FACULTY OF ENGINEERING
AND ARCHITECTURE

Pre-culling geometric linked building data for lightweight viewers

Linked Data

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in de ingenieurswetenschappen: architectuur

Supervisor: Prof. ir.-arch. Paulus Present

Counselors: Ir.-arch. Jeroen Werbrouck
Prof. dr. ir. arch. Ruben Verstraeten

Philippe Soubrier 01702837 philippe.soubrier@ugent.be
Academic year: 2022–2023

GHENT
UNIVERSITY

Contents

List of Figures

Short abstract

This is my short abstract.

Abstract

This is my abstract

Chapter 1

Introduction

1.1 Linked Data

-> what is linked data

-> what is a graph

1.1.1 RDF and triples

-> what is a triple (subject, predicate, object)

-> what is a turtle file

-> what is an rdf file

1.1.2 Ontologies and reasoning

-> what is an ontology

-> what is inference

-> what is a reasoner

1.1.3 Triplestores and SPARQL

-> what is a triplestore

-> what is SPARQL

-> what is a SPARQL endpoint

1.2 Context

1.2.1 3D viewers

-> Applications?

-> Who uses them?

-> What for?

1.2.2 BIM geometry

" In recent years, there has been a shift from vision to realization regarding the use of Building Information Models (BIM) within the architecture, engineering and construction (AEC) industries. Using modern modeling tools, such as Revit Architecture, ArchiCAD or Tekla Structures, the content produced by architects, designers and engineers has evolved from traditional 2D-drawings, sketches and written specifications to parametric, object-oriented 3D-models embedded with information to describe any building or facility in detail. As a digital representation of the physical and functional characteristics of a building, a BIM serves as a repository of information supporting a multitude of applications along the design and construction processes, including cost-estimation, energy analysis and production planning. " Eastman, 2008

The 3D model of a building consists of a multitude of sub-models, describing objects for all the different stakeholders participating in the project. Some describe very large objects, and some very small parts. Both can be defined in their most simple and abstract form or have an intricate and complex geometry. As a basic example, can a door simply be defined as a box, or up to the level of the screw-thread for the hinge system. The level of abstraction is here described as the Level of Detail (LOD), it is most of the time pre-selected for the needs of a Building Information Modelling (BIM) model, and is applied throughout a single model.

As shown in Figure 1.1, a standard BIM workflow goes through multiple phases each with their associated model and LOD. The LOD is a very important concept in the Architecture, Engineering and Construction (AEC) industry, as it allows for a very efficient workflow. Approaching the modelling step from a top-down perspective, starting with rougher geometries describing the rougher ideas of a concept model and evoluting to a more refined model for the construction phase. As last and longest standing model, can a higher LOD be used to describe subtle changes in the evolution of a building.

This amount of data, both accounting for the
-> Some data BB models

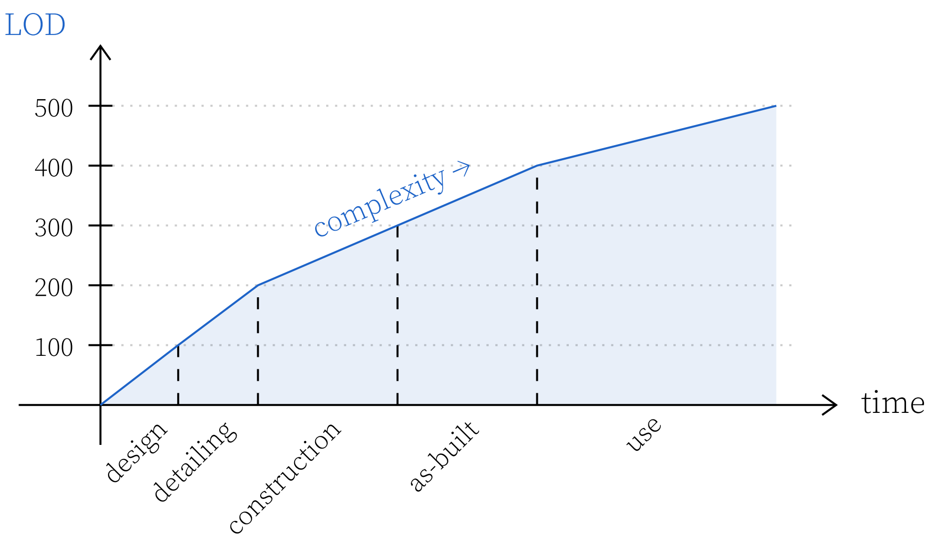

Figure 1.1: Evolution of LOD during the life-cycle of a building.

1.2.3 LDBIM

The interconnectivity of semantics can also be applied to geometry descriptions. Which could allow the co-existence of multiple LODs in a single model. Besides storing the evolution of a single element's geometry, it allows the linking of the different LODs to each other. In contrary to standard BIM models, as explained in 1.2.2.

1.2.4 Computing power dilemma

The enrichment of the Linked Data BIM (LDBIM)-graph also comes with a cost. The amount of data that needs to be stored and processed is much larger than a standard BIM model. Viewers greatly suffer from enrichment as most standard applications require the full model to be loaded in memory.

Although most office computers used in the AEC industry are powerful enough to handle large models, the hardware used in the field is not. The most common hardware used in the field is a tablet or a laptop. Both of these are not powerful enough to handle the amount of data required to display a LDBIM model. This is a problem as these low-power devices are the only solution when it comes to mobility. Furthermore, the size of the models that are more and more enriched can, in some cases, pose a problem to the hardware of a standard office computer.

1.3 Research questions

-> Taking advantage of database approach instead of file to avoid excess storage on high computational tasks

It can be inferred from 1.2.4 that filtering is necessary in order to visualize the geometry of a LDBIM model, due to its complexity and size. This filtering step, as shown in Figure 1.2, is also known as culling in 3D computer graphics. Culling is the process of removing objects, or parts of objects, from the scene that are not visible to the user. This is done to reduce the amount of data that needs to be processed and stored. Implementing such technology in the context of LDBIM is the proposal of this thesis. The main goal being the introduction of similar algorithms within this context. As culling algorithms are not new and part of a field of research in continuous expansion.

The research questions are therefore focused on the feasibility of this introduction. And propose a set of possible solutions tailored to this specific problem, while highlighting possibilities for future research and specific use cases.

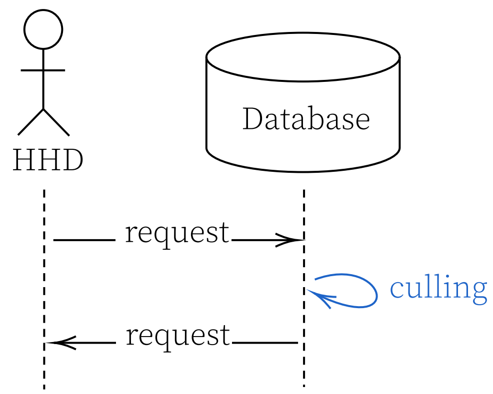

Figure 1.2: Basic principle

Figure 1.2 illustrates the basic idea of this thesis. Being the extra step inside the communication between a user, here represented as a Hand Held Device (HHD), and a database storing the model. A HHD has been chosen to illustrate a low-powered device used in the field, that requires a lightweight 3D viewer to visualize and explore the digital twin of the building. The HHD is assumed to have no knowledge of the LDBIM model, and only receives the geometry that is required to be displayed from the database. On the other hand, the database is assumed to have, or access to, all the knowledge of the model and the needed semantic to perform the culling.

1.3.1 To which extent can LDBIM geometry be culled to be streamed to lightweight viewers?

This thesis focuses on cumputing with data snippets or triples inside a LDBIM model, not wihtin. Meaning that the smallest unit of data that can be culled is the one described in one triple, in the most likely scenario: a single LOD of a single element. It implies that geometry is defined and separated at the object-level. It also implies that culling thechniques such as back-face culling will not be handled in this thesis, and will be left to the viewer itself, not the database.

Which snippets of data are needed by the viewer? Is part of the question. The basic needs of the viewer consists firstly of the geometry itself, selecting the right geometry format for the application as well as the additional visual information such as color, texture, etc. Secondly, the identifier of each element is of crucial importance to maintain the link to other semantic ressources in the graph. Enabling the viewer to retrieve those ressources for a multitude of usecases. Transforming it into a user-friendly visual query tool.

The impact of the culling on the performance of the viewer will greatly determine the mutl

1.3.2 Can existing semantic and ontologies be used to feed possible culling algorithms?

-> What are ontologies?

The

1.4 Research objectives

1.4.1 Bring forward the advantages of LDBIM for visualization of big 3D models

-> Showcase that existing models are already mature enough for these usecases.

->

1.4.2 Showcase the feasibility of LDBIM for visualization of big 3D models

->

Chapter 2

State of the art

-> Introduction to the chapter, both tools and existing approaches to create overview of the state of the art

2.1 Related Technologies and Tools

2.1.1 Qoniq and LOD Streaming for BIM

-> Why I chose this one (why special)

-> LOD streaming principle

-> Probably present it as a goal but in the older framework (for effectiveness(esthetics and performance))

In unity, the maximum amount of ram cannot exceed 2Gb[1]

2.1.1.1 Qoniq's approach to LOD streaming

(T. Strobbe, personal communication, November 25, 2022)

2.1.1.2 Advantages and challenges

2.1.1.3 Potential applicability to LDBIM

2.1.2 ld-bim.web.app

https://ld-bim.web.app/

-> Where does it come from

-> Detailed explanation of the features / capabilities

[1]"Unity - Manual: Memory in Unity WebGL", n.d.

-> Detailed fragmentation of missed opportunities / how this thesis positions itself to it

2.1.3 AEC related ontologies

-> Highlighting maturity and usefull data

2.1.3.1 BOT

2.1.3.2 FOG and OMG

2.1.4 GIS related ontologies

-> Highlighting maturity and usefull data

2.1.4.1 geoSPARQL

-> 2D Limitations

2.2 Existing Approaches in BIM 3D Viewers and Visualization Techniques

- DDS CAD Viewer
- Tekla BIMsight
- Autodesk Navisworks
- Solibri Model Viewer

2.2.1 General Features

2.2.2 Interoperability

2.2.3 Scalability

2.2.4 Collaboration and Data Sharing

2.2.5 Customization and Extensibility

Chapter 3

Culling approaches

3.1 AEC related ontologies

3.1.1 BOT

3.1.2 FOG and OMG

3.2 GIS related ontologies

3.2.1 geoSPARQL

Chapter 4

Setup

4.1 Participants

This is a diagram:

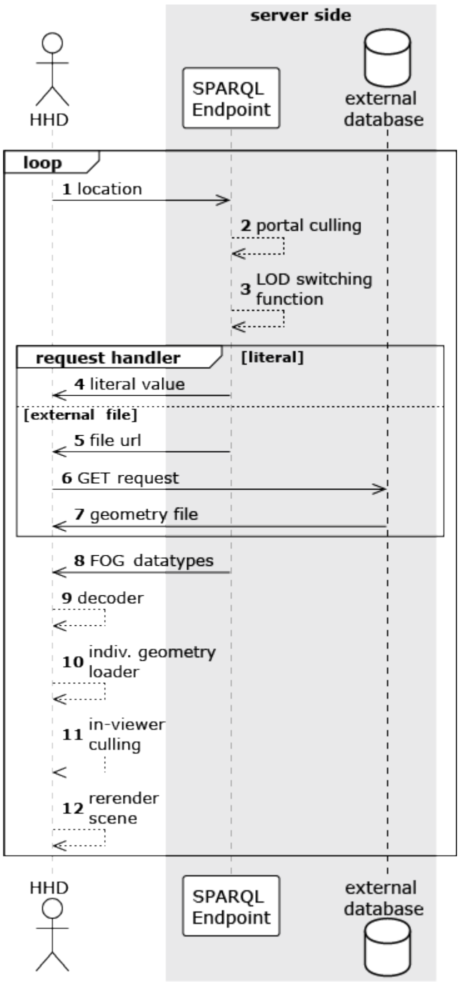

Figure 4.1: Sequence diagram

4.2 Framework

4.2.1 Nextjs

4.3 Querying

4.3.1 Front-end

4.3.2 Back-end

4.4 Rendering

4.4.1 Xeokit SDK

List of Acronyms

AEC	Architecture, Engineering and Construction	7
BIM	Building Information Modelling	7
BOT	Building Topology Ontology	
FOG	File Ontology for Geometry formats	
GIS	Geographic Information System	
HHD	Hand Held Device	9
LDBIM	Linked Data BIM	8
LOD	Level of Detail	7
OMG	Object Management Group	
SDK	Software Development Kit	

References

Eastman, C. M. (2008). Bim handbook : A guide to building information modeling for owners, managers, designers, engineers, and contractors. Hoboken NJ. https://lib.ugent.be/catalog/rug01:001359094

Referenced webistes

Unity - manual: Memory in unity webgl. (n.d.). https://docs.unity3d.com/2023.2/Documentation/Manual/webgl-memory.html