of domain-specific concepts (classes and properties) that will be further used in Step 3. The result of this step is a knowledge base compliant with the Multi-layered Semantic Content Model (ML-SCM – proposed in [12]). A CrR incorporates concrete semantic components and properties, which are specific to 3D content (cf. Sec. 3/1), e.g., meshes, groups of objects, materials, viewpoints, etc. Hence, this step is typically performed by a developer with expertise in 3D modelling, who is equipped with specific tools, e.g., 2D or 3D graphical editors for creating textures or meshes.

Step 2—mapping a CrR (created in Step 1) to domain-specific semantic concepts, which is accomplished using the mapping method proposed in the next section, enables 3D presentation of domain-specific knowledge bases, which are created in Step 3. The result of this step is a representation mapping (RM), which is a knowledge base compliant with the Semantic Mapping Model (SMM – proposed in [10]). Mapping is performed once for a particular domain-specific ontology and a CrR, and it permits the reuse of concrete components and concrete properties for forming 3D representations of various domain-specific individuals (which conform to the domain-specific ontology). Since mapping covers technical aspects related to ontologies and knowledge bases, which are firmly based on the semantic web standards (RDF, RDFS and OWL), this step is typically performed by a developer or a technician with skills in semantic modelling, who is equipped with appropriate semantic tools.

Step 3—the design of a conceptual semantic representation of 3D content (CpR) enables creation of 3D content at an arbitrarily chosen level of abstraction, which is determined by the domain-specific ontology used (cf. Sec. 3/1). The result of this step is a knowledge base compliant with the domain-specific ontology. This step can be performed many times for a particular domain-specific ontology, a CrR and an RM. Since this step requires the knowledge of particular domain-specific concepts, it is typically performed by a domain expert, who is equipped with a semantic modelling tool.

The other steps of content creation with SEMIC have been discussed in [10,11,12,13], and they include semantic content customization, discovery of hidden knowledge in the designed CrRs and CpRs and building final 3D content representations, which may be presented using multiple content browsers and presentation tools.

### *3.2. The semantic content model*

Interactive 3D content designed with SEMIC conforms to ML-SCM, SMM and a domain-specific ontology, which overall are incorporated in the Semantic Content Model (SCM—cf. [10]). SCM specifies content structure, which strongly affects the mapping of concrete content components and properties to domain-specific concepts, which is proposed in this paper. The general scheme of a 3D content representation, which is compliant with SCM is depicted in Fig. 1a.

The following elements (*mapping concepts*) of 3D content are incorporated in SCM to enable linking concrete content components and properties to domain-specific concepts: *presentable objects* (POs), *data properties* (DPs) with *literals*, *object properties* (OPs) with *descriptive individuals* (DIs), *descriptive classes* (DCs) and *relations* (RLs). POs are semantic individuals that belong to PO classes (PO-Cs), and which are the primary elements of the designed content, having independent representations (specific to the modality of content presentation), e.g., artefacts in a virtual museum exhibition, avatars in an RPG game, UI controls in a visual interface, sounds in an aural interface, etc. Every PO-C determines some properties that are inextricable from its POs—form the POs and give a sense of them in the selected presentation modality, e.g., the colour map of a picture, the geometry of a 3D shape, the structure of a complex object, the sampling frequency of a sound, etc.

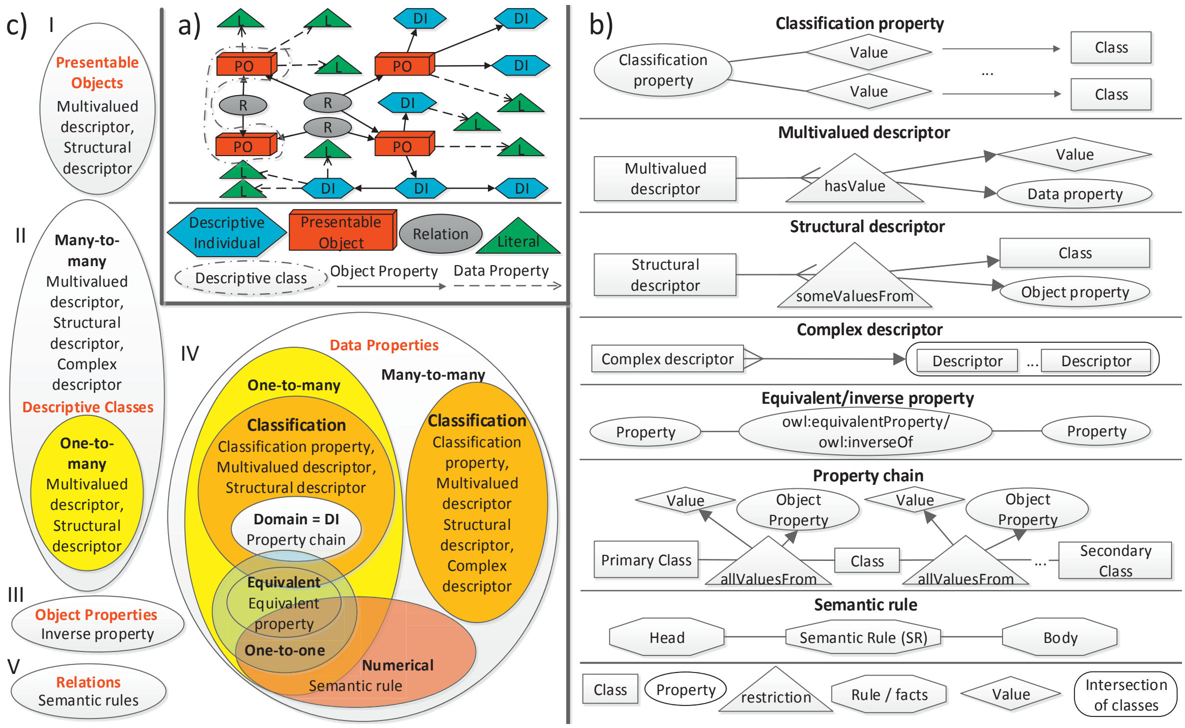


Fig. 1. The scheme of a 3D content representation compliant with the Semantic Content Model (a), *mapping patterns* used for linking concrete content elements to domain-specific concepts (b) and *mapping patterns* required for creating particular *mapping concepts* (c)

In addition to inherent properties of POs specified in PO-Cs, POs may be described by DPs and OPs, which determine their additional features. While DPs specify simple features, which may be expressed using single literals (e.g., scale, colour, coordinates, etc.), OPs specify complex features, which are expressed using DIs, which may aggregate multiple literals and other DIs (e.g., the material of a PO may be reflected by an individual aggregating literals that determine colour, transparency and shininess).

Furthermore, POs may be assigned to DCs, which determine DPs and OBs of the POs. In contrast to PO-Cs, DCs do not represent objects that have independent representation in the created content, but like DIs, they may aggregate multiple literals and DIs, which describe POs.

Finally, different POs can be combined using *relations* (RLs). Every RL links at least two participants (POs), which are connected one to another by mutual dependencies related to some DPs or OPs that determine presentable effects of the RL, e.g., a relation that determines the relative position of some POs links these POs and specifies their relative orientations and distances between them. RLs that link two POs may be encoded using OPs (binary RLs), while RLs that link more than two POs are encoded using individuals that link the POs by OPs (n-ary RLs).

## 4. Method of mapping 3D content representations

The method, which is proposed in this paper, is used to accomplish Step 2 of the SEMIC approach—mapping a CrR to domain-specific concepts. The result of mapping (an RM) enables 3D presentation of domain-specific knowledge bases (created in Step 3) by concrete components and concrete properties of 3D content included in the CrR (created in Step 1). An RM needs to cover all concepts (classes and properties) of the domain-specific ontology that need to be reflected in the modelled 3D content.

This step of modelling covers the creation of 3D content elements that does not require the use of any additional specific hardware or software for 3D modelling and may be done with a typical semantic editor. However, a visual semantic modelling tool can be developed to further simplify mapping. In comparison to the design of a CrR, mapping is semantically more complex in terms of the structures that need to be created, and it requires more semantic expressiveness.

Mapping is performed with semantic *mapping patterns* (proposed in Section 4.1), which are used according to *mapping guidelines* (proposed in Section 4.2). The restrictive use of the formally specified semantic web standards in the proposed patterns is preferred over the use of other concepts (in particular rules, which have high semantic expressiveness) because of the following two reasons. First, the semantic web standards provide concepts, which are widely accepted and can be processed using well-established tools, such as editors and reasoners. Second, complexity measures have been investigated and specified for the standards including a number of typical reasoning problems (such as ontology consistency, instance checking and query answering)[33], which allows for building applications with more predictable computational time.

### *4.1. Mapping patterns*

*Mapping patterns* (Fig. 1b) specify a means of creating *mapping concepts* (included in an RM) and link concrete components and concrete properties (included in a CrR) to domain-specific concepts (included in an ontology). The particular *mapping patterns* are described in the following sections.

***Classification property***. The *classification property* pattern enables reflection of a property P whose values are labels by a set of classes. In this patterns, for every possible classification value of P, an individual class is created and it is specified as an equivalent to an *OWL hasValue restriction* on P with the required P value. Consequently, every individual that has a particular classification value of P assigned, belongs to one of the created classes. For instance, objects made of metal, wood and plastic may belong to different classes. Every class created may be further described with different properties, using other patterns described in the next sections.

***Multivalued descriptor***. The *multivalued descriptor* pattern enables the specification of desirable DPs for semantic individuals of a common class. To make a class a *multivalued descriptor*, it needs to be specified as a subclass of the intersection of *OWL hasValue restrictions*. Every *restriction* indicates a required value for one of the desirable DPs. For instance, every gold object is yellow and reflects light—one *restriction* specifies colour, the other - shininess.

***Structural descriptor***. The *structural descriptor* pattern enables the creation of a complex structure of classes, which are linked by OPs. To make a class a *structural descriptor*, the class needs to be specified as a subclass of the intersection of *OWL someValuesFrom restrictions*. For instance, every physical object is made of a material, which is reflected by an individual of a linked class. The linked class may also be a *structural descriptor*, thus creating a complex structure of connected classes. In addition, *structural descriptors* can be extended with DPs by applying the *multivalued descriptor* pattern.

***Complex descriptor***. The *complex descriptor* pattern enables the specification of desirable DPs and OPs of individuals based on multiple classes that are assigned to the individuals. In contrast to the previous descriptor patterns, which enable *one class-to-many properties* mapping, complex descriptors allow for *many classes-to-many properties* mapping—determining desirable property values depending on the classes that are assigned to an individual and the classes that are not assigned to the individual. For instance, the colour of a wooden object is brown, while the colour of a waxen object is, e.g., pink, but it also depends on the object temperature. For every distinguishable combination of classes, a separate class (a *complex descriptor*) is created and it is specified as an equivalent to the intersection of the classes that are required and the complements of the classes that are not required. Due to the use of the complements of classes, the close world assumption has to be made to enable the use of this pattern. Every *complex descriptor* can be further extended with DPs and OPs by applying the *multivalued descriptor* and the *structural descriptor* patterns.

***Equivalent and inverse properties***. The *equivalent property* pattern enables the specification of a property as an equivalent to another property by using the `owl:equivalentProperty`. Equivalent properties may be processed in the same manner. A number of properties may be specified as mutually equivalent. For instance, 'includes', 'contains' and 'incorporates' may be counterparts in different ontologies.

The *inverse property* pattern enables the specification of an inverse property using the `owl:inverseOf`, e.g., 'includes' and 'included in' are inverse properties.

***Property chain***. The *property chain* pattern enables connection between individuals of two different classes by linking the classes via mediating classes, which are implemented as *OWL allValuesFrom restrictions*. Every mediator class is specified as a subclass of an *OWL allValuesFrom restriction* that indicates the next class in the chain using an OP. The linked class is also a subclass of an *OWL allValuesFrom restriction*. For instance, in an interior design system, a room includes only objects, which are made of natural materials.

***Semantic rule***. The *semantic rule* pattern is the most general of all of the patterns proposed, and it overtakes the previous patterns in terms of expressiveness. This pattern is used to create logical implications that determine selected properties of individuals (in the *head* of the *rule*) on the basis of other properties of individuals (in the *body* of the *rule*). For instance, every object *standing on* a table has the *y* coordinate calculated on the basis of the *y* coordinate of the table and the *heights* of both objects.

### *4.2. Mapping guidelines*

The use of particular *mapping patterns* for creating particular *mapping concepts* on the basis of concrete components and concrete properties is explained in this section as *mapping guidelines* and depicted in Fig. 1c.

***Mapping presentable objects***. For each domain-specific class C whose individuals need to have independent representations in the created content, create a separate PO-C class and specify it as a super-class of C. Specify DPs and OPs that are inherent to (determine the meaning of) the POs of the PO-C class. Use the *structural descriptor* pattern to link the POs with DIs (concrete components of the CrR) using concrete OPs (Fig. 1cI), e.g., incorporating sub-objects, indicating materials, animations, etc. Use the *multivalued descriptor* pattern to assign the required concrete DPs, e.g., colours, coordinates, dimensions, etc. In such a way, every domain-specific individual of C, which will be created in Step 3, will be described by all concrete properties assigned to the PO-C.

If C occurs in a hierarchy of domain-specific classes, its ascendant domain-specific classes should be described first. Additional presentational effects, which are not inherent to the ascendant classes, should be described directly for C (cf. Sec. 3/2).

***Mapping descriptive classes***. Each domain-specific class C that may be assigned to POs to specify their presentational properties and, in contrast to POs, that does not identify independent entities to be presented, specify as a DC and apply one of the following rules (Fig. 1cII).

1. If C exclusively determines different concrete properties that are not collectively determined by other domain-specific classes (*one class-to-many properties* mapping—cf. Sec. 3/3), describe the required structure of C objects (including all their sub-components) and the required concrete DPs of the C objects using the *structural descriptor* and the *multivalued descriptor* patterns, respectively.
2. If C collectively determines different concrete properties with other domain-specific classes (*many classes-to-many properties* mapping—cf. Sec. 3/4), first, use the *complex descriptor* pattern to create an individual DC for every considered combination of the classes that are assigned and the classes that are not assigned to the object. Second, use the *structural descriptor* and the *multivalued descriptor* patterns for each of these DCs to specify their structures (by concrete OPs) and concrete DPs (as described in the previous sections).

Like in the case of mapping hierarchies of PO-Cs, mapping hierarchies of DCs covers first—the mapping of ascendant DCs and second—the mapping of the descendant DCs (cf. Sec. 3/2).

***Mapping object properties and descriptive individuals***. For each domain-specific OP, which links DIs to other DIs or links DIs to POs, use the *inverse property* pattern to create its inverse OP, if it does not exist (Fig. 1cIII). Maintaining bidirectional links (OPs and their inverse OPs) between semantic individuals (POs and DIs) is recommended to enable flexible application of and reasoning on the *property chain* pattern (which uses OPs to link DIs to DIs and DIs to POs).

***Mapping data properties***. To each domain-specific DP that needs to have presentational effects to the described POs, apply one of the following rules (Fig. 1cIV).

1. If DP exclusively determines particular concrete properties, regardless of other DPs, DCs and DIs assigned to the described object (*one property-to-many properties* mapping, cf. Sec. 3/3), apply one of the following rules.

    (a) If the domain of DP is a PO-C, apply one of the following rules.

        i. If DP is equivalent to a concrete DP, indicate this fact using the *equivalent property* pattern.
        ii. If DP is a classification property (its domain is a finite set of classification data), use the following combination of *mapping patterns*. First, use the *classification property* pattern to create a separate DC for each possible value of DP. Second, extend the DCs to *structural descriptors* and *multivalued descriptors* assigning required concrete OPs and concrete DPs to them.

    (b) If the domain of DP is a DI and its range is a set of classification data, apply the following combination of *mapping patterns*. First, use the *classification property* pattern to create a separate DC for each possible DP value. Second, use the *property chain* pattern to specify the path between the DI and the described PO. Third, extend the DCs using the *structural descriptor* and the *multivalued descriptor* to specify the required structure (by concrete OPs) and concrete DPs of their POs.

2. If the range of DP is a set of numerical data for which a formula can be specified to determine the values of the linked concrete properties on the basis of DP, use the *semantic rule* pattern.
3. If DP collectively determines different concrete properties in combination with other DPs, DCs and DIs assigned to the POs (*many properties-to-many properties* mapping, cf. Sec. 3/4), perform the following steps. First, use the *classification property* pattern to specify a separate DC for every possible value of every considered DP. Second, use the *structural descriptor* and the *multivalued descriptor* patterns to specify structures and concrete DP for the DCs. Third, use the *complex descriptor* pattern to create a new DC that is the intersections of the appropriate DCs, as described in Section 4.2/*Mapping descriptive classes*.

Like in the case of hierarchies of PO-Cs and DCs, mapping domain-specific DPs 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 (cf. Sec. 3/2).

***Mapping relations***. Each domain-specific OP whose domain and range are PO-Cs, specify as an RL, and create a rule for it according to the *semantic rule* pattern (Fig. 1cV) determining the values of desirable properties of the participants of the RL on the basis of properties of other participants (cf. Sec. 3/5).

Each domain-specific class C that has no independent representation in the created content, and for which there are at least two domain-specific OPs that link C with some PO-Cs, specify as an RL and create a rule describing dependencies between particular properties of the PO.

## 5. An example of ontology-based content creation

In this example, a 3D scene, which presents an exhibition in a virtual museum of agriculture, has been modelled with a museum ontology (e.g., by a domain expert), and it is presented in Fig. 2 and in Listing 1. The example stresses the content elements created in Step 3 of the SEMIC approach. The CrR and RM, which provide concrete content elements and link them to domain-specific concepts, are not discussed in this example, as they are created only once for a particular domain-specific ontology and they have been explained in[11,12].

Artefacts in the scene are objects that belong to the six classes (lines 1-4). For every class of artefacts, an object is created and for some objects different materials are assigned (5-6). In addition, six stands that are made of metal are created (7-8) and placed in different locations inside the granary (9-10). Furthermore, the positions of artefacts are specified by a constraint—in a declarative way (which requires knowledge discovery) using rules encoded in the Prova language (11-15). Every artefact that is currently not on any stand, is placed on a stand, on which there is no artefact yet. It is only important to deploy all artefacts on some stands, but it is not important on which stand a particular artefact is placed. In the rules `notStandsOnOthers` and `nothingStandsOnIt`, negation as failure is used. Placing an artefact on a stand is performed by calculating the `x`, `y` and `z` coordinates of the artefact on the basis of the coordinates of the stand (16-20).

Fig. 2. A 3D scene from a virtual museum of agriculture

```
1.  :Artefact rdf:type owl:Class.
2.  (:Smoker, :Stamp, :Ring, :Coin, :Statue, :
     Sower)
3.     rdf:type owl:Class;
4.     rdfs:subClassOf :Artefact.
5.  (:smoker, ..., :sower) rdf:type (:Smoker, ...,
      :Sower);
6.     :madeOf ("clay", ..., "wood").
7.  (:stand1, ..., :stand6) rdf:type :Stand;
8.     :madeOf "metal";
9.     :includedIn :granary;
10.    :position ("...", ...).
11. standsOn(A, B) :- Artefact(A), Stand(B),
     notStandsOnOthers(A),nothingStandsOnIt(B).
12. notStandsOnOthers(A) :- standsOn(A, B), !,
     fail().
13. notStandsOnOthers(A).
14. nothingStandsOnIt(B) :- standsOn(A, B), !,
     fail().
15. nothingStandsOnIt(B).
16. x(A, AX) :- standsOn(A, B), x(B, BX), AX = BX.
17. z(A, AZ) :- standsOn(A, B), z(B, BZ), AZ = BZ.
18. y(A, AY) :- standsOn(A, B), y(B, BY),
    height(A, AHeight), height(B, BHeight),
    AY = BY+(AHeight+BHeight)/2.
```

Listing 1. The conceptual representation of the virtual museum scene

| Modelling paradigm | Declarative content creation | | | | Imperative content creation | Visual content creation | |
|---|---|---|---|---|---|---|---|
| Criterion \ Approach | SEMAC | Latoschik et al. | Troyer et al. | Kalogerakis et al. | ActionScript (Away3D), Java (Java3D) | SketchUp, 3DVIA | Blender, 3ds Max |
| **Conceptual content creation** | | | | | | | |
| using concepts directly related to 3D content | x | x | x | x | x | x | x |
| using abstract concepts (not specific to 3D content) | x | x | x | | x | - | x |
| using hierarchies of classes | x | x | x | x | x | - | x |
| using hierarchies of properties | x | x | | | x | - | x |
| reflection of complex features by classes | x | x | x | | x | - | x |
| reflection of complex features by properties | x | | | | x | - | x |
| reflection of complex features by individuals | x | | | | x | - | x |
| reflection of complex features by combinations of classes | x | | | | - | - | x |
| reflection of complex features by combinations of properties | x | | | | x | - | x |
| reflection of complex features by combinations of individuals | x | | | | x | - | x |
| reflection of relations between objects | x | | | | x | - | x |
| reflection of complex features by rules | x | | | x | - | - | - |
| separation of concerns in content creation | x | | | | | | |
| specifying compatibility between objects | x | | x | x | x | - | x |
| **Knowledge-based content creation** | | | | | | | |
| content modelling based on constraints | x | x | o | o | - | - | o |
| discovery of object properties based on object classes | x | | | | o | - | - |
| discovery of object classes based on object properties | x | | | | - | - | - |
| discovery of dependencies between objects based on object classes and object properties | x | | | | - | - | - |
| management of objects based on properties of other objects | x | | | x | x | - | x |

'x' – meets the criteria 'o' – partially meets the criteria
'-' – does not meet the criteria ' ' – information not available

Fig. 3. Comparison of the selected approaches to modelling 3D content

## 6. Implementation

In the prototype implementation of the SEMIC approach, the domain-specific ontologies, the semantic content models (ML-SCM and SMM) as well as content representations are implemented using the semantic web standards (RDF, RDFS and OWL), which permit the creation of semantic statements (facts) and the Prova declarative language[34], which permits the creation of horn clauses (rules) in the first-order logic.

Transformation of semantic 3D content representations to their final counterparts, which are encoded using selected 3D content representation languages, is implemented as a Java-based application, which uses the Pellet reasoner[35], the Apache Jena SPARQL engine[36] and the Prova rule engine[34]. Currently, final content representations are encoded in VRML, X3D and ActionScript with the Away3D library. However, other languages (imperative and declarative) could also be used.

In the implemented prototype, comprehensive semantic 3D content representations (comprised of both CrRs and CpRs linked by RMs) are expanded using the *restrictions* created according to the patterns proposed in this paper. First, new DIs are created according to *OWL someValuesFrom restrictions* and *OWL allValuesFrom restrictions* and linked to POs by OPs. Second, DPs of POs are set in the reasoning processes according to *OWL hasValue restrictions*. The reasoning process leads to knowledge discovery in the semantic content representation, thus the produced 3D content reflects both the explicit (directly specified) knowledge and the implicit (hidden) knowledge.

## 7. Discussion of selected approaches to modelling 3D content

The SEMIC approach has been compared to selected approaches to modelling 3D content, which are leading in terms of functionality, available documentation and the community of users—approaches to declarative semantic content creation (proposed by Latoschik et al., Troyer et al. and Kalogerakis et al.), imperative programming languages and programming libraries (ActionScript with Away3D and Java with Java3D) as well as environments for visual content creation (advanced environments—Blender and 3ds Max and user-friendly environments—SketchUp and 3DVIA). The comparative analysis performed aims to indicate the major gaps in the available approaches, which are to be covered by the proposed approach.

The analysis covers aspects related to conceptual and knowledge-based 3D content creation (Fig. 3). Conceptual content creation has been considered in terms of representation of 3D content at different levels of abstraction (detail) and the use of the well-established semantic web concepts (classes, individuals, properties and rules) in 3D content creation process. Overall, the available semantic approaches enable the use of basic semantic expressions (combinations of semantic concepts), such as classes and properties, at different levels of abstraction in modelling of content. However, they do not permit a number of more sophisticated combinations of concepts, which are essential to visualization of complex knowledge bases and which are covered by SEMIC. The imperative languages and visual environments permit complex conceptual content representations at different levels of abstraction, however, expressed imperatively, which is not convenient for knowledge extraction, reasoning and content management in web repositories. The available approaches do not support separation of concerns between different users, who have different modelling skills and experience, and are equipped with different modelling tools.

Knowledge-based 3D content creation has been considered in terms of building content representations with regards to discovered properties and dependencies of content objects, which may be hidden (not explicitly specified), but they are the logical implications of facts that have been explicitly specified in the knowledge base. On the one hand, this aspect of content creation is not available in imperative languages, including the languages used in the visual environments. On the other hand, although the available semantic approaches could be extended to enable knowledge-based modelling, currently, they do not address content creation based on extracted data.

## 8. Conclusions and future works

In this paper, a new method of modelling 3D content with domain-specific ontologies has been proposed. The method is an element of the SEMIC approach, which goes beyond the current state of the art by enabling content creation based on complex combinations of semantic concepts.

The proposed method can be used to facilitate creation of different types of content—semantic design of 3D content by domain experts at different levels of abstraction and visual design of ontologies and knowledge bases. Moreover, the approach can improve content management (indexing, searching and analysing of content) in content repositories. Finally, the created content is platform- and standard- independent and it can be presented using diverse 3D content presentation tools, as described in [13].

Possible directions of future research incorporate several facets. First, the proposed method can be evaluated in terms of the effectiveness of modelling 3D content using selected performance indexes, such as the size of CpRs and CrRs, the size of RMs (which are required to enable semantic modelling) and the Halstead metrics. Second, a visual semantic tool supporting the SEMIC approach can be developed. Finally, a specific rule description language can be devised to facilitate description of complex content behaviour.

## Acknowledgements

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

## References

1. *W3C, ISO/IEC 14772-1:1997. The Virtual Reality Modeling Language,* `http://www.web3d.org/x3d/specifications/`; 2014.
2. *W3C, ISO/IEC 19775-1:2008. Extensible 3D (X3D) (2008),* `http://web3d.org/x3d/specifications/`; 2014.
3. *COLLADA Digital Asset Schema Release 1.4.1,* `https://www.khronos.org/files/collada_spec_1_4.pdf`; 2014.
4. *Oracle, Java3D,* `http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html`; 2014.
5. *Away3D,* `http://away3d.com/`; 2014.
6. *Blender,* `http://www.blender.org/`; 2014.
7. *Autodesk, 3ds Max,* `http://www.autodesk.pl/products/autodesk-3ds-max/overview`; 2014.
8. *SketchUp,* `http://www.sketchup.com/`; 2014.
9. *3DVIA,* `http://www.3dvia.com/`; 2014.
10. 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,* 2013;**160**:244–257.
11. Flotyński, J., Walczak, K.. Semantic modelling of interactive 3d content. In: *Proceedings of the 5th Joint Virtual Reality Conference*. Paris, France; 2013, p. 41–48.
12. 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: IEEE; 2013, p. 541–548.
13. Flotyński, J., Walczak, K.. Multi-platform semantic representation of 3d content. In: *Proceedings of the 5th Doctoral Conference on Computing, Electrical and Industrial Systems*. Lisbon, Portugal; 2014, p. 63–72.
14. 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; 2006, p. 43–50.
15. Walczak, K., Cellary, W., White, M.. Virtual museum exhibitions. *IEEE Computer, Volume 39, Issue 3; March 2006;* 2006;:93–95.
16. Walczak, K.. Flex-vr: Configurable 3d web applications. In: *Proc. of the Int. Conf. on Human System Interaction HSI 2008*. Kraków, Poland; 2008, p. 135–140.
17. Walczak, K., Cellary, W.. X-vrml for advanced virtual reality applications. *IEEE Computer, Volume 36, Nr 3* 2003;:89–92URL: `http://www.computer.org/computer/co2003/r3toc.htm`.
18. Walczak, K.. Dynamic database modeling of 3d multimedia content. In: Cellary, W., Walczak, K., editors. *Interactive 3D Multimedia Content - Models for Creation, Management, Search and Presentation*. Springer. ISBN 978-1-4471-2496-2; 2012, p. 55–101.
19. 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, p. 272–279.
20. 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, p. 3–18.
21. Gutierrez, M., Vexo, F., Thalmann, D.. Semantics-based representation of virtual environments. *International Journal of Computer Applications in Technology* 2005;**23**:229–238.
22. 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: ACM; 2009, .
23. Flotyński, J.. Harvesting of semantic metadata from distributed 3d web content. In: *Proceedings of the 6th International Conference on Human System Interaction (HSI), June 06-08, 2013, Sopot (Poland)*. IEEE. ISBN 978-1-4673-5636-7; 2013, .
24. Flotyński, J., Walczak, K.. Describing semantics of 3d web content with rdfa. In: *The First International Conference on Building and Exploring Web Based Environments, Sevilla (Spain), January 27 - February 1, 2013*. ThinkMind. ISBN 978-1-61208-248-6; 2013, p. 63–68.
25. Flotyński, J., Walczak, K.. Attribute-based semantic descriptions of interactive 3d web content. In: Kiełtyka, L., editor. *Information Technologies in Organizations - Management and Applications of Multimedia*. Wydawnictwa Towarzystwa Naukowego Organizacji i Kierownictwa - Dom Organizatora. ISBN 978-83-7285-691-3; 2013, p. 111–138.
26. Flotyński, J., Walczak, K.. Microformat and microdata schemas for interactive 3d web content. In: Ganzha, M., Maciaszek, L., Paprzycki, M., editors. *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems Kraków, Poland, September 8-11, 2013*; vol. 1. Polskie Towarzystwo Informatyczne. ISBN 978-1-4673-4471-5; 2013, p. 549–556.
27. 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; 2008, p. 17–20.
28. Spagnuolo, M., Falcidieno, B.. 3d media and the semantic web. *IEEE Intelligent Systems* 2009;**24**:90–96.
29. Aylett, R., Luck, M.. Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence* 2000;**14**:3–32.
30. 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; 2007, p. 225–234.
31. 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, p. 25–39.
32. Latoschik, M.E., Frohlich, C.. Semantic reflection for intelligent virtual environments. In: *IEEE Virtual Reality Conference 2007*. Charlotte, USA; 2007, p. 305–306.
33. *W3C, OWL,* `http://www.w3.org/2001/sw/wiki/OWL`; 2014.
34. Kozlenkov, A., Paschke, A.. *Prova Rule Language,* `https://prova.ws/`; 2014.
35. *Clark&Parsia, Pellet: OWL 2 Reasoner for Java,* `http://clarkparsia.com/pellet/`; 2014.
36. *Apache Jena,* `http://jena.apache.org/`; 2014.