

Integration of CityGML and Collada for High-Quality Geographic Data Visualization on the PC and Xbox 360

Marc Herrlich, Henrik Holle, Rainer Malaka

► **To cite this version:**

Marc Herrlich, Henrik Holle, Rainer Malaka. Integration of CityGML and Collada for High-Quality Geographic Data Visualization on the PC and Xbox 360. Hyun Seung Yang; Rainer Malaka; Junichi Hoshino; Jung Hyun Han. 9th International Conference on Entertainment Computing (ICEC), Sep 2010, Seoul, South Korea. Springer, Lecture Notes in Computer Science, LNCS-6243, pp.270-277, 2010, Entertainment Computing - ICEC 2010. <10.1007/978-3-642-15399-0_27>. <hal-01055635>

HAL Id: hal-01055635

<https://hal.inria.fr/hal-01055635>

Submitted on 13 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Integration of CityGML and Collada for High-Quality Geographic Data Visualization on the PC and Xbox 360

Marc Herrlich, Henrik Holle, and Rainer Malaka

Digital Media Group
TZI, University of Bremen
Germany

`mh@tzi.de`, `henrik@holle.net`, `malaka@informatik.uni-bremen.de`

Abstract. Computer games and serious geographic information systems (GIS) share many requirements with regard to storage, exchange, and visualization of geographic data. Furthermore, there is a demand for high-fidelity photo-realistic and non-photo-realistic visualization. This poses at least two questions: Is there a single data format standard suitable for serious GIS-based applications and computer games that supports state-of-the-art visual quality? How can computer games and serious applications benefit from each other, especially platform-wise? In this paper we will investigate both questions by taking a closer look at the CityGML standard in comparison to COLLADA and we will report on our findings in integrating CityGML with mainstream game technology. The main contribution of this paper to the field is a suggested way of integrating an important features of CityGML and Collada for high-quality visualization, i.e. programmable shader effects, and demonstrating the feasibility of employing a game console as a cheap and widely available device for geodata visualization and possibly other geodata-centric applications.

1 Introduction

Currently, even in areas of mainstream information technology the terms *Serious Games* and *Serious Gaming* seem to be ubiquitous, albeit often used in a somewhat general manner. Even so, this development shows that the industry and the scientific community alike have come to recognize the benefits of employing game technology for other purposes. Besides research and applications directly targeted at exploiting the motivational and educational aspects of games, the scientific community values the comparatively cheap and powerful platform for mainstream hardware provided by game technology and the industry clearly appreciates the cost benefits, especially in areas where dedicated high-performance solutions are/were predominant, e.g. in military simulation. While there is an ongoing discussion among practitioners and theorists if the term *Serious Games* really applies to the general usage of game technology for non-entertainment applications that have no or very limited game-like appeal, this question is only of

limited importance for the scope of this paper and therefore we will not directly address it. For reference, we point to the taxonomy of Serious Games by Sawyer [1]¹. In our opinion the “seriousness” and/or “gameiness” of any game or application very much depends on the context and intention of usage and perception of the user.

Games as applications tend to quickly follow and adapt developments from other areas, e.g. trends like user-generated content or the interest in geographic information systems (GIS), sparked by applications such as Google Earth² or Microsoft’s Virtual Earth³. As game technology is of rising relevance to many other areas of information technology, it is equally important to investigate how both sides can benefit most from each other and what standards and techniques are available or needed. In this paper we will concentrate on the area of GIS-based applications. Serious GIS-based applications and games both have a high demand for *direct* geodata, i.e. geodata resulting from measurements and suitable post-processing, and *indirect* geodata, i.e. geodata that is the result of a modeling process that might or might not have any relations to the real world. Although, in serious applications most often direct data is dominant and in games indirect data is prevailing, it is often the case, that games also include direct data, e.g. in sport, racing, or simulation games, and that serious applications make use of indirect data, e.g. Google Earth. Therefore, it is consequential to take a closer look at the requirements of both applications and discuss the standards and techniques currently used or proposed. For the scope of this paper we will focus on discussing the possibilities, standards, and techniques of using direct geodata for games and applications based on game technology and show how to easily integrate the foundation for state-of-the-art rendering, i.e. so-called “shader effects”, with CityGML as a geodata standard. Furthermore, our prototype also runs on the Xbox 360, a so-called “next-gen” gaming console with a large user-base.

The rest of this paper is organized as follows. In section 2 we present some related works. In section 3 we discuss essential features of two important standards for geodata, i.e. CityGML and Collada, with regard to the requirements of applications concerned with high fidelity visualization. In the same section we suggest a small but important extension for CityGML in order to remedy one of the major shortcomings we discovered in our experiments. A prototype we created based on the XNA framework is presented in section 4. In section 5 we summarize and reflect our findings and conclude the paper by outlining open questions and future research directions.

2 Related Work

Already in 2002, Rhyne [2] argues that scientists today had much to learn from the computer games industry regarding computer graphics, visualization, and

¹ Conference presentation available online at www.seriousgames.org

² earth.google.de

³ www.microsoft.com/virtualearth

interfaces. In Rhyne’s opinion, scientific visualization applications could benefit from computer game technology, as most computer games are optimized for commodity hardware. She also states some drawbacks in using game engines with respect to data accuracy and reliability. Since 2002 computer games have surely continued to push the state-of-the-art in computer graphics and additionally we think that many scientific applications could also benefit from many of the interaction metaphors used in computer games, although this is not the focus in this paper. Nearly all computer games with a rich graphical world rely on so-called shader effects, which are “short” programs most often written in a special C-like language. We show in this paper how to include such programs, which are essential for high-quality real-time graphics today, into semantically rich data standards for geographic data.

Herwig and Paar [3] discuss the suitability of game engines for landscape visualization and planning. They present different usage scenarios and analyze the requirements of landscape architects concerning supporting tools and to what extent game engines can solve these problems. Fritsch and Kada [4] discuss indoor as well as outdoor visualization of geodata based on different game engines. They also compare the benefits of game engines in contrast to other software libraries and present concepts for integrating them with other tool for purposes like computer aided facility management. They conclude, that the conversion process of geodata into the data format of a game engine was one of the major obstacles for such applications. Both the works of Herwig and Paar and of Fritsch and Kada clearly demonstrate the suitability of computer game technology for general visualization and especially for geodata visualization applications. However, while theirs and some similar works deal with some of the fundamentals, they do not concern themselves with integrating essential features directly at the data format level, nor do they include game consoles as a platform.

Zeile et al. [5] present a case study of creating and visualizing a virtual 3d city model of Bamberg. Their report clearly demonstrates the complexity of integrating the huge number of different data sources. They also state that “aesthetic ambitions” and “degree of detailing” had been important criterions. This supports our assumption of a demand for high-quality real-time graphics in this application area, which is the domain of computer games employing shader programs.

Döllner et al. report in [6] and [7] on the backgrounds of their LandXplorer system and their usage of CityGML. In the presented general system architecture, game engines and our work would fit in well as “3D City Model Presentation Systems” component.

Kibria et al. discuss in [8] the requirements of virtual environments based on geodata. They suggest, that visual fidelity and - more general - realistic surface materials were very important for the users of such environments, which might require the additions to geodata standards we discuss in sections 3 and 4 of this paper.

3 Geodata Standards

In recent years, the demand for three dimensional city models has increased considerably. More and more data of cities, regions, and states is captured and evaluated. Based on this data, three dimensional city models are created and procured. Out of this development process many standards for geodata storage, exchange, and visualization have evolved. Some standards such as the Geography Markup Language [9] [10], Keyhole Markup Language [11], digital ground models (DGM), or ESRI Shapefiles [12] focus on the exchange of geometric data with geographic references. Other standards like X3D [13] or Collaborative Design Activity (COLLADA) [14] have been developed for the exchange of generic 3d scenery. In addition to topological and geometrical data, city models include a semantic model. The common data exchange standards like GML, X3D or COLLADA do not feature this. The comparatively new format CityGML [15] targets the problem by unifying geometric data and semantics into one data exchange format. Therefore, CityGML provides a decent format for the efficient and lossless data exchange within geodata content pipelines and is increasingly used by (German) land surveying offices. The use of the CityGML format enables a high degree of semantic and syntactic interoperability between applications. However, while CityGML models might include very rich semantic information, it is not clear if CityGML provides sufficient support for applications with high demands for photo-realistic visualization. The following paragraphs analyze the format regarding this aspect of CityGML, i.e. as a basis for applications generating high fidelity visualization.

3.1 CityGML

The Open Geospatial Consortium Inc. (OGC) has adopted CityGML 1.0.0, the open data model developed for the storage and exchange of virtual 3d city models, as an official standard. It is implemented as an Geography Markup Language (GML) application schema and uses only a subset of the complex GML profile. Unlike formats such as X3D or COLLADA, CityGML does not only represent the graphical appearance of city models but allows the modeling and representation of the semantic and thematic properties, respectively, i.e. taxonomies and aggregations of digital ground models, buildings, vegetation, water bodies, transportation facilities, and city furniture. The integration of geometric and thematic data into one format allows to utilize 3d city models for more use cases aside from pure visualization. The CityGML model distinguishes between four different level-of-details, from a simple block model (derived by extruding ground planes), up to architectural building models (out- and indoor).

3.2 Extending CityGML

As mentioned before, serious GIS-based applications and games both have a high demand for *direct* geodata. CityGML, the 3d city model exchange format, could be the data source for *direct* geodata for visualization purposes. The following

paragraph compares the geodata exchange format to the general and widely used 3d scenery exchange format COLLADA regarding its applicability to visualization purposes. On closer inspection, the CityGML data model can be divided into three different types of data: geometrical and thematical information and data concerning the appearance of objects.

CityGML's geometric model is derived from GML and mainly uses the basic type "polygon", which can be combined to complex surfaces. A thematic feature type, e.g. roof or door, can be assigned to each surface type. The modeling approach follows the boundary representation concept [16]. COLLADA's geometric modeling approach also follows the boundary representation concept but additionally supports polygonal meshes and modeling techniques closely related to modern graphic hardware programming interfaces.



Fig. 1. Prototypical textures derived from the thematical model

The thematical model is a unique feature in CityGML. COLLADA and other formats for the exchange of general 3d scenery do not feature this. They exclusively offer the modeling of geometrical data and surface properties, i.e. textures, materials, and (hardware) shader programs. Applications focusing on visualization have no direct need for the thematical model. Nevertheless, if thematical data is available, it can be used to enrich the presentation in case the appearance model, i.e. surface data, is missing or incomplete. Colors or prototypical textures could be derived by feature type (figure 1), e.g. prototypical textures for land use and roads, or coloration of roof and wall feature types. Instead of using photo-realistic rendering, non photo-realistic rendering techniques [17] can also be applied.

An area of concern regarding CityGML is the appearance model. Compared to general 3d scenery exchange formats, CityGML's appearance model just allows minimal modeling of presentation parameters. The current CityGML version only supports three different types of texture mapping and a very basic lighting and material model. COLLADA for example allows the definition of complex materials including shader programs or shader fragments using its COLLADA FX extension. For high-fidelity photo-realistic visualizations the use of shading techniques like bump or normal mapping are common. The limitation can be solved by combining COLLADA FX and CityGML using CityGML's application domain extension (ADE) functionality. The COLLADA FX material library can be embedded into a CityGML document. A new appearance type derived from the original appearance type can reference the COLLADA FX material. This extension allows backwards compatibility to existing geodata infrastructures on the one hand, and allows the definition and lossless exchange of complex material definitions on the other hand.

The implementation details of a CityGML ADE for ColladaFX are quite straight forward (albeit tedious because of the lengthy ColladaFX specification). Basically, the respective ColladaFX schema definitions will have to be imported into a common schema definition file. Assuming such a common schema definition, Collada materials based on shaders effects could be used by defining a special CityGML surface type. This surface type could then be assigned to CityGML features that shall be rendered with shader effects.

4 Geodata Visualization on the Xbox 360

We created a sample implementation for the visualization of CityGML models using the Microsoft XNA framework. The XNA framework is widely used by hobbyists and even an increasing number of professionals to create games for the PC but also for the Xbox 360 gaming console. Furthermore, it is more and more used as a managed graphics API for generic applications, including scientific visualization and other serious applications. An important feature of the XNA framework is the so-called "Content Pipeline", which provides a clean interface and workflow for using all kinds of assets, i.e. 3d models, textures, sounds, and more.

We extended the XNA Content Pipeline⁴ to enable the easy import and conversion of CityGML files. XNA content importers and processors for CityGML have been implemented. The importer loads the CityGML file, validates it against the schema specification and transforms it into a more general XML DOM representation. The processor transforms the CityGML model into an optimized format for the XNA graphics interface and performs important steps, like tessellating polygonal data to primitives stored in vertex and index buffers, generating texture coordinates if needed, creating/loading shader effects for lighting and texturing, and partitioning the data using common data structures like quad

⁴ msdn.microsoft.com/en-us/xna/default.aspx

trees. The resulting data can be read and used by any XNA application. Internally, the CityGML processor is divided into several sub-processors for specific categories of CityGML objects, e.g. buildings, water, or relief features.

```
//select all top level featuretypes via xpath
XmlNodeList featureTypeNodes =
    xmlDOM.SelectNodes("CityModel/cityObjectMember");

foreach(XmlNode xmlNode in featureTypeNodes)
{
    //Create processor via factory pattern
    CityObjectProcessor processor =
        CityObjectProcessorFactory.MakeProcessor(xmlNode);
    //set processing content node to current featuretype
    processingContext.Node = xmlNode.LastChild;
    //start processing
    processor.Process(processingContext);
}
```

Fig. 2. XNA CityGML Processor

A factory pattern is used to create the matching processor directly from the XML DOM representation using XPATH queries for retrieving top level features (figure 2).

5 Discussion and Future Work

Because our sample utilizes the XNA Content Pipeline, it runs not only on the PC platform but also enables the use of CityGML based applications on the Xbox360 platform.

Compared to common 3d scenery exchange formats, CityGML is not designed for high fidelity visualization applications due its rudimentary presentation model. The integration of COLLADA FX into CityGML as an application domain extension solves this issue and combines the advantages of both standards. The introduced XNA integration verifies on the one hand that CityGML data can easily be transformed to a real-time rendering friendly format using the XNA Content Pipeline and on the other hand that mainstream game technology is well suited as basis for serious GIS applications. There are many possible directions for future work. The questions remains, how to utilize the (theoretically) rich semantic information for entertainment applications, e.g. by enriching the game play based on the semantics of objects/areas in the world. Furthermore, it could be useful to integrate additional features from the COLLADA side, like COLLADA objects or physics. We would also like investigate further about the acceptance/demand and feasibility of serious GIS-based applications on the Xbox360 and other game platforms.

References

1. Sawyer, B., Smith, P.: Serious games taxonomy. Presentation at the The Serious Games Summit at the Game Developers Conference 2008 (2008)
2. Rhyne, T.M.: Computer games and scientific visualization. *Commun. ACM* **45** (2002) 40–44
3. Herwig, A., Paar, P.: Game engines: Tools for landscape visualization and planning? In Buhmann, E., Nothhelfer, U., Pietsch, M., eds.: *Trends in GIS and Virtualization in Environmental Planning and Design*, Wichmann, Heidelberg (2002)
4. Fritsch, D., Kada, M.: Visualisation using game engines. *Archiwum ISPRS* **35** (2004)
5. Zeile, P., Schildwächter, R., Poesch, T., Wettels, P.: Production of virtual 3d city models from geodata and visualization with 3d-game engines. a case study from the unesco world heritage city of bamberg. In Buhmann, Paar, Bishop, Lange, eds.: *Trends in Real-Time Landscape Visualization and Participation*, Wichmann, Heidelberg (2005)
6. Döllner, J., Baumann, K., Buchholz, H.: Virtual 3d city models as foundation of complex urban information spaces. In: *Proceedings of CORP 2006 & Geo-Multimedia 06*. (2006)
7. Döllner, J., Kolbe, T.H., Liecke, F., Sgouros, T., Teichmann, K.: The virtual 3d city model of berlin - managing, integrating, and communicating complex urban information. In: *Proceedings of the 25th Urban Data Management Symposium UDMS 2006*. (2006)
8. Kibria, M.S., Zlatanova, S., Itard, L., van Dorst, M.: GeoVEs as Tools to Communicate in Urban Projects: Requirements for Functionality and Visualization. In: *3D Geo-Information Sciences*. Springer Berlin Heidelberg (2008) 379–395
9. Open Geospatial Consortium Inc.: *OpenGIS Geography Markup Language (GML) Encoding Standard*. (2007) White Paper.
10. Kolbe, T.H.: Representing and exchanging 3d city models with citygml. In: *Lecture Notes in Geoinformation and Cartography*, Springer Berlin Heidelberg (2008) 15–31
11. Open Geospatial Consortium Inc.: *OGC KML*. (2008) White Paper.
12. Environmental Systems Research Institute: *ESRI Shapefile Technical Description*. (1998) White Paper.
13. ISO: *ISO/IEC 19775:2004 Information technology Computer graphics and image processing Extensible 3D*. (2005) White Paper.
14. Khronos Group: *COLLADA 1.5.0 Specification*. (2008) White Paper.
15. Open Geospatial Consortium Inc.: *City Geography Markup Language (CityGML) Encoding Standard*. (2008) White Paper.
16. Foley, J.D., VanDam, A., Feiner, S.K.: *Computer Graphics: Principles and Practice in C*. Addison-Wesley Longman (October 1995)
17. Lansdown, J., Schofield, S.: Expressive rendering: a review of nonphotorealistic techniques. *Computer Graphics and Applications, IEEE* **15** (1995) 29–37