

A Generative Approach to Virtual Museums

Daniel Sacher¹, Daniel Biella² and Wolfram Luther¹

¹*INKO, University of Duisburg-Essen, Lotharstr. 63, 47057 Duisburg, NRW, Germany*

²*ZIM, University of Duisburg-Essen, Forsthausweg 2, 47048 Duisburg, NRW, Germany*

Keywords: ViMEDEAS, ViMCOX, Replicave, Metadata, Virtual Museum, 3D Framework, X3D, X3Dom, WebGL, LIDO.

Abstract: In this paper we introduce an automated approach for generating virtual museum exhibitions based on metadata. We will give an insight into the incorporation progress of X3Dom, as an additional export format supported by the 3D content generation framework, Replicave. The feasibility of these concepts will be tested by providing new application examples within the scope of our ongoing cooperation with the Salomon Ludwig Steinheim Institute.

1 INTRODUCTION

In three recent papers (Biella et al., 2012), (Biella et al., 2010b) and (Biella et al., 2010a), we described the Virtual Museum Exhibition Designer using an Enhanced ARCO Standard (ViMEDEAS). ViMEDEAS is a set of authoring tools and frameworks to cover the entire design process of virtual museum planning, creation, archiving, dissemination and presentation.

The metadata set ViMCOX (Virtual Museum and Cultural Object Exchange Format) version 1.1 (Wolf et al., 2012), included in ViMEDEAS, describes virtual museums containing both classical and contemporary art, interactive exhibition content, assets, spatial exhibition design and outdoor areas. ViMCOX is based on international metadata standards and uses LIDO version 1.0 as interchange and harvesting format for cultural objects (Coburn et al., 2010).

The Replicave framework is used to present exhibitions locally or online (Biella et al., 2012). Replicave is a Java based X3D toolkit for programmatic modeling of 3D scenes. It generates 3D content at runtime, based on exhibition templates or ViMCOX metadata instances. Recent Web3D developments, such as HTML5, WebGL and X3Dom, overcome the disadvantages of plugin-based rendering and facilitate the use of widely adopted web techniques like CSS, JQuery and Ajax as well as providing content interactivity with existing web APIs and services. To this end, we propose a new development track using X3Dom as an additional visualization framework. It is our goal to ease development of user interfaces using HTML5 and JavaScript to enable more suitable illustrations of metadata or larger text-based content

embedded into HTML markup instead of displaying text-based metadata within the 3D scene.

In addition we will introduce our generative approach for generating and publishing virtual museums. This addresses metadata-based modeling, procedural generation of content and 3D visualization using the rendering platforms X3D and X3Dom.

2 PORTING REPLICAVE (X3Dom)

The framework Replicave was developed to provide a cost-efficient way to create virtual museum exhibitions by reusing 3D models and dynamically generated content based on room templates or ViMCOX metadata. The framework uses Java and X3D to generate 3D scenes. Replicave is roughly divided into three work packages: X3D markup generation, a template engine and factory classes for room creation as well as ViMCOX → X3D metadata mapping. X3D is an ISO standard 3D format, which is not natively supported by web browsers and requires plugin installation. X3Dom is an open source framework for rendering X3D content in web browsers. X3Dom uses WebGL and OpenGL ES as a rendering platform, which requires a WebGL capable browser such as Google Chrome, Safari or Firefox. The Adobe Flash Player version 11 plugin is used as a fallback renderer for non-WebGL browsers or if WebGL support is not available. X3Dom is developed and maintained by Fraunhofer IGD. X3Dom proposes a new HTML profile, which is an extension of X3D's interchange profile. The HTML profile does not include internal script nodes, pointing device sensor nodes or

support for X3D ProtoTypes (Jung et al., 2012). The embedded X3D scene graph can be manipulated using the X3Dom integration model (Behr et al., 2009) and JavaScript to access the HTML5 DOM tree, rather than using the Scene Access Interface (SAI) and X3D plugin interfaces (Web3D-Consortium, 2004).

To accomplish the adaption of X3Dom we considered the use of an additional interposed XSLT layer to convert the generated X3D markup on the fly. The X3D-Edit authoring tool already provides XSLT stylesheets to transform X3D markup into X3Dom web pages with xhtml markup (Savage-NPS, 2012). Partially or unsupported node types are detected during transformation, but are not omitted (cf. (X3D-Graphics, 2012)). Furthermore, this approach is suitable for converting static scenes but not in terms of dynamic scenes or interactive content generation and to utilize the advantages and new features of X3Dom as part of a SDK. Therefore, we decided to reuse and extend the existing content generation classes as well as the X3D writer package of the Replicave framework rather than just converting X3D content after export. Replicave's X3Dom extension is an independent content generator and SDK. Rapid porting and extension of source code was possible due to the markup similarity and almost identical node specification of X3D and X3Dom. Extensions to load *Shader*, create *CubeMaps* and to add JavaScript-EventHandler were implemented and are accessible through the Replicave API. X3Dom allows adding of JavaScript event handlers to the following X3D nodes: *Group*, *Transform* and *Shape*. It also supports these mouse events: *click*, *mousedown*, *mouseup*, *mousedown*, *mouseover*, *mouseout* and *mousemove* (Behr et al., 2011). Attaching of inline events is supported via the Replicave API by utilizing corresponding X3D node attributes (DOM Level 0). In addition, events can be attached or removed through the HTML markup (DOM Level 2).

Additional ports have been made to be conform with X3Dom. This includes the conversion of 23 pre-assembled assets from X3D ProtoTypes to native X3D Documents, because X3Dom currently doesn't provide support to load external or internal X3D ProtoTypes. Reordering of vertices for generating geometric window recesses of exhibition rooms, was also required to be conform with the ear clipping algorithm used to triangulate concave polygons in X3Dom.

We dispensed with the port of the existing exhibition-templates, room-templates and corresponding room connectors because a similar feature set and automatized generation of exhibition space can be adapted more flexibly using the proposed

metadata-based modeling approach described in section 3. The reuse of the software package to process ViMCOX XML instances, required modification of the 3D metadata mapping procedure described in (Biella et al., 2012). There are currently two different approaches for specifying rooms using ViMCOX metadata (Wolf et al., 2012):

1. To create rectangular room shapes, it is sufficient to specify two side lengths and the height of the room. Each wall is aligned to a compass direction. Referenced objects can be assigned to walls by specifying compass directions.
2. To design custom room shapes, it requires at least three wall definitions. Each wall needs to be specified separately by providing absolute position data of the bottom two corner points as well as wall heights for construction. Objects are assigned to walls by specifying the corresponding wall number. This allows arbitrary wall positioning to form the desired room shape.

The first method described, maps the ViMCOX metadata to a rectangular room template and uses the parameters for length and height to construct the room, while the second approach uses factory classes to generate walls and rooms based on the mapped metadata input (Sacher, 2011). It is sufficient to focus only on the second approach because of equal modeling capabilities and less effort is required to maintain two implementations. Therefore, we implemented an interface which calculates absolute wall positions and transforms the compass directions of referenced exhibits to wall numbers. This ensures that older museum versions remain executable without post-processing.

3 GENERATIVE EXHIBITIONS

On-the-fly content creation based on exhibition room templates (mediaroom, gallery, entrancehall, cloakroom) was presented in (Biella et al., 2012) and (Biella et al., 2010b). These templates can be configured using XML configuration files or programmatic through the Replicave API. The previously proposed template-based approach is rather static and limited due to the fact that only three templates are customizable in terms of individual positioning of exhibition content. Moreover, room appearances cannot be customized without manual programming, and the ability to create individual sophisticated room shapes is limited. Our metadata-based generative approach facilitates more flexible generation of exhibition space and enables automatic distribution of con-

tents, metaphoric room connectors and camera viewpoints marking points of interest.

The generative modeling paradigm, or procedural modeling, refers to parameters or lists of operations to generate 3D meshes and content using algorithms rather than modeling interactively with CAD software and exporting 3D models. Unlike the approach proposed in GML (Generative Modeling Language) (TU Graz, 2010), our parameter design refers not to low-level abstractions of components to describe 3D shapes such as points, lines or polygons, but to an abstract representation of stylistic devices such as (partition-)walls, windows, doorways or the complete room shape. Parameters are specified in the descriptive metadata sets of ViMCOX XML instances, which are used by Replicave as configuration for 3D content generation. Architectural design can be specified as 2D floorplan in the ViMCOX metadata set. The 2D data will be transformed to 3D when the virtual museum is generated. Floorplan data can be acquired using different techniques. Modeling approaches based on 2D floorplans—using shape grammar extensions and ontologies with export for different renderer (Tomas Trescak, 2012), transformation of Industry Foundation Classes (IFC) of buildings into CityGML XML instances as explicit geometry (J. Benner, A. Geiger, K. Leinemann, 2005) or 2D floorplanning using graph theory (Ancona et al., 2007)—can be found in the literature. Our concept involves two means of generating content—metadata and 3D content generation. This facilitates the exchange of authoring tools, the use of different 3D visualisation techniques based on client capabilities and dynamic generation of exhibitions based on larger collections. Our metadata processor, the MetaToolkit (Biella et al., 2012), provides interfaces and model classes for automatic distribution of exhibits, addition of predefined viewpoints for each exhibit and spatial modeling using algorithms. The MetaToolkit uses JAXB (Java Architecture for XML Binding) to accomplish a three tier schema mapping XML \leftrightarrow POJO \rightarrow X3D. Schema-derived data classes generated by JAXB are transformed into corresponding model classes to achieve convenient use in partial applications of ViMEDEAS and also to serve as input for eased mapping of metadata to actual 3D content, generated by Replicave (Biella et al., 2012).

An abstract architectural overview is presented in Figure 1. Multimedia content; exported cultural object metadata (LIDO); 3D models like (interactive) exhibits, 3D buildings or 3D interior rooms; 2D pictures; or floorplans as 2D point set can be used in ViMCOX. Algorithms can facilitate 2D floorplanning and automatic determination of exhibit distribution or

room layout. Authoring tools can modify the content base as well as ViMCOX metadata instances. The dissemination and visualization layers are middleware for assembling and publishing virtual museum instances, locally or on the web, as well as interpreting and rendering ViMCOX metadata instances on different visualization platforms. ViMEDEAS currently only support X3D and X3Dom. Our generative modeling approach becomes more efficient when larger content bases need to be processed, which will be described in the following application example as part of our ongoing research (cf. section 3.1). The generated show rooms will serve as an initial starting point for further manual processing and visual design. Current research addresses the development of API interfaces to provide support for automatized generation of virtual exhibition rooms based on simplified XML configuration files and virtual objects stored as ViMCOX instances. The exploratory stage focuses on paintings and basic room shapes—rectangular shapes (aligned to compass-directions) or room shapes in form of regular polygons. Our algorithm requires a list of walls for each room and an ordered list of exhibits and doors to be placed on the corresponding walls. Parameters for defining left and right spacing between objects are optional and were initially set to 1.0m. The same applies for heights of walls and vertical positioning of the individual virtual objects. Initial textures for walls, ceiling and floor are set to the Replicave preset. The algorithm then calculates the individual wall lengths and determines two side lengths (North-South and West-East) for the construction of rectangular rooms. For rooms in shape of regular polygons, the largest wall will be used as the construction parameter for calculating all wall coordinates. These calculations are done to preserve symmetric room shapes and to determine additional spacing parameters to center the objects as groups on walls, if desired. To calculate wall lengths, the algorithm refers to the virtual object dimension stored in the linked ViMCOX metadata instances. In the case of the ViMCOX object type painting, the corresponding dimensions are the image area, frame size and passe-partout size. Afterwards, the algorithm determines the positions of each virtual painting or door and calculates a default viewpoint for each object. The result is a single assembled ViMCOX metadata instance, containing the generated room instances with viewpoints, thresholds (doors) and referenced objects as well as copies of the initial virtual object instances. The generated virtual rooms can then be visualized, locally or online, using the Replicave framework (cf. section 3.2).

3.1 Application Example

The virtual Fleischhacker museum hosts 196 pictorial exhibits; 3D assets like plants, pillars, glass vitrines, benches and information tableaux; as well as 29 reconstructed tombstones and one greatly enlarged reconstruction of a signet, which serves as eye-catcher in the entrance hall of the virtual museum. The exhibition design, texts and room arrangements were elaborated by Dr. Barbara Kaufhold and will be thematically arranged throughout fourteen exhibition rooms. All rooms are in rectangular shape except for the octagonal entrance hall, an oval hallway and room fourteen, where we have design freedom to demonstrate stylistic devices for vivid room design. Room fourteen will host tombstones and provides access to outdoor areas.

Fleischhacker's pictorial estate was scanned and recorded as ViMCOX XML instances and has been, where applicable, enriched with additional descriptive metadata. For this purpose, we provided a simplistic metadata editor to automatize data acquisition: Generating thumbnail and texture, assigning unique identifiers, determining virtual dimensions of each digital copy and creating a suitable folder structure for further automated-processing. The virtual dimensions were predefined to have an area of no more than one square meter while preserving the original aspect ratio of each image. The metadata editor was implemented in C#. To map the ViMCOX schema definition to runtime classes, we used the Microsoft XML Schema Definition Tool included in the .Net framework (MSDN, 2011). The EPIDAT database developed and hosted by the Steinheim Institute, provides epigraphic information about Jewish tombstones, including descriptive information, inscriptions and transcriptions (Steinheim Institute, 2011). These additional metadata will be stored with the virtual tombstone objects to be visualized in the virtual world.

Ground plot sketches to roughly indicate the spatial design as well as positional information of the virtual objects were made available to us. Other initial parameters for object arrangement and interior design were defined as follows: The bottom edge of each image will be aligned using an invisible auxiliary line at 1.1m; information tableaux will be positioned at ground level; spacing between digital objects (images, tableaux) on each wall was set to 1.0m, while the spacing between doors and objects was set to 0.5m; and the initial wall height amounts 2.5m.

3.2 Dissemination

Automatic dissemination and deployment of virtual

museums on the web is part of our ongoing research. We introduced an automatized approach to assemble and distribute Web Application Archives (WAR) interfacing Apache ANT programmatically on application level at runtime in (Biella et al., 2012) and (Sacher, 2011). The generated WAR file layout is segmented in four parts: The Java libraries needed to do the schema mapping and to generate 3D content—the XML processor and the Replicave framework libraries for generating X3D and X3Dom, ViMCOX XML instances specifying virtual exhibitions; JavaServer Pages (JSP) to interface the Java APIs to visualize X3D and X3Dom prospectively. It also has all required resources, such as X3D models (exhibits and assets), textures, multimedia objects and other user content. The intended dissemination and presentation workflow can be described as follows: Automatic or user-generated ViMCOX XML instances are gathered from the content creator's workspace to determine and assemble all required resources. This includes normalization of path statements to match web-application conventions and relative file path statements. The virtual museum will be generated on server side at runtime. Metadata instances are processed and the virtual museum will then be generated and presented in accordance with the selected visualization method (X3D or X3Dom).

4 CONCLUSIONS

In general, a metadata-based modeling approach has the flexibility to exchange authoring software, metadata processors or visualization frameworks. The APIs of the MetaToolkit and Replicave enable the generation of exhibition space and facilitate automatic distribution of exhibits. Automatized spatial design is not limited to rectangular or regular n -sided polyan-gular room shapes but can also utilize point lists representing ground plots or the topographic maps of rooms and buildings. Procedural modeling based on native metadata still requires knowledge of software engineering and it is mainly designed for experienced users and IT experts. We believe that exhibition design is a iterative and creative process. Our (semi-)automatic approach is suitable for larger content bases and when regular polygonal room shapes are desired. The supported room shapes are appropriate for virtual museums without extravagant architectural design. In addition, we support the combination of manually created, auto-generated, prototyped or dummy content, which can be replaced during the design process. The generated exhibition rooms serve

as an initial starting point for further manual processing and visual design. This includes the assignment of suitable textures, color coordination, fine-tuning of object positions and scale to create more lifelike exhibition rooms and to suit the curator's preferences and perceptions. We are targeting to assess our approach, by comparing the users experience perceived in different types of museum exhibitions comprising the same content on display: real museum exhibitions, equivalent 3D reconstructions and (semi-)automatically generated museums with and without post-processing.

In future prospects we want to combine recommender systems—taking into account user profiles including demographic data, user's content and architectural preferences, their experience in 3D worlds (addressing navigation aid or modes as well as available object interaction modes) or their ability to learn—to generate different exhibitions, assemble different virtual museum tours as well as adding suitable educational or supplicant material to support the virtual museum visitor. A architecture for adaptive virtual museums was proposed in (Lepouras and Vassilakis, 2004). Another application scenario is the generation of fictional exhibitions using content filtering based on metadata categories such as epochs, artistic style, object material, artists, collections or multimedia content.

The main drawback of the current development track is that X3Dom does not support all the X3D sensor nodes that were available in native X3D, such as the pointing device sensors *SphereSensor*, *CylinderSensor* and *PlaneSensor*. These sensor nodes were used to implement user-to-object interaction, including rotating, scaling and translating exhibits. Another shortcoming in comparison with X3D plugins is the switching and selecting of viewpoints or even the animated transition between multiple viewpoints of a scene (viewpoint tour/camera path animation). This feature set was formerly accessible using the built-in interfaces of X3D browser plugins. These features are not part of the X3Dom specification and are therefore designated to be implemented by web developers. Thus, we need to re-implement equivalent behavior using JavaScript.

The proposed automatization process for publishing exhibitions on the web requires manual selection of the desired visualization method and format. A more convenient way would be to detect client capabilities using JavaScript, i.e., to detect the WebGL rendering context or X3D browser plugins to provide the appropriate visualization method based on client capabilities.

Moreover, Replicave generates 3D content server-sided and requires a servlet environment to deliver

web-based exhibitions. We are not currently considering a JavaScript-based port of Replicave due to the broad feature set of ViMCOX and the great effort conceivably required to implement complete metadata mapping using JavaScript and JSON.

When generating virtual museums, Replicave serves the same generated content and provides almost the same viewing experiences using X3D or X3Dom. Nevertheless, X3D-capable browser plugins, like BSContact (Bitmanagement Software GmbH, 2012), support per-vertex lighting; thus, the shading quality of lit objects will increase depending on the object's polygon count. On the other hand, X3Dom supports per-pixel lighting and therefore provides different viewing experience when light sources are added to the scene.

The new development track using X3Dom as a second visualization method is a further step towards multiplatform—desktop, web and mobile—virtual museums. Initial tests to load generated virtual museums on mobile platforms succeeded. For testing purpose we used a Google Nexus 7 tablet running Android 4.1.2 and Firefox 16.0.2. We observed that the X3D navigation mode *EXAMINE* works well using touch input. However, the required X3D navigation mode *WALK*—to simulate walkthroughs—currently does not match the user experience provided by X3D plugins.

ACKNOWLEDGEMENTS

Many thanks to our project partners from the Steinheim Institute and our students for valuable summaries of their diploma theses and for carefully implementing the software, building virtual museums as well as testing many models and examples.

REFERENCES

- Ancona, M., Drago, S., Quercini, G., and Bogdanovych, A. (2007). Rectangular dualization of biconnected planar graphs in linear time and related applications. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.3445>. [Online; accessed 31.01.2013].
- Behr, J., Eschler, P., Jung, Y., and Zöllner, M. (2009). X3DOM: A DOM-based HTML5/X3D integration model. In *Proceedings of the 14th International Conference on 3D Web Technology, Web3D '09*, pages 127–135, New York, NY, USA. ACM.
- Behr, J., Jung, Y., Drevensek, T., and Aderhold, A. (2011). Dynamic and interactive aspects of X3DOM. In *Proceedings of the 16th International Conference on*

- 3D Web Technology, Web3D '11, pages 81–87, New York, NY, USA. ACM.
- Biella, D., Luther, W., and Baloian, N. (2010a). Beyond the ARCO standard. In *16th International Conference on Virtual Systems and Multimedia (VSMM)*, pages 184–191.
- Biella, D., Luther, W., and Baloian, N. (2010b). Virtual Museum Exhibition Designer Using Enhanced ARCO Standard. In *XXIX International Conference of the Chilean Computer Science Society (SCCC)*, pages 226–235.
- Biella, D., Luther, W., and Sacher, D. (2012). Schema migration into a web-based framework for generating virtual museums and laboratories. In *Proceedings of the International Conference on Virtual Systems and Multimedia (VSMM 2012)*, Milan, Italy, IEEE CS Press, pages 307–314.
- Bitmanagement Software GmbH (2012). BS Contact. <http://www.bitmanagement.com/de/products/interactive-3d-clients/bs-contact>. [Online; accessed 05.11.2012].
- Coburn, E., Light, R., Mckenna, G., Stein, R., and Vitzthum, A. (2010). LIDO: Lightweight Information Describing Objects, Specification Version 1.0.
- J. Benner, A. Geiger, K. Leinemann (2005). Flexible generation of semantic 3d building models. *Technical Services Quarterly*, pages 17–22. [Online; accessed 05.11.2012].
- Jung, Y., Keil, J., and Behr, J. (2012). Demo: X3DOM Declarative (X)3D in HTML5. <http://www2012.wwwconference.org/proceedings/nocompanion/Dec3D2012.01.pdf>. [Online; accessed 05.11.2012].
- Lepouras, G. and Vassilakis, C. (2004). Adaptive virtual reality museums on the web. <http://www.uoc.edu/humfil/articles/eng/pujol0304/pujol0304.pdf>. [Online; accessed 31.01.2013].
- MSDN (2011). XML Schema Definition Tool (Xsd.exe). <http://msdn.microsoft.com/en-us/library/x6c1kb0s%28v=vs.100%29.aspx>. [Online; accessed 05.11.2012].
- Sacher, D. (2011). Ein webbasiertes Rahmenwerk für virtuelle Laboratorien und Museen: Schemaintegration und Implementierung. Diploma thesis, University of Duisburg-Essen.
- Savage-NPS (2012). X3D to X3Dom Stylesheet. <http://www.web3d.org/x3d/stylesheets/X3dToX3dom.xslt>. [Online; accessed 05.11.2012].
- Steinheim Institute (2011). Epidat - Epigraphical database. <http://www.steinheim-institut.de/cgi-bin/epidat>. [Online; accessed 05.11.2012].
- Tomas Trescak (2012). *Intelligent Generation and Control of Interactive Virtual Worlds*. PhD thesis, Universitat Autònoma de Barcelona.
- TU Graz (2010). Generative Modeling Language (GML). <http://www.generative-modeling.org/>. [Online; accessed 05.11.2012].
- Web3D-Consortium (2004). Information technology: Computer graphics and image processing – Extensible 3D (X3D), Part 2: Scene access interface (SAI). <http://www.web3d.org/files/specifications/19775-1/V3.0/index.html>. [Online; accessed 05.11.2012].
- Wolf, V., Song, Y., Sacher, D., Luther, W., and Biella, D. (2012). ViMCOX: Virtual Museum and Cultural Object Exchange Format, Specification Version 1.1. http://www.vimedea.com/wordpress/?page_id=40. [Online; accessed 05.11.2012].
- X3D-Graphics (2012). <http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter08-UserInteractivity/Doors.xhtml>. [Online; accessed 05.11.2012].

APPENDIX

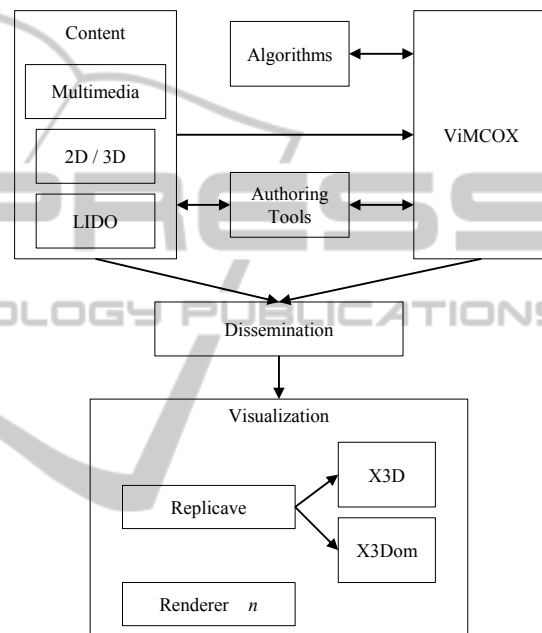


Figure 1: Abstract architectural overview.