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

1 Introduction 6
1.1 Linked Data . . . 8
1.1.1 RDF and triples . . . 8
1.1.2 Ontologies and reasoning . . . 10
1.1.3 Triplestores and SPARQL . . . 11
1.1.4 Complexity of the graph . . . 11
1.2 Research questions . . . 12
1.2.1 Can LDBIM be culled? . . . 12
1.2.2 Can existing semantics be used? . . . 13

2 State of the art 14
2.1 Related Technologies and Tools . . . 14
2.1.1 Qoniq and LOD Streaming for BIM . . . 14
2.1.2 ld-bim.web.app . . . 14
2.1.3 AEC related ontologies . . . 15
2.1.4 GIS related ontologies . . . 15
2.2 Existing Approaches in BIM 3D Viewers and Visualization Techniques . 15
2.2.1 General Features . . . 15
2.2.2 Interoperability . . . 16
2.2.3 Scalability . . . 16
2.2.4 Collaboration and Data Sharing . . . 16
2.2.5 Customization and Extensibility . . . 16

3 Culling approaches 17
3.1 AEC related ontologies . . . 17
3.1.1 BOT . . . 17
3.1.2 FOG and OMG . . . 17
3.2 GIS related ontologies . . . 17
3.2.1 geoSPARQL . . . 17

4 Setup 18
4.1 Participants . . . 18

4.2 Framework 19
4.2.1 Nextjs 19
4.3 Querying 19
4.3.1 Front-end 19
4.3.2 Back-end 19
4.4 Rendering 19
4.4.1 Xeokit SDK 19

References 21

Referenced webistes 22

List of Figures

1.1 Sequence diagram - basic concept 7
1.2 Illustration of culling principle, based on Cohen-Or et al., 2003 8
1.3 Triple structure 10
1.4 Evolution of LOD during the life-cycle of a building. Based upon the Macleamy Curve (Ilozor & Kelly, 2012) 11

4.1 Sequence diagram 18

List of Tables

1.1 Size of test-models in Johansson et al., 2015 7

Short abstract

This is my short abstract.

Abstract

This is my abstract

Chapter 1

Introduction

From 2D, to 3D and now to BIM. The evolution of the Architecture, Engineering and Construction (AEC) industry has been a long and complex one. The introduction of 3D modeling was the first major step in the industry's evolution, as it allowed for more accurate representations of buildings. No longer solely relying on 2D drawings, a 3D model of a building can be used to create various representations, from a simple 2D floor plan to a full 3D model. Following the adoption of 3D modeling, the implementation of Building Information Modelling (BIM) emerged as another significant milestone. BIM adds an extra layer of information on top of the 3D model. As the digital representation of a building's physical and functional characteristics, BIM serves as a repository for semantics originating from various applications throughout the design and construction processes, including cost estimation, energy analysis, and production planning.
-> needs refs to support point (about infustries in AEC)

However, as mentioned in Werbrouck, 2018, the next challenge for the AEC industry is related to the ecosystem of current BIM softwares, which remains closed off to other disciplines. This data management challenge is currently being adressed by the Linked Building Data Community Group and other research entities, such as the University of Ghent, through the use of Web of Data technologies[1]. This emerging milestone will be discussed in this thesis under the term Linked Data BIM (LDBIM).

Proposal

Each of these evolutions has brought, and will continue to bring, a significant amount of data together. This volume is expected to grow exponentially in the future as the industry shifts towards a more digital approach and opens up to other stakeholders. The data graphs will not only expand in terms of semantics but also in geometry. This makes visual querying, or simply put, 3D exploration of models, an increasingly diffi-

[1] W3C, 2023.

cult task. Especially when looking at newer devices used in the industry such as mobile phones, and tablet, which are becoming more and more powerful, but still have limited computational resources in comparision to office computers.

To bring this volume of geometric data in perspective, Table 1.1 shows the size of the test-models used in Johansson et al., 2015 , a study from 2015 on the performance of BIM viewers for large models with the following description:

" Although the Hotel model contains some structural elements they are primarily architectural models. As such, no Mechanical, Electrical or Plumbing (MEP) data is present. However, all models except the Hospital contain furniture and other interior equipment. " (Johansson et al., 2015)

Model	# of triangles	# of objects	# of geometry batches
Library	3 685 748	7318	11 195
Student House	11 737 251	17 674	33 455
Hospital	2 344 968	18627	22 265
Hotel	7 200 901	41 893	62 624

Table 1.1: Size of test-models in Johansson et al., 2015

These models demonstrate how basic BIM models can already contain a significant amount of data. LDBIM will not only bring together new stakeholders but also have the capability to keep track of multiple geometry versions for each object, should they occur. Therefore, this thesis proposes a new approach to the visual querying of LDBIM models, wherein viewers will not have to load the entire model into memory. Instead, after filtering at the source, only the geometry needed for the visual tasks at hand will be loaded, while maintaining the original link to each resource for further processing and use cases. This filtering step is commonly referred to as culling in the computer graphics industry and is illustrated in Figures 1.1 and 1.2.

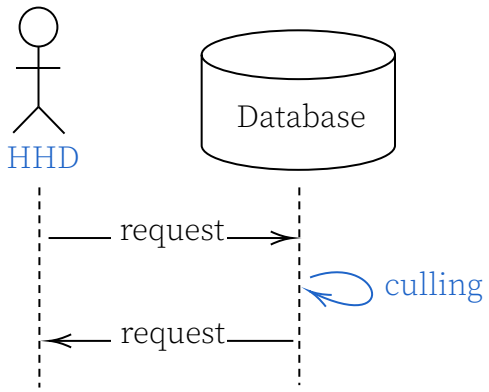

Figure 1.1: Sequence diagram - basic concept

Figure 1.1 illustrates the basic idea of this thesis, presenting an extra step in the com-

munication between a user, represented here by a Hand Held Device (HHD), and a database storing the model. An HHD has been chosen to exemplify a low-powered device used in the field, which 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 needs to be displayed from the database. On the other hand, the database is assumed to possess, or have access to, all the knowledge of the model and the necessary semantics to perform the culling.

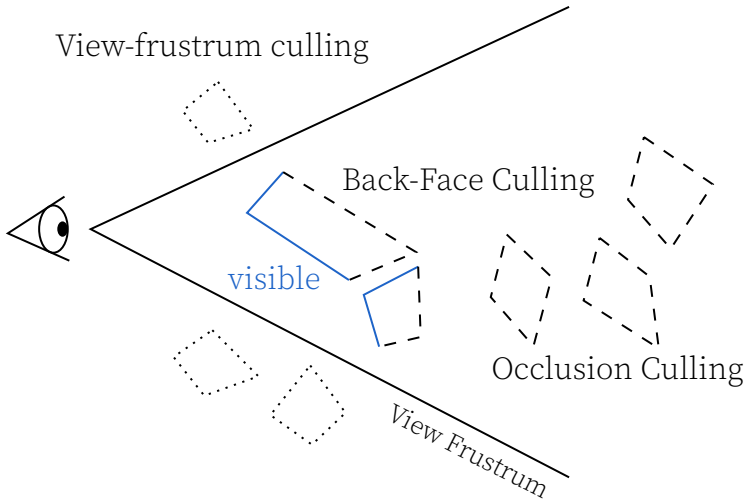

Figure 1.2: Illustration of culling principle, based on Cohen-Or et al., 2003

Figure 1.2 showcases multiple culling techniques to showcase some culling principles. The first technique, *frustum culling*, is used to determine which objects are visible to the user. The second technique, *occlusion culling*, is used to determine which objects are occluded by or behind other objects. And lastly, *back-face culling*, is used to determine which faces, and not whole objects, are facing away from the user.

1.1 Linked Data

As mentioned in the Introduction, the evolution from BIM to LDBIM is an evolution of the data *management* layer. "Linked Data", as stated by the World Wide Web Consortium (W3C), is a collection of interrelated datasets on the Web, formatted in a standard way that is accessible and manageable by Semantic Web tools. The same applies to the relationships among them.[2] The following collection of Semantic Web technologies explores the required environment to achieve this goal.

1.1.1 RDF and triples

At the core of the Semantic Web is the RDF, a data model for describing resources on the Web. RDF is a graph data model that consists of *triples*, which are statements about

[2] W3C, 2015b.

```
@prefix fog: <https://w3id.org/fog#> .
@prefix omg: <https://w3id.org/omg#> .
@prefix bot: <https://w3id.org/bot#> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix flupke:   <http://flupke.archi#> .

flupke:room1 rdf:type bot:Zone ;
    bot:containsElement flupke:coneOBJ ;
    bot:containsElement flupke:cubeGLTF ;

flupke:coneOBJ omg:hasGeometry flupke:coneOBJ_geometry-1 ;
    rdf:type bot:Element .

flupke:cubeGLTF omg:hasGeometry flupke:cubeGLTF_geometry-1 ;
    rdf:type bot:Element .

flupke:coneOBJ_geometry-1 rdf:type omg:Geometry ;
    fog:asObj_v3.0-obj
    ↪  "https://raw.githubusercontent.com/flol3622/AR-Linked-BIM-viewer/
    ↪  main/public/assets/database_1/coneOBJ.obj"^^xsd:anyURI
    ↪  .

flupke:cubeGLTF_geometry-1 rdf:type omg:Geometry ;
    fog:asGltf_v1.0-gltf
    ↪  "https://raw.githubusercontent.com/flol3622/AR-Linked-BIM-viewer/
    ↪  main/public/assets/database_1/cubeGLTF.gltf"^^xsd:anyURI
    ↪  .
```

Listing 1: Example of an Resource Description Framework (RDF) database in turtle format

resources. A triple consists of a subject, a predicate, and an object. The subject is the resource that is being described, the predicate is the property of the subject, and the object is the value of the property. Both the predicate and the object can, in turn, become the subjects of other triples. Listing 1 shows an example of an RDF database described in the Turtle format.

The basic, yet versatile, structure of a triple is illustrated in Figure 1.3. Both the subject and object are considered as nodes in the data graph, and they are linked by the predicate, which is considered as an edge. Multiple triples can thus create and link multiple nodes or enrich a connection between two by creating new edges between them. Each element contains a single resource that can be one of the three types: a URI, a literal, or a blank node. A Uniform Resource Identifier (URI) points to a resource on the web and,

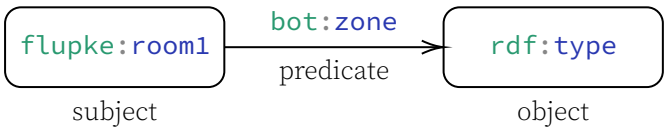

Figure 1.3: Triple structure

as its name states, is unique and unambiguous, thus enabling queries and reasoning of the same nature. A literal is a value, and a blank node is an anonymous resource, sometimes used as a placeholder when the exact resource is not known or not necessary to specify. Due to their nature, a subject must be either a URI or a blank node, a predicate exclusively a URI, and the object may be any of the three types. As URI descriptions can be very long, a prefix can be used to shorten them. This is illustrated in Listing 1 with the `@prefix bot: <https://w3id.org/bot#>`, which declares that `bot:Zone` refers, in its full length, to the address `<https://w3id.org/bot#Zone>`.

This basic concept can be extrapolated to describe and store any kind of data. The advantage for the AEC industry would be to allow any stakeholders to describe and enrich the knowledge base of a building.

1.1.2 Ontologies and reasoning

When looking at Listing 1, we can differentiate between two types of statements: some refer to classes or properties: `flupke:room1`, while others refer to facts associated with them: `rdf:type`. The former is referred to as TBox for "terminology", and the latter is referred to as ABox for "assertions". The TBox is the part of the ontology that describes the classes and properties of the domain, while the ABox is the part of the ontology that describes the facts about the domain.

By developing an ontology, we can describe the domain of interest and the relationships between the classes and properties. This is achieved by defining the classes and properties of the domain and their relationships. The ontology is then used to reason about the domain, allowing us to infer new facts based on the ontology and the facts we know about the domain. This is done by a reasoner, which is a software capable of performing reasoning on the ontology and associated data. The reasoner can be used to infer new facts, check if the facts we know are consistent with the ontology, and check if the ontology itself is consistent.[3] It is often integrated with RDF databases, also known as triplestores or graph databases.

Classes, properties, and their relationships can be defined using Resource Description Framework Schema (RDFS), which is a vocabulary for describing RDF schemas using a basic set of constructs. As an extension of RDFS, Web Ontology Language (OWL) is a vocabulary for describing ontologies using a more expressive set of constructs tailored

[3] W3C, 2015a.

to the needs of ontologies. Both RDFS and OWL are considered to be formal ontologies themselves, as they describe the classes and properties of the domain of RDF.

1.1.3 Triplestores and SPARQL

As briefly discussed in 1.1.2, triplestores are RDF databases that store data in the form of a graph. They are used to store and query Linked Data and are often integrated with a reasoner. The data itself is retrieved and modified using the SPARQL Protocol and RDF Query Language (SPARQL).[4] In contrast to Structured Query Language (SQL), SPARQL queries are able to work across multiple triplestores, called SPARQL endpoints. These are known as federated queries, and their results are combined into a single result set. This is useful when the data is distributed across multiple triplestores in a decentralized manner.[5] For example, multiple stakeholders participating in a project, each with their own database.

1.1.4 Complexity of the graph

The complexity of the graph is a major concern when working with LDBIM. This section discusses the origins of the different sources of geometric data that enrich it.

1.1.4.1 BIM geometry

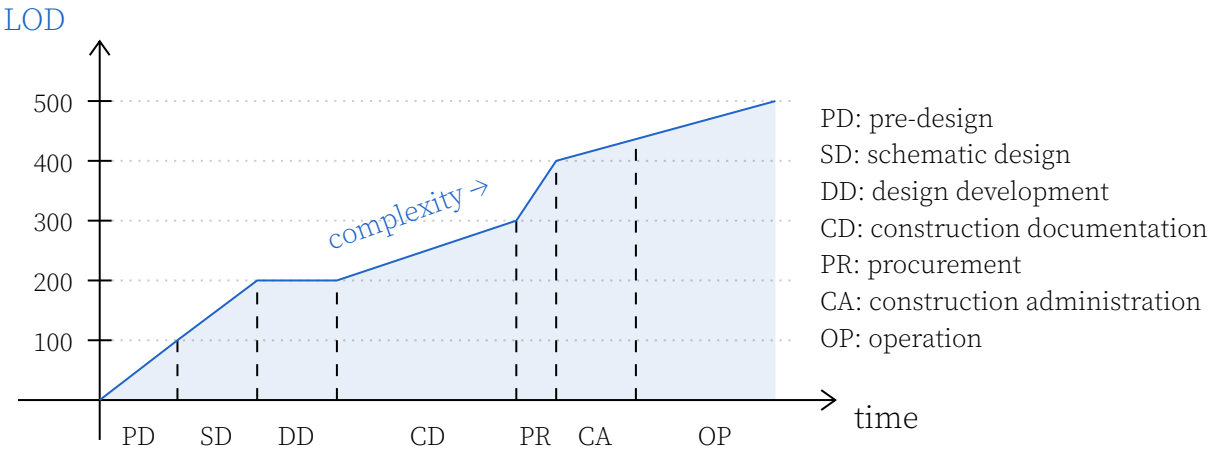

Figure 1.4: Evolution of LOD during the life-cycle of a building.
Based upon the Macleamy Curve (Ilozor & Kelly, 2012)

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. For instance, a door can simply be defined as a box, or up to the level of the screw-thread for the hinge system. The

[4] W3C, 2015c.
[5] Ontotext, 2022.

level of abstraction is here described as the Level of Detail (LOD), which is most of the time pre-selected for the needs of a BIM model, and is applied throughout a single model.

As shown in Figure 1.4, a standard BIM workflow goes through multiple phases, each with their associated model and LOD. This makes it an important concept in the AEC industry, as it allows for a very efficient workflow. The modeling step is approached from a top-down perspective, starting with rougher geometries describing the broader ideas of a concept model and evolving to a more refined model for the construction documentation phase. As the last and longest-standing model, a higher LOD can be used to describe subtle changes in the evolution of a building during the operation phase.

1.1.4.2 LDBIM geometry

The interconnectivity of semantics can also be applied to geometry descriptions. This could allow the co-existence of multiple LODs in a single model database. Besides storing the evolution of a single element's geometry, it enables the linking of the different LODs described in 1.1.4.1 to each other. Not only this, but extending onto the size of the models described in Table 1.1, already existing Mechanical, Electrical and Plumbing (MEP) as structural geometry can be added alongside many other stakeholders.

1.2 Research questions

Implementing culling algorithms 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. It aims to propose a set of possible solutions tailored to this specific problem, while highlighting possibilities for future research and specific use cases.

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

This thesis focuses on computing with data snippets or triples inside a LDBIM model, not within. 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 techniques 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 consist 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 resources in the graph. This enables the viewer to retrieve those resources for a multitude of use cases, transforming it into a user-friendly visual query tool.

1.2.2 Can existing semantics and ontologies be used to feed possible culling algorithms?

In contrast to the computer graphics industry, this interconnected context already contains both explicit and implicit, through inferencing, relationships inside the graph. Similarly to Johansson et al., 2009 and their paper where they used the semantics of a BIM model in Industry Foundation Classes (IFC) format to develop culling techniques. However, this thesis will focus on the use of Semantic Web resources. Therefore, it will analyze both AEC-specific and AEC-related ontologies, such as Geographic Information System (GIS)-related, to see if they can be used to feed culling algorithms.

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)
-> in contrary to Johansson et al., 2015

2.1.1.2 Advantages and challenges

2.1.1.3 Potential applicability to LDBIM

-> as stated before advantage of LOD existing

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, 2023.

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

2.1.3 AEC related ontologies

-> as proposed in Johansson et al., 2015
–> using BIM semantics to introduce new culling Techniques (5)
–> CHC++
-> Present the following as related work but still in early stage, needs way more research (for computer scientists)

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

-> Johansson et al., 2015
–> table 3 about Acceleration techniques

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:

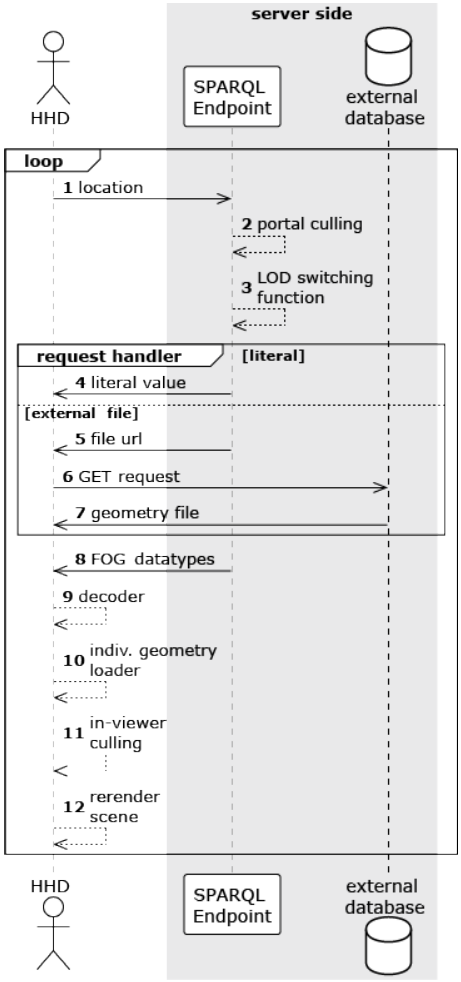

Figure 4.1: Sequence diagram

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 6
BIM Building Information Modelling 6
BOT Building Topology Ontology
FOG File Ontology for Geometry formats
GIS Geographic Information System 13
HHD Hand Held Device 8
IFC Industry Foundation Classes 13
LDBIM Linked Data BIM 6
LOD Level of Detail 12
MEP Mechanical, Electrical and Plumbing 12
OMG Object Management Group
OWL Web Ontology Language 10
RDF Resource Description Framework 9
RDFS Resource Description Framework Schema 10
SDK Software Development Kit
SPARQL SPARQL Protocol and RDF Query Language 11
SQL Structured Query Language 11
URI Uniform Resource Identifier 9
W3C World Wide Web Consortium 8

References

Cohen-Or, D., Chrysanthou, Y. L., Silva, C. T., & Durand, F. (2003). A survey of visibility for walkthrough applications. *IEEE Transactions on Visualization and Computer Graphics, 9*, 412–431. https://doi.org/10.1109/TVCG.2003.1207447

Ilozor, B. D., & Kelly, D. J. (2012). Building information modeling and integrated project delivery in the commercial construction industry: A conceptual study. *Journal of Engineering, Project, and Production Management, 2*, 23–36. https://doi.org/10.32738/JEPPM.201201.0004

Johansson, M., Roupé, M., & Bosch-Sijtsema, P. (2015). Real-time visualization of building information models (bim). *Automation in Construction, 54*, 69–82. https://doi.org/10.1016/J.AUTCON.2015.03.018

Johansson, M., Roupé, M., Johansson, M., & Roupé, M. (2009). *Efficient real-time rendering of building information models. vr for decision making view project virtual produktion planning with the help of bim and visualization, phase iii view project efficient real-time rendering of building information models*. https://www.researchgate.net/publication/220758081

Werbrouck, J. (2018). Linking data : Semantic enrichment of the existing building geometry. https://lib.ugent.be/catalog/rug01:002494740

Referenced webistes

Ontotext. (2022). *What is sparql?* https://www.ontotext.com/knowledgehub/fundamentals/what-is-sparql/

Unity. (2023). *Manual: Memory in unity webgl*. https://docs.unity3d.com/2023.2/Documentation/Manual/webgl-memory.html

W3C. (2015a). *Semantic web: Inference*. https://www.w3.org/standards/semanticweb/inference

W3C. (2015b). *Semantic web: Linked data*. https://www.w3.org/standards/semanticweb/data

W3C. (2015c). *Semantic web: Query*. https://www.w3.org/standards/semanticweb/query

W3C. (2023). *Linked building data community group*. https://www.w3.org/community/lbd/