Multi-Platform Semantic Representation
of Interactive 3D Content

Jakub Flotyński, Krzysztof Walczak

Poznań University of Economics,
Niepodległości 10, 61-875 Poznań, Poland
{flotynski, walczak}@kti.ue.poznan.pl
http://www.kti.ue.poznan.pl

Abstract. In this paper, a semantic approach to building multi-platform 3D content is proposed. The presented solution is intended to enable flexible and efficient creation of 3D presentations covering a wide range of target platforms – visualisation tools, content representation languages and programming libraries. Referring to the semantics of particular content elements can facilitate conceptual knowledge-based content creation at arbitrarily high levels of abstraction, and it can improve indexing, searching and analysis of 3D content in a variety of application domains on the web.

Keywords: 3D web, Semantic Web, 3D content, multi-platform, ontology.

1   Introduction

Widespread use of interactive 3D technologies and multimedia platforms, including powerful and omnipresent mobile devices, 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. 3D technologies become increasingly used in various application domains, such as education, training, entertainment and social media, significantly enhancing possibilities of presentation and interaction with multimedia information sources, thus increasing collective awareness of their users.

However, to reach a high number of recipients on the web, support for a diversity of hardware and software systems must be provided. Currently, wide coverage of different hardware and software systems by 3D presentations is typically achieved by providing separate implementations of various 3D content browsers and presentation tools. However, in contrast to the development of individual 3D content browsers and presentation tools for different target systems, compatibility of 3D content representation with diverse popular presentation platforms could improve the reuse of 3D content components and the overall use of 3D content on the web. In such an approach, once 3D content is created, it can be presented using different platforms. Moreover, such approach does not require users to install additional software, but it can leverage well-established 3D content browsers and presentation tools that may already be installed on the users' systems (e.g., Adobe Flash Player or WebGL/X3DOM-compliant web browsers). However, currently, the development of 3D platforms is driven by large industry players in a competitive environment and the issue of multi-platform 3D content presentation is still neglected, resulting in

fragmentation of content and technologies. This is an important obstacle preventing the mass use of 3D application interfaces.

The main contribution of this paper is a multi-platform semantic representation of interactive 3D content. The approach permits flexible and efficient creation of 3D content for a variety of target presentation platforms. In the proposed solution, once the structure of 3D content is designed, it can be automatically transformed into different final presentation forms, which are suited to different 3D content presentation platforms. The selection of the target platforms to be used is an arbitrary decision of a system designer. Referring to the semantics of particular 3D content components and the conformance to the well-established Semantic Web standards enables 3D content representation that is independent of particular browsers and presentation tools, permits reflection of complex dependencies and relations between content components, and can facilitate indexing, searching and analysis of 3D content in a variety of web applications.

2   Interactive 3D Content in Collective Awareness Systems

Support for content presentation across different platforms is essential for building collective awareness systems. Several works have been devoted to multi-platform 3D content presentation. In [1], a specific 3D browser plug-in for different web browsers has been described. In [2], an approach to hardware multi-platform 3D content presentation based on MPEG-4 has been proposed. In [3], an approach to multi-platform visualisation of 2D and 3D tourism information has been presented. In [4], an approach to adaptation of 3D content complexity with respect to the available resources has been proposed. In [5], a multi-platform on-line game has been presented. In [6], integrated information spaces combining hypertext and 3D content have been proposed to enable dual-mode user interfaces – embedding 3D scenes in hypertext and immersing hypertextual annotations into 3D scenes – that can be presented on multiple platforms on the web. The aforementioned works cover the development of 3D content presentation tools and environments as well as contextual platform-dependent content adaptation. However, they do not address comprehensive and generic methods of content transformation to improve building of multi-platform 3D content presentations.

Collective awareness systems require sharing the meaning of data and content between users and systems. Numerous works have been devoted to semantic 3D representation of information. In [7], integration of X3D and OWL using scene-independent ontologies and the concept of semantic zones have been proposed. In [8], an ontology for X3D as well as semantic properties for coupling VR scenes with domain knowledge have been described. In [9], 3D content representation based on reusable elements with specific roles has been introduced. In [10], an approach to generating virtual words upon domain ontologies has been considered. In [11], semantic entities in VR applications have been discussed. The aforementioned approaches address different aspects of semantic modelling of 3D content, but they lack general solutions for comprehensive conceptual creation of 3D content with

respect to its components, properties and relations, at an arbitrarily high level of semantic abstraction.

3   The Method of Semantic 3D Content Creation

Although several approaches have been proposed for semantic modelling of 3D content, they lack general and comprehensive solutions for flexible creation of 3D representation of information. The following requirements, which have been specified for an approach to 3D content creation, go beyond the current state of the art in semantic modelling of 3D content. First, the approach should enable conceptual declarative modelling of content with discovery of hidden knowledge, which is not specified explicitly, but has impact on the modelled content. Second, it should reduce the effort in content design by enabling modelling of complex content components and properties at arbitrarily chosen levels of abstraction, including both the aspects that are directly related to 3D content and the aspects that are specific for a particular domain (e.g., to facilitate content creation by domain experts who are not IT-specialists). Third, it should be independent of particular hardware and software platforms to enable multi-platform 3D content presentation.

This section provides an overview of a method of semantic creation of interactive 3D content (proposed in [12]), which leverages the model of multi-platform 3D content proposed in the next section. In the presented method, the creation of 3D content is a sequence of partly dependent activities (Fig. 1). Modelling of 3D content in the first three activities – design of a concrete representation of 3D content, mapping the concrete representation to domain-specific concepts, and design of a conceptual representation of 3D content – is performed by a developer and a domain expert, depending on the expertise required in the particular modelling activity. The two following activities of the method – expanding the content representation and building the final content representation – are performed automatically by specific software. The activities precede 3D content presentation, which may be done using various content presentation platforms.
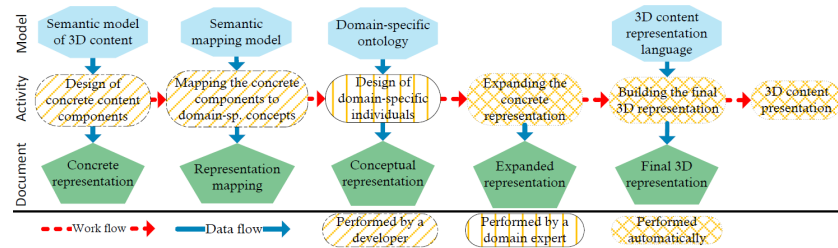
Fig. 1. Semantic creation of 3D content

The design of a concrete semantic representation (activity 1) – provides particular elements of 3D content to enable representation of domain-specific concepts that are further used in activity 3. The result of activity 1 is a concrete semantic representation of 3D content, which is a knowledge base compliant with the platform-independent
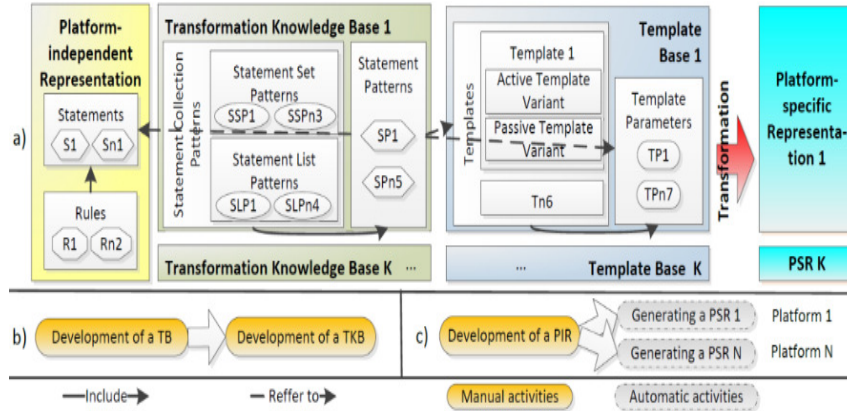
semantic model of 3D content (PIM – proposed in [13]). The knowledge base incorporates semantic components and properties that are directly related to 3D content, e.g., meshes, groups of objects, materials, viewpoints, events, etc.

The mapping of a concrete 3D content representation (created in activity 1) to domain-specific semantic concepts enables 3D presentation of domain-specific knowledge bases (created in activity 3) [14]. Mapping is performed once for a particular domain-specific ontology and a concrete representation, and it permits the reuse of concrete components and properties for forming 3D representations of various domain-specific individuals (which conform to the domain-specific ontology).

The design of a conceptual semantic representation (activity 3) enables creation of 3D content at a high level of abstraction with a domain-specific ontology. This activity can be performed many times for a particular domain-specific ontology, a concrete representation and a mapping. The following activities of the content creation process can be performed automatically. Expanding the content representation multiplies the concrete components (created in activity 1), which are associated with domain-specific concepts, and assigns them directly to domain-specific individuals (created in activity 3). Building a final 3D representation is a transformation of the expanded representation to its final 3D counterpart, which is encoded using a particular 3D content representation language. This stage of the method is an extension of the approach proposed in [15], and it uses the multi-platform model of 3D content, which is the main contribution of this paper.

4   Multi-Platform Model of 3D Content

Although numerous solutions have been proposed for creating semantic representations of 3D content, they do not enable flexible creation of multi-platform 3D content, which can be used for visualisation of various types of information using a multitude of available content presentation tools.

Fig. 2. Multi-platform model of 3D content (a), introducing a new platform to the system (b), and modelling multi-platform 3D content (c)

In this section, a new multi-platform model of 3D content is proposed (Fig. 2a). The model extends the semantic content model (cf. [13,14]) and enables the last activity of the content creation method (cf. [12] and Section 3) – building the final 3D content representation. The model consists of four parts: *platform-independent content representations* (PIRs), *platform-specific content representations* (PSRs), *template bases* (TBs) and *transformation knowledge bases* (TKBs). The model enables transformation of PIRs to PSRs, which may be visualised with diverse 3D content representation languages (e.g., X3D, Java), programming libraries (e.g., Java3D, Away3D) and game engines (e.g., Unity, Unreal), which are determined by the content presentation platforms to be used. Transformation is performed using TKBs and TBs, which are specific to particular presentation platforms. An individual TB and its corresponding TKB are created once every time a new presentation platform is introduced to the system (Fig. 2b). Since a TB and a TKB are added, they may be used for the development of various PSRs that are permitted by the new presentation platform (Fig. 2c). The particular parts of the proposed content model are described in the following subsections.

4.1   Platform-independent Representations

A *platform-independent representation* (PIR) of 3D content is a knowledge base comprised of platform agnostic semantic elements: individuals – reflecting 3D content components, and properties – reflecting dependencies and relations between the components. Every PIR is compliant with the PIM (cf. [13] and Section 3). A PIR declaratively describes 3D content, as it is a set of semantic *statements* (Ss) and *rules* (Rs), which may be described using, e.g., RDF and SWRL, respectively. Every S is a triple consisting of a *subject*, a *property* and an *object*. Ss are used to describe aspects of 3D content that are related to geometry (e.g., shape), structure (e.g., sub-components), space (e.g., position), appearance (e.g., texture) and simple behaviour (e.g., animations). Every R is an *implication* including a *body* and a *head*, which are conjunctions of Ss. An R is interpreted in the following way: if the *body* is satisfied, then the *head* is also satisfied. Rules are used to describe complex behaviour of 3D content, in particular complex animations and interactions, which may depend on multiple factors, such as time, state of objects, user actions, etc. PIRs can be presented on different platforms when transformed to PSRs with appropriate TBs and TKBs.

4.2   Platform-specific Representations and Template Bases

A *platform-specific representation* (PSR) of 3D content is a counterpart to a PIR, and it includes 3D objects and 3D scenes, which are encoded using a 3D content representation language. Every PSR is built upon a TB, as it is a combination of *templates* (Ts), which are parameterised fragments of code (e.g., sequences of instructions) of a particular 3D content representation language. Ts may be linked to individual Ss as well as to Ss that are included in Rs of PIRs. For some of the target platforms used, this context of linking may determine the form of the T used. Every T may be given in two *template variants* that differ depending on the context of its use.

Active template variants (ATVs) are used for the independent Ss (to express logical facts) and the Ss included in the *heads* of Rs (to express logical results), while *passive template variants* (PTVs) are used for the Ss included in the *bodies* of Rs (to express logical conditions). For instance, the S `[object pim:shape "cone"]` may be transformed to an ATV `[object.shape="cone"]` or to a PTV `[object.shape.isEqual("cone")]`, which are encoded in an object-oriented language.

To enable flexible transformation of PIRs to PSRs, the granularity of Ts should reflect the semantic granularity of their corresponding Ss – Ts should neither extend nor narrow the meaning of the linked Ss. Composing basic Ss into semantically more expressive Ss is performed in the previous activities of the semantic modelling method presented. *Template parameters* (TPs) enable the reuse of individual Ts in different contexts in combination with other Ts, e.g., by joining or nesting Ts. All PSRs that are based on a common TB and which are presented on a common target platform are generated from PIRs using a common TKB.

4.3 Transformation Knowledge Bases

A *transformation knowledge base* (TKB) links an individual TB to the PIM, thus enabling reflection of semantic Ss by parameterised Ts. A TKB specifies a transformation of PIRs to the corresponding PSRs that are to be presented on a common target platform. Semantic elements contained in a TKB do not influence the modelled 3D content. A TKB consists of *statement patterns* (SPs), each of which matches a group of possible Ss that should be processed collectively. Linking a group of Ss to an SP is performed by semantic generalisation. A generalisation may pertain to the *subject*, the *property* or the *object* of Ss. For instance, a possible generalisation (an SP) of the S `[object pim:color "red"]` in terms of property, is the SP `[object pim:appearanceProperty "red"]`, while a possible generalisation of the S `[object rdf:type pim:Mesh3D]` in terms of object, is the SP `[object rdf:type pim:GeometricalComponent]`.

A target PSR, which is generated based on Ss of a PIR, may require an exchange of TPs in a group of different Ts, which are associated with these Ss, and it may require a specific order of Ts. For this purpose, SPs may be gathered into *statement collection patterns – statement set patterns* (SSPs), which do not respect the order of the associated Ts, and *statement list patterns* (SLPs), which do respect the order of the associated Ts in the resulting PSR. For instance, while a declarative PIR including the pair of Ss `[light pim:intensity "10". light rdf:type pim:DirectionalLight.]` does not depend on the order of the Ss, a corresponding imperative PSR may require the following order of the associated Ts (imperative code instructions) `[DirectionalLight light = new DirectionalLight(); light.intensity = 10;]`.

5   Example of a Multi-Platform 3D Content Representation

In this section, an example of a multi-platform 3D content representation is discussed. The example 3D scene (Fig. 3) could be used, e.g., in a collective educational system or a virtual museum system. The scene includes a light source and a 3D model of a plough, which has been retrieved from a repository of a virtual museum of agriculture. In the example, the Adobe Flash Player is selected as the target presentation platform. However, in general, arbitrary selected multiple platforms could be used. A PIR of the scene is presented in Listing 1. It uses the PIM (the `pim` prefix) to describe the 3D content components regardless of any particular target platform. A TB and a TKB, which enable encoding of PIRs with the ActionScript imperative programming language and the Away3D library, are presented in Listings 2 and 3. The listings include only elements that are crucial for the discussion.

The S1 and S2 statements (Listing 1) create the scene and set its background with the appropriate colour. The Ss are processed according to the SLP1 (Listing 3), which includes two SPs. The S1 is dynamically linked to T1 (by its *property* and *object*) and the S2 is dynamically linked to T2. The T1 is a template of an ActionScript document and it initializes a 3D view, while the T2 sets the background of a scene. The order of the SPs in the SLP1 determines the order of the T1 and T2 in the resulting document (a PSR). The SLP1 requires a new object to be created before any property of this object is set (according to the imperative programming paradigm). If the T linked to the SP1 includes the `$actions` TP, the T linked to the SP2 is injected into this TP, else the T of the SP2 follows the T of the SP1 in the final PSR. The S3 and S4, which create a light source, are processed in the same order (determined by the SLP1) as the previous Ss, independently of their initial order in the PIR.

The plough, which is a virtual museum artefact, is a structural component that consists of the three geometrical components—a box, a wheel and a frame (S5-S7). The SLP2 is applied to the S5-S7—in the resulting PSR, instructions that create new objects in the scene, precede instructions that link these objects by properties (T3). In the example, the resulting PSR (an ActionScript document) includes the Ts in the required order, with the proper TPs specified. The PIR has been created with the Protégé editor. However, a tool for visual semantic modelling can be developed.

Fig. 3. A final 3D content representation

```
S1: scene rdf:type pim:Scene.
S2: scene pim:backgroung "beige".
S3: light pim:intensity "10.0".
S4: light rdf:type pim:DirLight.
S5: plough rdf:type
       pim:StructuralComponent.
S6: plough pim:includes box ,
       wheel , frame.
S7: box, wheel, frame rdf:type

       pim:GeometricalComponent.
```

Listing. 1. A platform-independent representation (PIR) of 3D content

```
01: T1: {
02:    $declarations
03:    public function Main():void {
04:      var $object:View3D = new
         View3D();
05:      $actions } }
06: T2: { $object.backgroundColor =
       $data; }
07: T3: {
       $object1.addChild($object2); }
```

Listing. 2. A platform-specific template base (TB) for an object-oriented language

```
SLP1: { SP1: ?object rdf:type
       pim:ContentComponent.
  SP2: ?object pim:dataProperty
       ?data.
  SP1.$actions == SP2 }
SLP2: { SSP: { SP1: ?object1
       rdf:type pim:ContentComponent.
  SP2: ?object2 rdf:type
       pim:ContentComponent. }
  SP3: ?object1 pim:objectProperty
       ?object2. }
```

Listing. 3. A transformation knowledge base (TKB)

6   Conclusions and Future Works

In this paper, a new approach to building multi-platform representations of 3D content has been proposed. The presented solution has several important advantages in comparison to the available approaches to 3D content presentation. First, it is more convenient for environments, which cover various hardware and software systems, as the development of TBs and TKBs requires less effort than the development of individual content models or content browsers. Second, the possible use of well-established 3D content presentation tools, programming languages and libraries liberates users from the installation of additional software, which can improve the dissemination of 3D content. Third, the use of Semantic Web standards permits conceptual knowledge-based content modelling that refers to hidden information inferred using, e.g., RDF, OWL or SWRL reasoners. Moreover, semantic

representation enables more efficient and flexible methods of indexing, searching and analysis of the content regarding complex dependencies and relations between its components, provides methods of describing rules of combining different components and permits description of complex content behaviour.

The following directions of future research are possible. First, the proposed approach needs to be implemented for selected content representation languages (e.g., ActionScript and X3D), and combined with the approach to transformation of 3D content description formats proposed in [15]. Second, the approach should be evaluated and compared to other solutions in terms of both: the possibilities of high-level conceptual content creation and the effort in the implementation of multi-platform 3D content presentations. Third, the complexity of TBs and TKBs for different target languages – imperative and declarative – could be compared. Furthermore, a visual modelling tool supporting the semantic content creation can be developed. Finally, a persistent link between semantic and final content representations can be proposed to provide real-time synchronisation of the content state.

References

1. Mendes, C. M., Drees, D. R., Silva, L., Bellon, O. R.: Interactive 3D visualization of natural and cultural assets. In: Proc. of the 2nd Workshop on eHeritage and Digital Art Preservation. pp. 49--54. Firenze, Italy (2010)
2. Celakovski, S., Davcev, D.: Multiplatform real-time rendering of MPEG-4 3D scenes with Microsoft XNA. In: ICT Innovations 2009, pp. 337--344. Springer-Verlag (2010)
3. Almer, A., Schnabel, T., Stelzl, H., Stieg, J., Luley, P.: A tourism information system for rural areas based on a multi-platform concept. In: Proc. of the 6th Int. Conf. on Web and Wireless Geographical Inf. Systems, pp. 31--41. Hong Kong, China (2006)
4. Tack, K., Lafruit, G., Catthoor, F., Lauwereins, R.: Platform independent optimisation of multi-resolution 3D content to enable universal media access. In: The Visual Computer, Volume 22, Issue 8, pp. 577--590. Springer-Verlag (2006)
5. Han, J., Kang, I., Hyun, C., Woo, J., Eom, Y.: Multi-platform online game design and architecture. In: Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction, pp. 1116--1119. Springer-Verlag Berlin, Heidelberg (2005)
6. Jankowski, J., Decker, S.: A dual-mode user interface for accessing 3d content on the world wide web. In: Proc. of the 21st Int. WWW Conf. pp. 1047--1056. Lyon, France (2012)
7. Pittarello, F., Faveri, A.: Semantic description of 3D environments: a proposal based on web standards. In: Proceedings of the 11th International Conference on 3D Web Technology. pp. 85--95. Columbia, MD, USA (2006)
8. 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, pp. 43--50. Alexandria, VA, USA (2006)
9. Walczak, K.: Flex-VR: Configurable 3D Web Applications. In: Proc. of the Int. Conf. on Human System Interaction, pp. 135--140. Kraków (2008), ISBN: 1-4244-1543-8
10. 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 – Volume 83, Australian Computer Society, Inc. Darlinghurst, pp. 3--18. Australia (2007), ISBN: 978-1-920682-64-4

11. Latoschik, M.E., Frohlich, C.: Semantic Reflection for Intelligent Virtual Environments. In: IEEE Virtual Reality Conference, pp. 305--306. Charlotte, NC, USA (2007)
12. Flotyński, J., Walczak, K.: Semantic Modelling of Interactive 3D Content. In: Proc. of the 5th Joint Virtual Reality Conference, pp. 41--48. Paris, France (2013)
13. Flotyński, J., Walczak, K.: Semantic Multi-layered Design of Interactive 3D Presentations. In: Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 541--548. Kraków, Poland (2013)
14. Flotyński, J., Walczak, K.: Conceptual Semantic Representation of 3D Content. In: LNBIP, Volume 160, pp. 244--257. Springer-Verlag, Berlin Heidelberg (2013)
15. Flotyński, J., Dalkowski, J., Walczak, K.: Building multi-platform 3D virtual museum exhibitions with Flex-VR. In: Proceedings of the 18th International Conference on Virtual Systems and Multimedia, pp. 391--398. Milan, Italy (2012)