Connected Object Data in Heritage Conservation

The Value in Natural Stone Restoration Projects

M.A. Janssen Steenberg

2023

TU/e

Master Thesis

Construction Management & Engineering

[This page is intentionally left blank]

Connected Object Data in Heritage Conservation:
The Value in Natural Stone Restoration Projects

Student: M.A. (Marijn) Janssen Steenberg
1580159
m.a.janssen.steenberg@student.tue.nl
marijn@janssensteenberg.com

Program: Master Construction Management and Engineering
Eindhoven University of Technology (TU/e)
Groene Loper 6, 5612AZ Eindhoven, The Netherlands

Course: Graduation project Construction Management and Engineering
7CC40 | ECTS 40

Committee | Chairman: dr. ir. P. (Pieter) Pauwels
Associate Professor
Information Systems in the Built Environment
Department of the Built Environment | TU/e

Committee | 1st Supervisor: ir. A.J.A. (Alex) Donkers
Doctoral Candidate
Information Systems in the Built Environment
Department of the Built Environment | TU/e

Committee | 2nd Supervisor: prof. dr. M.R.V. (Michel) Chaudron
Full Professor
Software Engineering and Technology
Department of Mathematics and Computer Science | TU/e

Company Advisor: ing. T. (Tommy) van Beem
Manager Digitalisation
Nico de Bont | TBI

Company: Aannemingsbedrijf Nico de Bont B.V.
part of: TBI Holdings B.V.
Molenstraat 2, 5262 ED Vught, The Netherlands

Academic Year: 2022-2023

Date Final Colloquium: 30-08-2023

This master thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Integrity.

[This page is intentionally left blank]

PREFACE

Dear reader,

After four years of studying the bachelor Architecture and Construction Engineering at the Avans University of Applied Sciences in 's-Hertogenbosch and three years studying the (pre-)master Construction Management and Engineering at the Eindhoven University of Technology, this thesis marks the end of my educational career. During the various internships I have had over the years, I have been inspired by the value that digitalisation can bring to practice in the construction industry. Since my first introduction with BIM, I have been interested to learn everything about the ever-evolving world of digitalisation in construction. In my view digitalisation is essential to address the numerous challenges faced in the built environment, such as the impacts of climate change, housing and labour market shortages, and the creation of pleasant neighbourhoods.

This thesis is the convergence of my seven years of studying. I am very glad that I could perform research in topics I am interested in, while at the same time broadening my knowledge of new subjects such as the world of heritage conservation, linked data technologies and the creation of a web application with the limited programming skills I had.

I would like to thank everybody that assisted me in writing this thesis for their guidance. First, I would like to thank my supervisors from the Eindhoven University of Technology, Pieter Pauwels, Alex Donkers, and Michel Chaudron for their academic guidance over the past months. I specifically want to thank Pieter for inspiring me to explore the world of Linked Data, Michel for his valuable feedback with his different perspective and Alex for all his ideas and advise in our feedback sessions. Furthermore, I would like to thank my supervisor Tommy van Beem and all my other colleagues from Nico de Bont | TBI for their support and input. Finally, and most importantly, I would like to thank my friends and family for all the support they gave me during my entire studies.

I hope you will enjoy reading this work!

Marijn Janssen Steenberg

Eindhoven, August 2023

TABLE OF CONTENTS

PREFACE 5
SUMMARY 8
SAMENVATTING 10
ABSTRACT 12

OVERVIEW – ABBREVIATIONS 13
OVERVIEW – FIGURES 15
OVERVIEW – LISTINGS 16
OVERVIEW – TABLES 16

1 INTRODUCTION 17
1.1 Background 17
1.2 Problem Statement 18
1.3 Reading Guide 19

2 OBJECT INFORMATION IN HERITAGE CONSERVATION 20
2.1 Conservation of Built Heritage in a Dutch Context 20
2.2 Data in the Conservation Cycle: the Case of Natural Stone 23
2.3 Conclusion 27

3 CONNECTING DATA IN THE CONSTRUCTION INDUSTRY 28
3.1 Building Information Moddeling 28
3.2 Linked Data 31
3.3 Connected Data 38
3.4 Existing Ontologies for Natural Stone Restorations 39
3.5 Digitisation in Restoration Projects 45
3.6 Conclusion 46

4 METHODOLOGY 47
4.1 Research Approach 47
4.2 Requirements Definition 48
4.3 System Architecture Design 52

5 PROTOTYPE DEVELOPMENT 53
5.1 Case Project 53
5.2 System Architecture Implementation 54
5.3 Data Preparation 54
5.4 Graph Creation 59
5.5 Data Visualisation 64

6 PROTOTYPE VALIDATION 72
6.1 Competency Questions 72
6.2 Example Process 75
6.3 Discussion 78

7 CONCLUSION 79
7.1 Summary of Research 79
7.2 Contribution 81
7.3 Recommendations 82

REFERENCES 84

APPENDICES 90
Appendix I: Prefixes and Namespaces 90
Appendix II: Heritage LBDviz 91
Appendix III: SPARQL Query - Spreadsheet Export 92

SUMMARY

Historical buildings in the Netherlands hold significant cultural and economic value. These buildings are key elements of the country's cultural heritage, as they establish a physical link to the past. The Dutch government has implemented various policies and regulations to safeguard and preserve the nation's historical buildings. During restoration projects of historical buildings, preserving information is as important as the conservation of the monument itself. This ensures that valuable information is available for any future restoration projects. In the Netherlands, building conservation is governed through self-regulation and guidelines established by the Dutch Cultural Heritage Agency and the ERM Foundation. ERM guidelines provide technical specifications, quality standards and information requirements for restoration work, ensuring the preservation of historical value throughout a continuous process. This "Conservation Cycle" involves inspections, damage diagnoses, intervention strategy using the "Conservation Ladder", on-site intervention works, and the monitoring of the building during operation. The choices made and information gathered during the entire conservation cycle should be documented to provide valuable information for future projects on the same or other buildings. Currently, a wide range of standards and data formats are used to document the relevant aspects in natural stone restoration projects, such as cultural-historical value, damage assessment, damage identification, material type, and shape. These different standards and file formats, create a situation that is difficult to manage.

Building Information Modelling (BIM) has emerged as a significant collaborative process in the Architecture, Engineering, and Construction (AEC) industry. BIM enables the creation of digital representations of building structures that are data-rich and object-oriented. Linked data, which is based on semantic web technologies, facilitates the creation of knowledge graphs, establishes relationships between data objects, provides context and enhances the understanding of information. By combining BIM principles, linked data and the specific requirements of heritage projects, data from various sources can be connected and structured according to the relevant standards in restoration projects. This approach can be considered the next step towards advancing BIM maturity in the sector. This should enable professionals to make informed decisions, ensure the preservation of historical and cultural significance, and efficiently manage the complex challenges of restoring historical structures. This thesis aims to facilitate and improve the structuring and visualisation of data during the restoration of natural stone objects on historical buildings by using connected data, the combination of linked data technologies and files. This is accomplished by the development of a prototype tool.

The development of this tool is started with an analysis of the existing process on-site, to determine the necessary steps and data generated throughout the process. The process of conserving natural stone facades consists of a series of tasks depending on the type of intervention to be executed. Discussing the resulting process schema with stakeholders assisted in defining what information needs exist during the different stages of natural stone restoration. These needs are used to establish eight competency questions. These questions are translated into three sets of different use cases that can be used to develop the system. The developed use cases should consider multiple types of users. The use cases for the beginner user focus on providing recognisable output, while also introducing new possibilities to the user. More advanced users should be provided with functionalities for finding traditional information through the system. Lastly, professionals are granted the ability to create queries based on their extensive knowledge of the data structure.

The development continued with the establishment of a conceptual system architecture that can deliver the defined use cases. The practical implementation of this system architecture is developed using the case project of the Monument on Dam Square. As sources from the case project were generated using a traditional process, some preparation is necessary before their use as input data. This includes preparing multiple spreadsheets and creating an IFC model where every object in the Monument is represented by an individual IfcElement. Furthermore, a data structure is formed by utilising various existing ontologies that were identified in the literature study. This is supplemented by a newly created RII ontology that follows a modular and flexible structure. This data structure is subsequently used to construct an RDF graph of the case project data. In addition, to make sure the several non-RDF files can be connected to the graph, they are made available through a local live server, giving a URI to every file that is added to the graph. To demonstrate the possibilities of combining RDF and non-RDF data in the context of restoration projects, a front-end web application is developed that addressed the defined use cases. The resulting Heritage LBDviz expands the existing LBDviz and includes various additional features to address the use cases that exist in natural stone restoration projects. In addition to the Heritage LBDviz, the graph database can also automatically generate an enhanced version of the traditional spreadsheet used in the current process, demonstrating the new possibilities that arise from utilising connected data.

To validate the proposed system, a specific object is selected in the case project. This object is used to reflect, together with industry experts, on the value of the created system in answering the competency questions of stakeholders. During the validation, it is concluded that the developed prototype already provides significant savings in time and minimises the likelihood of errors. Furthermore, it enhances awareness of data value and its vendor-neutral approach that makes a project less reliant on specific software.

SAMENVATTING

Historische gebouwen in Nederland zijn van grote culturele en economische waarde. Deze gebouwen zijn belangrijke elementen van het culturele erfgoed van het land, omdat ze een fysieke link naar het verleden vormen. De Nederlandse overheid heeft verschillende beleidslijnen en wetten geïmplementeerd om de historische gebouwen te beschermen en te behouden. Tijdens restauratieprojecten van historische gebouwen is het bewaren van informatie net zo belangrijk als het behoud van het monument zelf. Dit waarborgt dat waardevolle informatie beschikbaar is voor eventuele toekomstige restauratieprojecten. In Nederland wordt het behoud van gebouwen geregeld door zelfregulering en richtlijnen die zijn opgesteld door de Rijksdienst voor het Cultureel Erfgoed en de Stichting ERM. ERM-richtlijnen voorzien in technische specificaties, kwaliteitsnormen en informatievereisten voor restauratiewerkzaamheden, waardoor het behoud van historische waarde in een continu proces wordt gewaarborgd. Deze conserveringscyclus omvat inspecties, schadediagnoses, interventiestrategieën met behulp van de restauratieladder, interventiewerkzaamheden op locatie en het monitoren van het gebouw tijdens de exploitatie. De keuzes die worden gemaakt en de informatie die wordt verzameld tijdens de hele conserveringscyclus moeten worden gedocumenteerd om waardevolle informatie te conserveren voor toekomstige projecten voor hetzelfde of andere gebouwen. Momenteel wordt er een breed scala aan standaarden en gegevensformaten gebruikt om de relevante aspecten in natuursteenrestauratieprojecten te documenteren, zoals cultuurhistorische waarde, schadebeoordeling, schade-identificatie, materiaaltype en vorm. Dit creëert een situatie die moeilijk te beheren is.

Building Information Modelling (BIM) is uitgegroeid tot een belangrijk samenwerkingsproces in de architectuur-, ingenieurs- en bouwsector. BIM maakt het mogelijk om digitale representaties van bouwwerken te maken die rijk zijn aan gegevens en object georiënteerd zijn. Linked Data, gebaseerd op semantische web technologieën, faciliteert het maken van kennisgrafen met het leggen van relaties tussen data. Hierdoor wordt data voorzien van context en verbetert het begrip van informatie. Door BIM-principes, Linked Data en de specifieke vereisten van erfgoedprojecten te combineren, kunnen gegevens uit verschillende bronnen in restauratieprojecten met elkaar verbonden en gestructureerd worden volgens de relevante standaarden. Deze aanpak kan worden beschouwd als de volgende stap in de ontwikkeling van BIM in de sector. Dit zou professionals in staat moeten stellen om weloverwogen beslissingen te nemen, het behoud van historische en culturele betekenis te garanderen en de complexe uitdagingen van het restaureren van monumenten efficiënt aan te gaan. Deze thesis heeft als doel het structureren en visualiseren van gegevens tijdens de restauratie van natuurstenen objecten op historische gebouwen te vergemakkelijken en te verbeteren door gebruik te maken van verbonden data, een combinatie van Linked Data met bestanden. Dit wordt bereikt door de ontwikkeling van een prototype tool.

De ontwikkeling van de tool is gestart met het analyseren van het bestaande proces op de bouwplaats, om de stappen en gegevens te bepalen die tijdens het proces worden gegenereerd. Het conserveringsproces van natuurstenen gevels bestaat uit een reeks taken, afhankelijk van het type interventie dat moet worden uitgevoerd. Het resulterende processchema is met belanghebbenden doorgesproken voor het definiëren van de informatiebehoeften tijdens de verschillende fasen van een natuursteenrestauratie. Deze behoeften zijn gebruikt om acht competentievragen op te stellen. Deze vragen zijn vertaald in drie sets van verschillende usecases die gebruikt kunnen worden om het systeem te ontwikkelen. De usecases voor de beginnende gebruiker met herkenbare output, een voor ervaren gebruikers met een nieuwe tool en een voor de gevorderde gebruiker met veel data kennis.

De ontwikkeling ging verder met het opstellen van een conceptuele systeemarchitectuur die de gedefinieerde usecases kan realiseren. De praktische implementatie van deze systeemarchitectuur is ontwikkeld aan de hand van het casusproject van het Monument op de Dam. Aangezien de brongegevens van het casusproject via een traditioneel proces zijn gegenereerd, is enige voorbereiding nodig voordat ze als invoergegevens kunnen worden gebruikt. Dit omvat het voorbereiden van meerdere spreadsheets en het creëren van een IFC-model waarin elk object in het Monument wordt gerepresenteerd door een individueel IfcElement. Verder is er een datastructuur gevormd door gebruik te maken van verschillende bestaande ontologieën die in de literatuurstudie zijn geïdentificeerd. Deze is aangevuld met een nieuw gecreëerde RII-ontologie die een modulaire en flexibele structuur volgt. Deze datastructuur is vervolgens gebruikt om een RDF-grafiek te bouwen met de gegevens van het casusproject. Om ervoor te zorgen dat de verschillende niet-RDF bestanden aan de grafiek gekoppeld kunnen worden, worden ze bovendien beschikbaar gemaakt via een lokale live server, waarbij een URI wordt gegeven aan elk bestand dat aan de grafiek wordt toegevoegd. Om de mogelijkheden van het combineren van RDF en niet-RDF data in de context van restauratieprojecten te demonstreren, is een front-end webapplicatie ontwikkeld die de gedefinieerde usecases invult. De resulterende Heritage LBDviz breidt de bestaande LBDviz uit en bevat verschillende extra functies om de usecases die bestaan in natuursteen restauratieprojecten te adresseren. Naast de Heritage LBDviz kan de graafdatabase ook automatisch een verbeterde versie genereren van de traditionele spreadsheet die in het huidige proces wordt gebruikt, waarmee de nieuwe mogelijkheden worden gedemonstreerd die ontstaan door het gebruik van verbonden data.

Om het voorgestelde systeem te valideren, is in het casusproject een specifiek object geselecteerd. Dit object is gebruikt om samen met experts uit de industrie te reflecteren op de waarde van het ontwikkelde systeem voor het beantwoorden van competentievragen. Tijdens de validatie is geconstateerd dat het ontwikkelde prototype al een aanzienlijke tijdsbesparing oplevert en de kans op fouten vermindert. Bovendien vergroot het de bewustwording over de waarde van gegevens en door de leveranciersonafhankelijke aanpak is een project minder afhankelijk van specifieke software.

ABSTRACT

The Netherlands' historical buildings are of immense cultural and economic importance, connecting the country to its past. To preserve these structures, the Dutch government has implemented policies, and restoration projects focus on conserving both the buildings and valuable information for future efforts. Building conservation in the Netherlands follows the "Conservation Cycle," which involves inspections, damage diagnoses, intervention strategies, on-site works, and ongoing monitoring. Documenting relevant aspects of restoration projects creates challenges due to the use of various data formats and standards. Building Information Modelling (BIM) has emerged as a valuable collaborative tool in the Architecture, Engineering, and Construction industry. BIM, along with linked data based on semantic web technologies, enables the creation of knowledge graphs, facilitating linking between data objects. This thesis proposes connected data, the combination of linked data technologies and files, as a hybrid approach for structuring and visualising data.

This thesis develops a prototype tool that combines BIM and Linked Data technologies to improve data structuring and visualisation during the restoration of historical buildings with natural stone elements. The tool is designed to serve different users, including beginners, advanced users, and professionals, providing them with relevant functionalities and information. The research develops a conceptual system architecture and implements it using a case project, the Monument on Dam Square. By preparing data from the case project and creating a data structure using various existing ontologies, an RDF graph is constructed. A front-end web application called Heritage LBDviz is developed to demonstrate the possibilities of combining RDF and non-RDF data for restoration projects. The proposed system is validated using a specific object from the case project, and industry experts confirm its value in saving time, reducing errors, and enhancing data awareness. The system's vendor-neutral approach makes projects less reliant on specific software, and as more data is connected, it opens possibilities for more innovative applications in the restoration domain.

Keywords
Heritage Conservation, Natural Stone Restoration, Building Information Modelling, Linked Data, Heritage LBDviz

OVERVIEW – ABBREVIATIONS

AAT Arts & Architecture Thesaurus, *a thesaurus of concepts, published in SKOS by the Getty institute*. p. 24, 44

AEC Architecture Engineering and Construction, *an acronym to describe the construction industry including design and engineering*. p. 17, 28, 39, 45, 79, 81

API Application Programming Interface, *a way for computer programs to communicate with each other*. p. 55

BAG Basisregistratie Adressen en Gebouwen, *Dutch registry of addresses and buildings.* p. 39

BIM Building Information Modelling, a *digitalised collaboration process.* p. 17, 28, 38, 45, 79

BOT Building Topology Ontology, *an LBD ontology for describing the core topological concepts of a building.* p. 39, 41, 57, 59

BPMN Business Process Model and Notation, *a standard for specifying business processes.* p. 48

CDE Common Data Environment, *a central BIM collaboration environment*. p. 17, 28

CSS Cascading Style Sheets, *a language used for describing the presentation of a HTML document.* p. 65

CSV Comma Separated Values, *a file format*. p. 31, 54, 59

CTO Construction Tasks Ontology, *an LBD ontology for describing tasks in construction.* p. 42

CWA Closed World Assumption, *inverse of OWA*. p. 33

DOT Damage Topology Ontology, *an LBD ontology for describing damages on building elements.* p. 41, 42, 57

ERM Erkende Restauratiekwaliteit Monumentenzorg, *Dutch foundation that promotes* quality *in restoration by publishing guidelines.* p. 20, 21, 23, 25, 27, 48, 79

ERM-BRL ERM-Beoordelingsrichtlijn, *ERM process guideline*. p. 20

ERM-URL ERM-Uitvoeringsrichtlijn, *ERM technical guideline*. p. 20, 21, 23, 25, 26, 44

ETL Extract-Transform-Load, *process of combining data from multiple sources*. p. 52, 54, 59

GUID Globally Unique Identifier, *identifier that is presumed to be globally unique*. p. 51, 65

HBIM Heritage/Historical Building Information Modelling, *BIM for heritage projects*. p. 38

HTML HyperText Markup Language, *markup language for web pages*. p. 65

ICDD Information Container for Linked Document Delivery, *enables description of a set of interlinked construction documents in a structured way using SWT*. p. 18, 35, 77

IFC Industry Foundation Classes, *open standardised data schema for exchanging and sharing BIM data.* p. 18, 28, 30, 31, 35, 38, 39, 54, 59, 65, 72, 79, 82

ISO International Standards Organisation, *developer and publisher of international standards*. p. 20, 28, 30, 35

JSON JavaScript Object Notation, *a data format*. p. 32, 43, 59, 71

LBD Linked Building Data, *a network of modular ontologies developed by LBD-CG*. p. 39, 59

LBD-CG Linked Building Data Community Group, *W3C community group of experts in area of BIM and SWT*. p. 39, 41, 78, 79, 81

MDCS Monument Diagnosis and Conservation System, *an online tool that facilitates the diagnosis of damage and supports the conservation of historical buildings*. p. 24, 25, 41

NEN Nederlandse Norm, *developer and publisher of Dutch standards*. p. 20, 21, 24, 35, 63

OMG Ontology for Managing Geometry, *an LBD ontology for describing the geometry of a building*. p. 43, 57

OWA Open World Assumption, *the assumption that if something is not known to be true, it is not necessarily false*. p. 33, 35

OWL Web Ontology Language, *W3C core ontology for defining ontologies*. p. 33, 36, 57, 63, 64

RCE Rijksdienst voor het Cultureel Erfgoed, *the Dutch governmental cultural heritage agency*. p. 20, 24, 26, 39, 42, 44

RDF Resource Description Framework, *W3C standard for modelling data*. p. 32, 33, 34, 35, 35, 43, 54, 59, 63, 64, 65, 68, 79

RDFS Resource Description Framework Schema, *W3C core ontology for defining ontologies*. p. 31, 33, 36, 44

SHACL Shapes Constrained Languages, *W3C standard for validating RDF data*. p. 18, 35, 72, 78, 82

SKOS Simple Knowledge Organization System, *W3C ontology for defining thesauri*. p. 35, 36, 39, 41, 42, 44, 57, 81

SPARQL SPARQL Protocol and RDF Query Language, *query language for RDF databases*. p. 31, 34, 35, 36, 59, 63, 64, 65, 71, 72, 75

SWT Semantic Web Technologies, *set of technologies standardised by W3C*. p. 31, 33, 38

URI Uniform Resource Identifier, *unique identifier of data*. p. 31, 32, 41, 59, 63, 67, 70, 79

URL Uniform Resource Locator, *type of URI*. p. 31

W3C World Wide Web Consortium, *international standardisation body for open web standards*. p. 31, 34, 35

XML Extensible Markup Language, *a data format*. p. 32, 59

OVERVIEW – FIGURES

Figure 1: Reading Guide....19
Figure 2: Conservation Cycle & Ladder (Based on Lubelli et al., 2021) 22
Figure 3: Information Aspects and Standards in Conservation Cycle 23
Figure 4: BIM Maturity (based on Bew & Richards, 2008) 28
Figure 5: Information Management Workflow, based on Klemt-Albert et al. (2018) and ISO 19650-1 (2018) 29
Figure 6: Part of the IFC Data Model (Borrmann, Beetz, et al., 2018) 30
Figure 7: Examples of Linked Data Technologies: RDF Triple 32
Figure 8: Examples of Linked Data Technologies: RDF Literals and Blank Nodes 32
Figure 9: Examples of Linked Data Technologies: RDF Ontologies....33
Figure 10: Examples of Linked Data Technologies: SPARQL 34
Figure 11: Examples of Linked Data Technologies: SHACL....35
Figure 12: Property Levels 37
Figure 13: Towards a Web of Data 38
Figure 14: Connected Data as a step towards a Web of Data 38
Figure 14: Dutch Cultural Heritage Agency Linked Data Structures: CHT, CHO, CEO....39
Figure 15: Building Topology Ontology (BOT) 40
Figure 16: Stone Component Ontology (SCO)....40
Figure 17: Damage Topology Ontology (DOT) 1/3 41
Figure 18: Damage Topology Ontology (DOT) 2/3....41
Figure 19: Damage Topology Ontology (DOT) 3/3....42
Figure 20: Construction Tasks Ontology (CTO) 42
Figure 21: Ontology for Managing Geometry (OMG) with extension FOG....43
Figure 22: Research Methodology 47
Figure 23: Process Schema of the Natural Stone Restoration Process 49
Figure 24: Use Case Diagram of proposed System 51
Figure 25: Proposed conceptual System Architecture 52
Figure 26: Restoration of the National Monument 53
Figure 27: National Monument on Dam Square....53
Figure 28: System Architecture of Prototype 54
Figure 29: Difference between IFC Models before and after editing in SketchUp 55
Figure 30: Overview of used Ontologies 57
Figure 31: Restoration Intervention Information (RII) ontology - Interventions....57
Figure 32: Restoration Intervention Information (RII) ontology - Images....58
Figure 33: Restoration Intervention Information (RII) ontology - Properties 58
Figure 34: Example of Data in GraphDB....63
Figure 35: LBDviz by Donkers et al. (2023)....65
Figure 36: Heritage LBDviz - Finding Object Label 66
Figure 37: Heritage LBDviz - Built-in IFC.js Viewer Functionalities....67
Figure 38: Heritage LBDviz - Showing Images in Viewer....67
Figure 39: Heritage LBDviz - Highlighting Objects in IFC Viewer with Query Button 68
Figure 40: Sequence Diagram - Highlighting Objects in IFC Viewer with Query Button 69
Figure 41: Heritage LBDviz - Highlighting Objects in IFC Viewer from Results....70
Figure 42: Heritage LBDviz - Creating Links to Objects in IFC Viewer 71
Figure 43: Spreadsheet Export from Graph Database.... 71

OVERVIEW – LISTINGS

Listing 1: Fragment of the CSV File of the Photos 56
Listing 2: Fragment of the Turtle File from the IFCtoLBD Converter 59
Listing 3: Pseudocode Fragment of Python Script - Creating RDF Graph 60
Listing 4: Pseudocode Fragment of Python Script - Insert Data into Graph 60
Listing 5: Pseudocode Fragment of Python Script - Insert Query for every Row 61
Listing 6: Pseudocode Fragment of Python Script - SPARQL INSERT WHERE Query 62
Listing 7: Pseudocode Fragment of Python Script - SPARQL Select Query as additional Input Data 62
Listing 8: Example SPARQL 64
Listing 9: Pseudocode Fragment of JavaScript Code - Finding Object Label 66
Listing 10: Pseudocode Fragment of JavaScript Code - Showing Images in Viewer 68
Listing 11: Pseudocode Fragment of JavaScript Code - Highlighting Objects in IFC Viewer with Query Button 69
Listing 12: Pseudocode Fragment of Python Script – Export Data from Graph 71
Listing 13: SPARQL Query CQ1 72
Listing 14: SPARQL Query CQ2 72
Listing 15: SPARQL Query CQ3 1/2 73
Listing 16: SPARQL Query CQ3 2/2 73
Listing 17: SPARQL Query CQ7 74

OVERVIEW – TABLES

Table 1: Current Competency Questions 50
Table 2: Layers of the proposed System Architecture 52

1 INTRODUCTION

The introduction presents the research background. Additionally, the problem to be addressed in this thesis is introduced and the questions to be answered are defined. Finally, this section contains a reading guide for the rest of this document.

1.1 BACKGROUND

Historical buildings in the Netherlands hold significant cultural and economic value. These buildings are key elements of the country's cultural heritage, as they establish a physical link to the past. The Dutch government has implemented various policies and regulations to safeguard and preserve the nation's historical buildings (Erfgoedwet, 2021). During restoration projects of historical buildings, preserving information is as important as the conservation of the monument itself. This ensures that valuable information is available for any future restoration projects. This information is documented using various standards, data formats, and systems, resulting in a situation of fragmented information that is difficult to manage.

Building Information Modelling (BIM) is being increasingly adopted by the AEC industry, digitalising collaborative workflows in design, construction and operation. BIM offers numerous opportunities and challenges in terms of enhancing information and project management, ensuring the production and use of high-quality information. The Industry Foundation Classes (IFC) file format is widely recognised for storing information about objects in BIM models. Current Common Data Environment (CDE) solutions such as Autodesk Construction Cloud[1] and Trimble Connect[2] provide ways to link information to BIM-models. However, the capabilities of a platform are often limited, and the vendor lock-in of these systems makes it difficult to link related information across different systems.

For instance, during the five-year restoration of the Dom Tower in Utrecht, where ten thousand natural stone elements underwent repairs or replacements, each of these objects was assigned a digital code by the contractor Nico de Bont (Beerda, 2022). This process is executed by placing pins on PDF building elevations, where each pin is given a unique identifier using Autodesk software. The system is employed to monitor the status of each element, save metadata about it, and save pictures throughout the project to guarantee a verifiable process. QR codes are utilised on elements that are temporarily stored or delivered from the masonry. This links the physical object to the system, which displays its current status and position within the building. This method is more efficient than the traditional approach, which involved numbered tags on each element that corresponded to numbers on a construction drawing, sometimes supplemented by spreadsheets. However, system capabilities are limited, and complete exports are confined to PDF format, which causes a loss of semantics. Linked data offers promising technical possibilities that could enhance this situation by creating structured data with semantics.

Digitalisation through the use of linked data technologies could help with this effort and with new challenges such as better documenting the as-built files, which is a type of Digital Building Logbook (European Commission et al., 2021; Gómez-Gil et al., 2022). Linked data technologies allow for the connection and structuring of currently fragmented datasets from various domains and stakeholders, thereby making information easier to find. As more data accumulates, this structured data, including semantics, can also be used over time to effectively train and improve predictive systems.

[1] *https://construction.autodesk.com/*

[2] *https://connect.trimble.com/*

1.2 Problem Statement

Connecting heritage building data with BIM models by using linked data technologies could help in structuring, validating, visualising, and sharing connected data. Other research has already made some efforts in structuring data in a built heritage context. Hamdan et al. (2019) developed a modular web ontology for defining damage objects and their topology. Bonduel (2021) contributed to the developed of a network of different new and existing modular ontologies to describe built heritage cases. However, this research also concludes that this network, that describes, among others, objects, properties, damages, and tasks is not extensive enough for use in real projects. After this structure is defined, one needs to make sure the data is also entered correctly. Therefore, before the connected data should be used in practice, one should make sure the data meets the information requirements that are set. Various research proposes a workflow where linked data technology SHACL is used to automate the rule-compliance checking process in a construction project (Hagedorn & König, 2021; van den Bersselaar, 2022). This improves the data quality of both the BIM models and other heritage building data.

When validated, the connected data should be available to answer the questions of an end user. Bonduel (2021) suggested further work where a demo application should be developed to illustrate the implementation of the created data structure. Donkers et al. (2023) created a single page web application that has a full page IFC viewer with several menu windows on top of the viewer that can be used to ask questions to the data structure based on the selected object in the viewer. This web application does not focus on the specific questions of heritage projects. However, other research has already shown that it can be used throughout projects in another context (Donkers et al., 2023; van der Hall, 2023). The current functionalities of this viewer are limited and use in a heritage context requires specific functionalities that are relevant to the processes in a heritage project.

Working with connected data also enables improvements in sharing data. In practice data sharing is done with documents. Linked data technologies can however, aid in the transition from a document-based workflow towards a data-driven one. With an Information Container for linked Document Delivery (ICDD), it is possible to describe a set of interlinked construction documents in a structured way using semantic web technologies. Hagedorn et al. (2023) recognise the ICDD concept to provide a structured data package, which can be shared at the end of a construction project, without limitations on data type. However, they do see it only as an intermediate step to a fully linked data workflow without using files. Van der Pas (2022) proposed a system, based on linked data technologies, that ensures that data of buildings remains interpretable, accessible, and findable during the operational phase of a building.

1.2.1 Research Gap & Objective

Other research has already made efforts that could help in structuring, validating, visualising, and sharing connected data in the context of restoration projects. However, in practice, there still exists a research gap in structuring and visualising connected data in the construction phase of heritage projects. This thesis sets the objective to facilitate and improve data structuring, validating, visualising, and sharing during the restoration of natural stone objects in historical buildings, by using connected data structures that are vendor independent, extendable, and flexible. The created solution should use these new techniques, while being integrated with existing processes and tools on site. In this thesis, research is done into the capabilities of linked data technologies in combination with BIM models and other files to achieve this. This combination is defined as connected data.

1.2.2 Research Questions

The following main research question has been formulated:

> *What value can be created by using linked data technologies and BIM for structuring and visualising a connected data model during natural stone restoration projects on historical buildings?*

To help answer the main question, four sub-questions have been defined:

1. *Which object-related data is currently used during natural stone restoration projects?*
2. *What are semantic web technologies, linked data, and connected data; and why are they promising for structuring and visualising a data model of a natural stone restoration project?*
3. *What questions arise in the current process of a natural stone restoration project?*
4. *What is a suitable workflow for using a connected data model during a natural stone restoration project?*

1.3 Reading Guide

Figure 1 outlines the structure of this thesis. Chapter 2 explores the state of built heritage conservation in The Netherlands and identifies the information utilised in the restoration of natural stone objects through literature research. Chapter 3 explores how linked data technologies can assist in managing difficulties arising from a wide range of data formats and standards encountered during information documentation in natural stone restoration projects, with Building Information Modelling as a supporting technology. Chapter 4 describes the methods used to design a prototype for structuring and visualising connected data. Furthermore, this chapter outlines the requirements for the functionalities of the system and the resulting conceptual system architecture, established through a dialog with industry experts. Next, a prototype implementation of this system has been developed using a case project. Chapter 5 demonstrates the different stages of the prototype development process and the various functions created within the visualisation tool, Heritage LBDviz. However, the chapter begins with a description of the case project that was used. In Chapter 6, the validation of the prototype system is presented in relation to the requirements developed with industry experts. Initially, the validation process focuses on evaluating the system's ability to answer the various Competency Questions. Then, a specific object is selected to reflect the functionalities of the system throughout the process. Finally, the findings are discussed. The concluding Chapter 7 summarises the research results and explains the contribution of the study in scientific and societal contexts. Finally, recommendations for further research and implementation are provided.

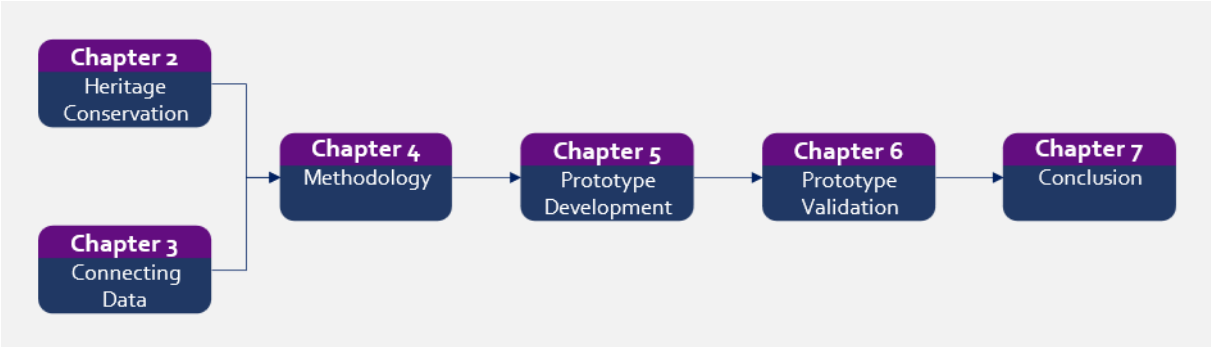


Figure 1: Reading Guide

2 OBJECT INFORMATION IN HERITAGE CONSERVATION

The literature research starts with exploring the state of built heritage conservation in The Netherlands and identifies the information that is used in restoration of natural stone objects.

2.1 CONSERVATION OF BUILT HERITAGE IN A DUTCH CONTEXT

This section highlights the significance of historical buildings in the Netherlands and their protection through legislation. It explains the development of regulation over the years, the current self-regulation system used in the sector, the role of the Monumentenwacht organisations and the Conservation ladder as a decision-making tool for interventions.

2.1.1 Monuments in the Netherlands

Historical buildings in the Netherlands are of great value both culturally and economically. These buildings are an important part of the country's cultural heritage, as they provide a tangible connection to the past. They also contribute to the country's tourism industry, as many visitors are drawn to the Netherlands to see its historical buildings and learn about the country's rich history.

During the first years following the Second World War, the country focused on reconstruction. The housing shortages needed to be solved and there was a great desire to look forward and modernise. This resulted in the demolition of many old buildings, and even entire city centres were destroyed (Plasterk & Cramer, 2009). Therefore, the first Monument Act of 1961 was aimed at protecting as many valuable buildings and structures from before 1850 as possible. The law resulted in a list of 34.000 monuments that were now legally protected objects by the state. In the next decade, several protected village and city areas were also added to the list. In the 1990s, buildings from later periods (1850-1940) were also designated as national monuments. More recently, objects from the reconstruction period (1940-1965) have also been designated. In 2012, the requirement that properties be at least 50 years old to qualify was removed. The current law states that an object can receive the status if it is a monument of national or regional interest because of its beauty, significance for science or cultural-historical value (Erfgoedwet, 2021). Currently, there are over 60.000 building structures that have the status of national monument ('Rijksmonument' in Dutch) and in addition, there are more than 70.000 provincial and municipal monuments (van der Horst et al., 2022). The Dutch Cultural Heritage Agency (RCE) maintains a register of all national monuments (RCE, n.d.-b).

2.1.2 Quality in Restoration through Self-Regulation

In 2009, a significant change in government policy began with the Modernisation of the Monument Care policy, the tasks and responsibilities of the Dutch Cultural Heritage Agency (RCE) were further decentralised (Naldini & Hunen, 2019). With the help of the RCE, the foundation "Erkende Restauratiekwaliteit Monumentenzorg" (ERM) was transformed into a platform for creating and managing guidelines. Within this framework various branches involved in conservation created guidelines under the supervision and guidance of the ERM. This had as a result that each branch, ranging from architects and contractors to masons and carpenters, has developed its own guidelines together with experts and authorities. A commission of experts has the task to approve and monitor the guidelines. This has created a system of self-regulation to meet quality standards. The guidelines are not legal regulations, but practical guidelines created for and by the market. They are in terms of purpose similar to standardisation by organisations like NEN and ISO. The foundation ERM has the task to distribute the execution guidelines to all interested parties and also provides the opportunity to get certified by specialist external auditors (Naldini et al., 2021).

The ERM guidelines are widely used by restoration professionals in the Netherlands. They cover various aspects of restoration work, including the planning and design of restoration projects, the use of appropriate materials and techniques, and the supervision of the work. The aim of the standards is to ensure that restoration work is carried out in a way that preserves the historical value of the monument (Naldini & Hunen, 2019). There are two types of guidelines developed by the ERM: ERM-URL and ERM-BRL (ERM, n.d.). ERM-URL stands for "ERM-Uitvoeringsrichtlijn" and provides technical specifications, legal requirements, and procedural aspects for the restoration of historical monuments. They describe how a specific part of the restoration work on a monument should be carried out using historically appropriate techniques while also integrating modern technology. Guidelines also provide technical specifications for materials, product use, connections, and more. An ERM-BRL, "ERM-Beoordelingsrichtlijn" in Dutch, provides requirements and work agreements for companies to systematically ensure quality in their restoration work. The document outlines the requirements that a company must meet to become certified. Working according to the relevant ERM-URLs is one of these requirements.

2.1.3 Conservation of Monuments in the Netherlands

In the 1970s it became apparent that preventive conservation and maintenance were necessary after the restoration of a monumental building. To address this need, the Monumentenwacht organisations were established at the provincial level. Its responsibilities include conducting regular inspections, evaluating the state of conservation, reporting problems, and conducting minor interventions (Naldini et al., 2018; Torre & Moioli, 2021). The Monumentenwacht organisation works based on a subscription model, when subscribed owners get access to the services of the organisation. The subscription to the Monumentenwacht is voluntary and not obligatory (Naldini et al., 2021). However, membership is considered affordable and a good investment, this also results from the fact that most Monumentenwacht organisations are partly funded by provincial subsidies (Naldini et al., 2018). As a result of the decentralisation of the Dutch Cultural Heritage Agency (RCE), the assistance of Monumentenwacht has become essential for private owners of non-iconic buildings to determine necessary maintenance activities (Naldini & Hunen, 2019). In such cases, Monumentenwacht inspectors mainly oversee the conservation plan, whereas, for iconic buildings, a restoration team is typically formed, consisting of an architect, experts from the Dutch Cultural Heritage Agency (RCE), contractors, and frequently Monumentenwacht inspectors (Naldini et al., 2021). The Dutch Monumentenwacht has served as a model for many other countries, including neighbouring Belgium (Vandesande & van Balen, 2016; Zijlstra et al., 2021).

According to Lubelli et al. (2021), the basic conservation process can be described by the steps of the conservation cycle, see Figure 2. Throughout this process, all decisions by different stakholders, the reasons for these decisions, including the materials and techniques chosen, should be recorded so that they can be accessed in later steps. Monumentenwacht tasks mainly focus on the start of the conservation cycle, signalling and documenting damage based on visual inspections. Inspections can be conducted according to the ERM-URL 2005 guideline for inspections on monumental buildings, which is based on the handbook of Monumentenwacht and the NEN 2767 standard (ERM, 2020b). The identified damages should then be diagnosed to determine the cause of the damage. Often, a visual inspection is not sufficient to make a complete diagnosis. In these situations, additional investigations by experts are necessary in the diagnosis of the damage process and possible causes (Lubelli et al., 2021). There are also cases where specialists are not consulted, and intervention decisions are made by the owner and contractor based on the inspection report from Monumentenwacht (Heinemann & Naldini, 2018). The documentation of these findings is essential for the selection of a successful intervention strategy together with the stakeholders.

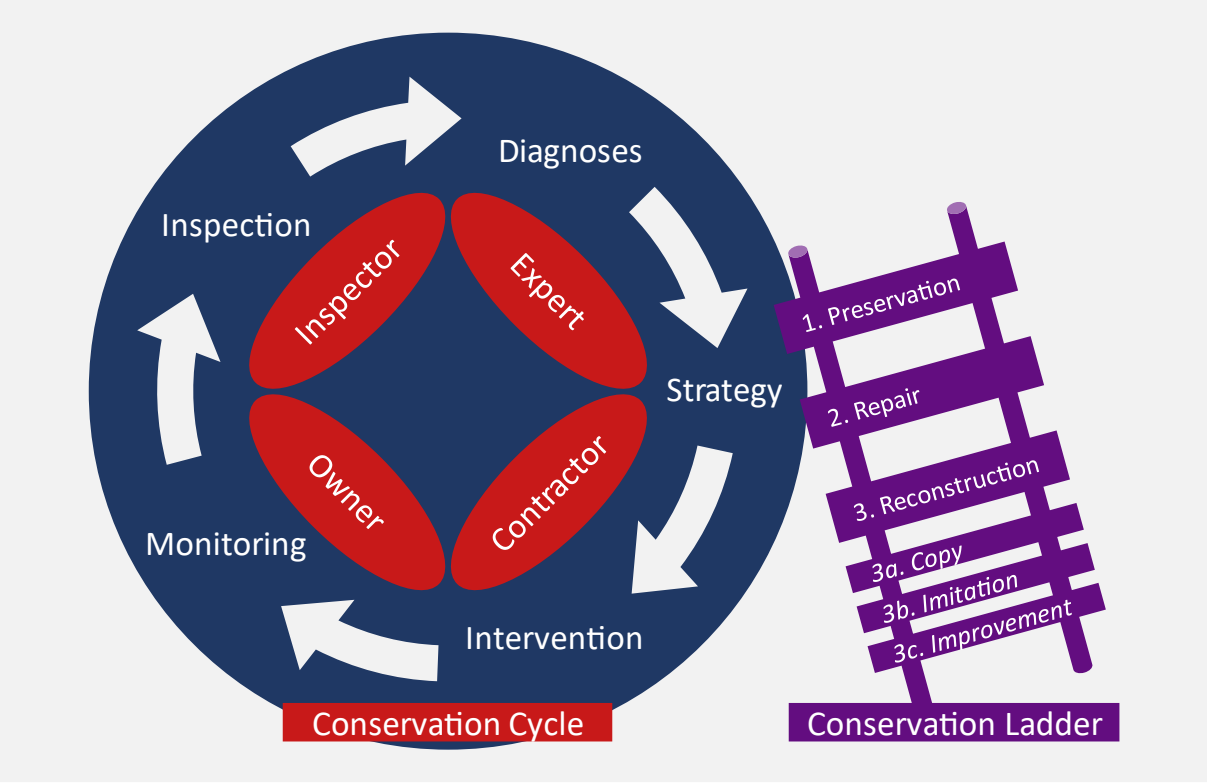


Figure 2: Conservation Cycle & Ladder (Based on Lubelli et al., 2021)

When determining the strategy for intervention, possible solutions should be compared to determine whether they solve the problem and achieve the intended goal of the restoration. The "Conservation ladder" ("Restauratieladder" in Dutch), developed by the ERM can be a helpful instrument in making these decisions, see Figure 2. It is a practical formulation of the principles of historical preservation as set out, among others, in the Venice Charter (ERM, 2021a). The ladder consists of three steps with a preferred sequence: (1) preservation/maintenance, (2) repair and (3) reconstruction. The (3) reconstruction is subdivided into (3a) copy, (3b) imitation and (3c) improvement. The principles state that every intervention, to a greater or lesser extent, compromises the historical value. Therefore, it is wise to consider whether the intervention is necessary at all: refraining from an intervention can be the best choice in certain situations. The following requirements are imposed on an intervention: (1) limit the scope of the intervention or "as much as necessary and as little as possible," (2) the intervention should be durable, prevent or postpone further intervention as much as possible, (3) the intervention is appropriate within the given situation and (4) replacement is preferably done in the same material or technique. After a decision is made, the next step is the planning and execution of the interventions by the contractor in accordance with the ERM guidelines. To close the conservation cycle, all the decisions in the entire process must be properly recorded. This automatically lays the basis of information at the start of the next conservation cycle (Lubelli et al., 2021).

2.2 DATA IN THE CONSERVATION CYCLE: THE CASE OF NATURAL STONE

The previous section made clear that the whole process of the conservation cycle should be documented, as different stakeholders will need to use the information throughout the different steps and the information should be used as input at the start of a new cycle. Information is currently documented using a variety of standards and formats. Some of these are industry standards, but many are also created on a project-by-project basis. As a result, there is still much work to be done to create a common terminology (Naldini et al., 2021). Figure 3 shows some of the information aspects and standards used in the different phases of the conservation cycle. It also illustrates how this creates a complex situation where information is passed on to different stakeholders over the different phases using different standards and data formats.

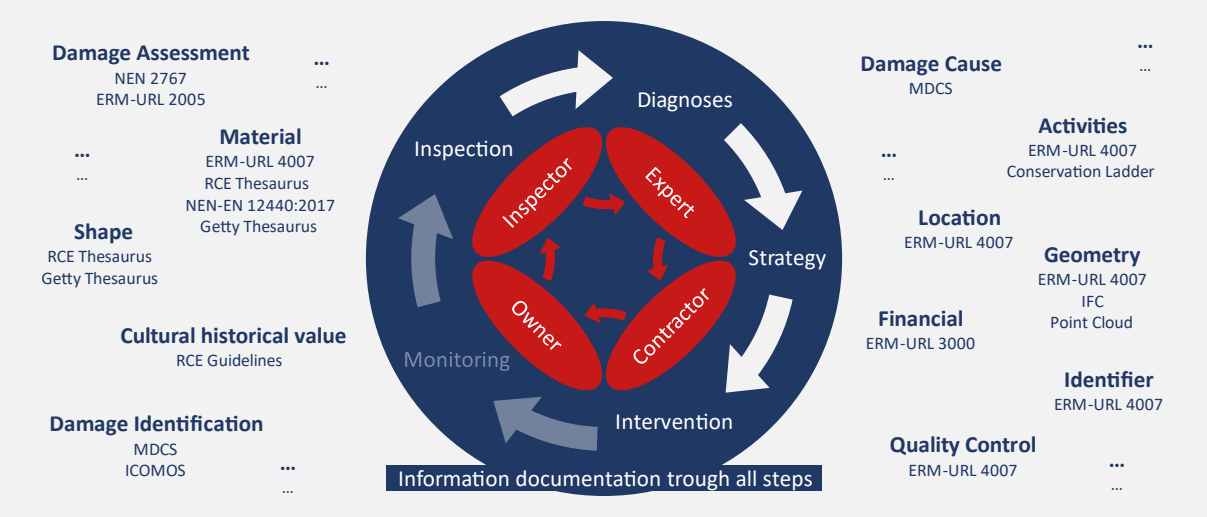


Figure 3: Information Aspects and Standards in Conservation Cycle

The rest of this section will give an overview of all relevant information aspects that should be documented for every object, during the planning and execution of a restoration project involving natural stone. This includes four out of the five steps of the conservation cycle, only monitoring is excluded as this is not part of the scope. The different object data aspects were identified following the processes as described in the relevant ERM guidelines.

2.2.1 Inspection

During an inspection, the technical and historical state of the building are identified. This consists, among other things, of identifying damages, specifying material, construction, technique, detailing, design, colouring, and cultural-historical value assessment. For natural stone in specific, cultural historical value, damage assessment and identification, material and shape were identified as the relevant aspects based on the ERM-URL 4007 (Natural stone restorations).

Cultural-historical Value

Determining the reason for preserving an object is essential in heritage conservation (Franken & Meijer, 2013). This involves identifying the qualities that make it valuable for society. Therefore, an assessment needs to be done by a specialist before any intervention. The Dutch Cultural Heritage Agency has created a set of guidelines for conducting historical research and evaluating an object's cultural-historical value (Hendriks, L. and van der Hoeve, 2009). Before any research, a decision must be made on the observation level of the concluding report. This defines on what scale the value assessment should be conducted, this can for example be done on the building level, for every specific space, building element or building component. The cultural heritage value assessment consists of

five components: (1) general historical values, (2) ensemble or urbanistic values, (3) architectural-historical values, (4) building archaeological values, and (5) values based on the history of use. Every object is valued by its integrity and the rarity of each component. To assess the cultural heritage values objectively, it is important to include a reference basis in the report. The cultural heritage value assessment can be either contextual or internal. The former compares the cultural heritage values of the object with similar objects on a national, regional, or local level, while the latter assesses the components and aspects of the object. A report should be objective and independent of other interests and includes value assessment maps per floor, with a rationale in words and pictures. The maps use three colours to indicate the value level of the object: (1) blue for high monument values, (2) green for positive monument values, and (3) yellow for indifferent monument values.

Damage Assessment

When creating a full condition assessment of a building according to ERM-URL 2005, 2 types of inspections are suggested (ERM, 2020b). Type A assesses condition based on the NEN 2767, it gives every building part a condition score between 1 (perfect) and 6 (very severe) based on the severity of the damages in that specific part and specified descriptions. Damages can be classified as severe, serious, or minor defects (*NEN 2767-1+C1*, 2019). This NEN standard has been used as a basis for the European CEN/TS 17385:2019 which describes a method to assess the physical condition of all types of immobile constructed assets. The type B condition inspection assesses the quality impression of every building part on a scale of 4, from (1) good, where no visible repair is needed to (4) poor, where repair is needed within 2 to 5 years and consolidation measures are urgent.

Damage Identification

To identify damage types both the ERM-URL 2005 (Inspections) and the ERM-URL 4007 (Natural stone restorations) refer to the damage atlases of the Monument Diagnosis and Conservation System (MDCS) (ERM, 2020b, 2021b). MDCS is an online tool that facilitates the diagnosis of damage and supports the conservation of historical buildings (van Hees & Naldini, 2020). The MDCS damage atlases contain terminology on damages related to brick, mortar, natural stone, plaster and concrete (TNO et al., n.d.). Internationally, the Illustrated Glossary on Stone Deterioration Patterns, published by ICOMOS International Specialist Committee for Stone (2008), relies partly on the MDCS and concentrates solely on natural stone (Lubelli et al., 2021). The damage types of the MDCS atlases are presented in a hierarchical order, ranging from surface damage to material loss and loss of cohesion. Cracks are identified by patterns in individual materials or within the construction. The ERM-URL 2005 prescribes that one should clarify damages visually by documenting photos (overview, location and details) and/or simple sketches.

Shape

Besides material, the shape of the element is also relevant. The shapes often vary from highly detailed ornaments to rectangular blocks. The cultural history thesaurus of the RCE defines a broad range of shapes that natural stone may have (*RCE Erfgoedthesaurus*, n.d.). Furthermore, the Getty Art & Architecture Thesaurus (AAT) also contains numerous definitions of different natural stone shapes (J. Paul Getty Trust, n.d.).

Material

When deciding about an intervention it is important to determine the type of natural stone. The relevance can be expressed in multiple ways: (1) historically, as it can help determine the stone's origin, (2) technically, as it assists in explaining any observed signs of ageing based on the stone's type and origin and in (3) intervention selection, as appropriate conservation techniques and materials are

dependent on the type of stone (Lubelli et al., 2021). The ERM-URL 4007 gives in appendix 3 a taxonomic overview of relevant natural stone classes (ERM, 2021b). The first version of this list was published in 2001 by the Dutch Cultural Heritage Agency (RCE) (RCE, n.d.-a). These materials are also part of the broader material section of the cultural history thesaurus of the RCE (*RCE Erfgoedthesaurus*, n.d.). Furthermore, in a European context, the NEN-EN 12440:2017 provides a list of most European natural stones grouped by country of origin (*NEN-EN 12440*, 2017). Internationally the Getty Art & Architecture Thesaurus (AAT) also has a section on materials (J. Paul Getty Trust, n.d.). However, it does not go to the same depth in terms of different types.

2.2.2 Damage Diagnoses

Damage Cause

It is only possible to determine a successful intervention when the damage type, cause and mechanism are known (Lubelli et al., 2021). The MDCS atlas not only allows uniform identification of the damages found but also gives insight into potential causes of the damage. For every type of damage in the MDCS, suggestions are given on possible causes, assisting in determining the root cause of the damages. Often, conducting only a visual inspection fails to provide a full diagnosis. In these cases, it is essential to carry out specific investigations to identify the damage mechanism or its underlying cause.

2.2.3 Intervention Strategy

When all the information about the existing state is gathered, a decision can be made on the need for any type of intervention. Large and complex restoration projects, especially in the case of inner-city churches with high site design costs, consider restoration horizons of 25, 30, or 50 years to avoid frequent scaffolding. In these cases, intervention decisions are not only taken based on the current state but also on the anticipated technical developments (Lubelli et al., 2021). The horizon also establishes a minimum lifespan for the planned intervention. Next to the horizon, several factors influence possible interventions, including but not limited to the availability of materials, techniques, site accessibility, vulnerability, historical significance, level of ambition, and budget constraints.

Conservation Ladder

The ERM conservation ladder provides valuable guidance for selecting between different options for a particular intervention (Lubelli et al., 2021). Its basic principles require an intervention to do as much as necessary but as little as possible. The intervention should be a solid solution, should not lead to damaging surrounding objects and preferably uses the same material or technique for replacement. Each intervention should therefore be classified using the steps of the conservation ladder (ERM, 2020a). In the practice of stonework in the restoration of monuments, the category of preservation (1) will not involve adding a volume of stone to the part, repair (2) will involve adding a small volume of stone, while reconstruction (3) will involve replacing the entire part (ERM, 2021b). In preservation, the focus is on the conservation of the current state, by slowing the decay of the historical stonework. During a repair a missing part is fully or party resorted using a piece of natural stone, mortars or adhesives. In a reconstruction of stonework, there is the possibility to copy, imitate or improve the stone. In copying (3a) the shape, as well as the material, machining and fastening, are done according to the part being replaced. Tools that were not available in the past are often used to work natural stone. In that situation, we speak of an imitation (3b) since other techniques are used. A similar effect occurs when the type of stone for the replacement part is not the same as that of the part to be replaced. This often happens in practice because the stone type of the part to be replaced is sometimes no longer available. With imitation (3c) the aim is to make the new situation more durable than the old situation. For example, if the stone type of the part to be replaced is still available, a

different stone type may still be chosen for the replacement part, as this is expected to make the replacement part last longer.

Activities

Appendix 4 in ERM-URL 4007 gives a taxonomic overview of relevant activities that are part of the intervention process. The text providing an overview of the treatment, handling and processing of natural stone was created around 2000 by a group of experts on the initiative of the predecessor of the RCE (ERM, 2021b). Most of these types of activities related to natural stone are also part of the broader activities section of the cultural history thesaurus of the RCE (RCE Erfgoedthesaurus, n.d.). Every intervention will exist out of a series of activities, which activity is dependent on the type of intervention. Some preservation or repair interventions can be done in situ. However, in many cases, natural stone elements will have to be removed or dismantled, with the latter being put back in after treatment or repair and the former will be replaced by a reconstruction. Dismantlement can also be the result of measures on surrounding elements. Interventions that require removing or dismantling the element consist of relatively more activities, as the elements need to be transported and stored. The number of completed activities of an intervention provides insight into progress and could be used to determine a status.

2.2.4 Intervention

To carry out the various activities on site, different information will need to be documented. Based on the ERM-URL 4007 (Natural stone restorations), the aspects that are listed below have been identified.

Geometry

Dimensions are essential when conducting most interventions. Geometry can be used to determine the location of each element by its relationship to other elements. Dimensions can be recorded in both 2D and 3D drawings. The maximum margin of error depends on the use of dimensioning. For geometry intended to only indicate the location of an object, it may be much higher than for geometry intended to produce a new element. In the construction industry, a broad range of 2D and 3D geometry descriptions are used. Consequently, there can be multiple geometry descriptions of the same object. There are differences in format, measured or represented accuracy, degree of detailing, spatial dimension (2D or 3D) and the geometry representation type (Bonduel, 2021). Pauwels et al. (2022) distinguish four types of geometric data: constructive solid 3D geometry, less detailed 3D boundary representations (BREPs), 2D geometry based on lines and points, and finally point cloud models. This also explains the broad range of file types and applications that are used for creating and handling geometry. The description of geometry often consists of long lists of data points, representing the points, polylines and surfaces that combined describe the object.

When an existing stone is replaced, each component must be measured and drawn at a 1:1 scale or moulds must be made on-site (ERM, 2021b). The measurements must not only include the part that needs to be replaced but also the surrounding work to ensure a good fit with the new block. When dismantling and rebuilding natural stone parts, the whole structure must be measured and drawn at a scale of 1:10. Digital measurement and scanning techniques can be used to take measurements. However, it should be noted that producing an end product directly from a point cloud may result in a significantly different product than parts that were measured and drawn based on the original stylistic language. Direct copying of existing objects is not recommended either, as weathered objects do not represent their original shape.

Identifier

All removed or dismantled stone elements should be identifiable with a unique physical identifier, ensuring that it can be identified on the building site, at the stonemasonry, in storage or when being transported between these places (ERM, 2021b). The traditional approach for this is having a numbered metal tag on each element that corresponds to a number on a construction drawing. However, there are also examples where QR codes have been used for this (Beerda, 2022). During the restoration works, it must be known where each removed or dismantled stone element is located, e.g., in storage, at the construction site or with the stonemason (ERM, 2021b). Identifiers are used to create overviews and identify the objects at the specific locations.

Quality & Financial

During the restoration works, quality is ensured using an inspection plan (ERM, 2020a). ERM-URL 4007 prescribes that there should be minimum inspection moments when a new block is completed in the stonemason's workshop, when parts arrive on site and, of course, during the final handover of the work (ERM, 2021b). Meeting this final quality check also initiates the billing process. As the interventions that are done may deviate from the initial strategy, there may be deviations between the budget and the actual billing. The ERM-URL 3000 prescribes that the contractor must define a procedure for additional and reduced work (ERM, 2020a). In the case of natural stone, it is interesting to have insight into the budgeted and invoiced costs for each element.

2.3 CONCLUSION

Historical buildings in the Netherlands hold cultural and economic value, attracting tourists and preserving the country's heritage. The conservation of these buildings is regulated through self-regulation and guidelines established by the Dutch Cultural Heritage Agency (RCE) and the Foundation ERM. The ERM guidelines provide technical specifications and quality standards for restoration work, ensuring the preservation of historical value. The process involves inspections, damage assessment, identification of cultural-historical value, determination of appropriate materials and techniques, and decision-making using the "Conservation ladder." The Monumentenwacht organisation conducts regular inspections and assists private owners in determining necessary maintenance activities. The whole conservation cycle should be documented to provide valuable information for future restoration projects. For natural stone restoration projects, aspects such as cultural-historical value, damage assessment, damage identification, material type, and shape need to be documented. Documenting these aspects is done using a wide range of standards and data formats, creating a situation that is difficult to manage.

3 CONNECTING DATA IN THE CONSTRUCTION INDUSTRY

This second part of the literature research focuses on how connected data, the combination of linked data technologies with Building Information Modelling, can assist in managing the difficulties arising from the wide range of data formats and standards encountered during information documentation in natural stone restoration projects.

3.1 BUILDING INFORMATION MODDELING

Digitalisation has transformed most global industries, resulting in a significant increase in productivity. The Architecture, Engineering, Construction (AEC) industry, is often seen as lagging, partly because of the fragmented nature of the industry (Borrmann, König, et al., 2018). In the AEC industry, multiple parties or stakeholders collaborate to create buildings, and in bigger projects, the number of parties can grow and change rapidly. However, the interdisciplinary stakeholders all have other preferred software packages, which causes the need for smart ways of collaborating. Building Information Modelling (BIM) is the collaboration process of creating a digital representation of a building structure that is data-rich and object-oriented. The resulting BIM models can be used to extract views and data to produce information that can be used to make decisions and improve the design, construction, operation and maintenance of the structure. By using the BIM approach for construction projects, the information loss over the different phases is reduced significantly. BIM has gained much popularity in recent years and is becoming widely accepted in the industry.

3.1.1 Collaborating using BIM

As the AEC industry increases the use of collaborative Building Information Modeling (BIM), the maturity of the implemented workflows across the industry can be measured with the UK BIM maturity model developed by Bew & Richards (2008). This model outlines four levels of BIM maturity, with higher levels indicating deeper integration of BIM, see Figure 4. At maturity level 0, there is no coordination in data exchange. Level 1 involves collaborative efforts among various parties through the exchange of 2D and 3D geometric files, which are stored locally. Level 2 represents a scenario where companies utilise complex BIM models, sharing them with other parties and disciplines within a common data environment (CDE). Level 3 suggests employing an integrated solution based on open standards like Industry Foundation Classes (IFC), where all project data is connected. All information is exchanged online and seamlessly integrated across disciplines and parties, with a shift from file-based collaboration to object-based collaboration.

To achieve maturity level 3, the adoption of open standards and a transition from file-based to object-based collaboration is essential. However, currently, the industry predominantly operates at levels 0, 1, or 2, varying based on geographical location and project size. Adoption of BIM level 3 remains relatively rare in today's landscape and is seen as a target for the future (Borrmann, König, et al., 2018).

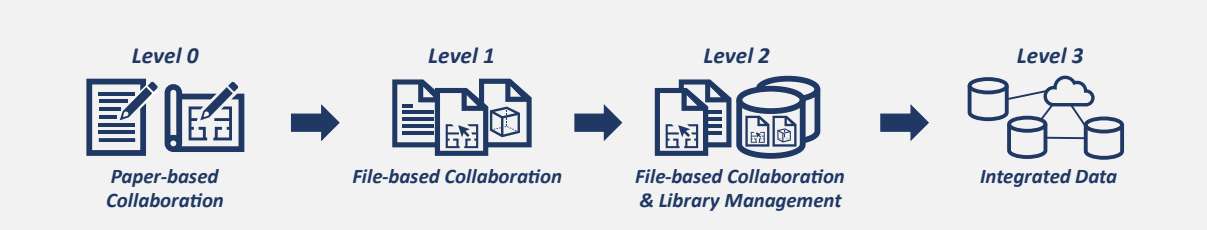


Figure 4: BIM Maturity (based on Bew & Richards, 2008)

The highly fragmented nature of the industry requires standardised and vendor-neutral exchange formats for growing the maturity of collaboration and information exchange. BuildingSMART, a non-profit organization, plays a central role in addressing interoperability challenges within the AEC industry by initiating, developing, creating, and promoting open digital standards for BIM processes, such as IFC, BCF and IDS. Most importantly the Industry Foundation Classes (IFC), is an open neutral data model that allows users to exchange digital building models (Borrmann, König, et al., 2018). This enables 'Open BIM' workflows, where software from different vendors can be used in conjunction with each other. This contrasts with 'Closed BIM', where proprietary data formats are used for data exchange, which greatly limits the software products that can be used.

The Dutch construction industry, like most others, works mainly with the 'Aggregate BIM' concept (Van Berlo et al., 2012). Hereby there is a central or coordination model as a result of the periodic aggregation of multiple disciplinary BIM models. These models are created by the various disciplines involved, such as the architect and the structural engineer. A Common Data Environment (CDE) is often used to exchange IFC data models in a central collaboration environment, as the centralisation of data and information is the basis for all collaborative processes (Preidel et al., 2018). The coordination model, created by combining the different disciplinary IFC models in the CDE, is for example used for clash detection, the generation of cost documents, planning, etc.

In a CDE, information requirements, status definitions and a workflow for sharing and approval processes are used to organise the production and exchange of information during different project stages, see Figure 5. This happens in dedicated access areas for the different project stakeholders (Preidel et al., 2018). This concept is based on the methodology presented in ISO 19650 (Scheffer et al., 2018). Many different information containers can exist within one CDE. An information container contains files or parts thereof, according to the agreed formats and structures in the information requirements. The ISO 19650 (2018) states that an information container can have three different states, namely work in progress, shared or published, in addition, an old version can be archived (Preidel et al., 2018). The transition from one state to another is subject to an approval process, where requirements are checked, and the information is compared to other relevant information.

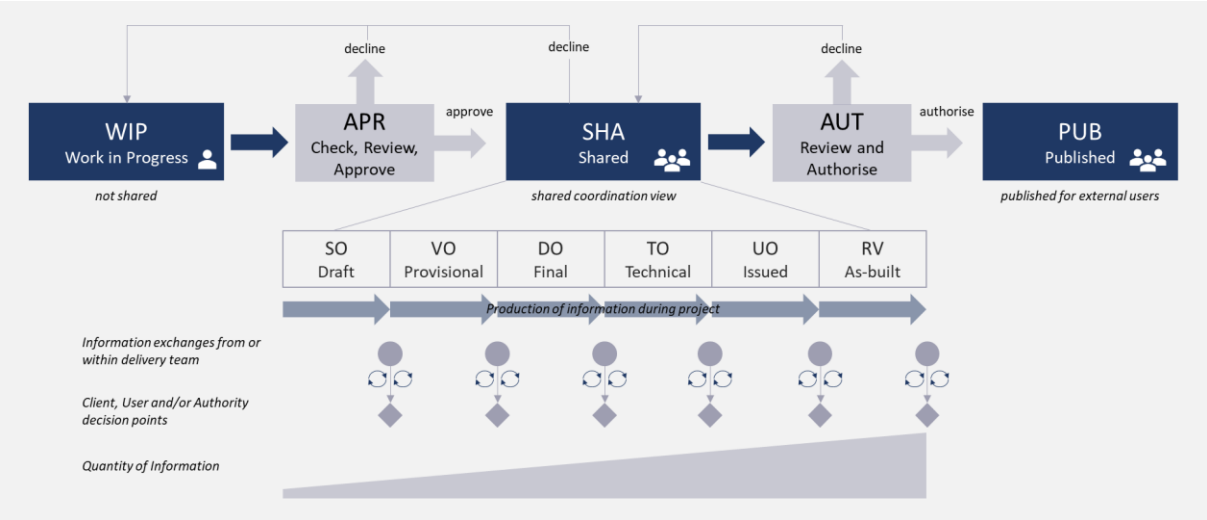


Figure 5: Information Management Workflow, based on Klemt-Albert et al. (2018) and ISO 19650-1 (2018)

3.1.2 Industry Foundation Classes

The Industry Foundation Classes (IFC), maintained by BuildingSMART, is an open international standard for exchanging and sharing Building Information Model (BIM) data among software applications used in the AEC industry (Borrmann, Beetz, et al., 2018). The IFC standard is defined and standardised in ISO 16739-1:2018. The IFC data schema is object-oriented and can describe building-related information using an object-based hierarchy that consists of a large number of entities. By following the complex IFC data structure, entities can be used to describe objects such as a 'wall' or a 'window', see Figure 6.

The IFC standard enables the exchange of building information between different software applications, which would up until then all save their contents in a proprietary file format that is not readable by other programs. Despite its complexity and challenges, the standard is seen as the best solution for a vendor-neutral collaborative workflow (Borrmann, Beetz, et al., 2018). As IFC is open source, it can be used in a wide range of databases and software packages that support the importing and/or exporting of IFC files. The IFC standard is a standardised digital description of the built asset, and it promotes vendor-neutral workflows across a wide range of software platforms, for many different use cases.

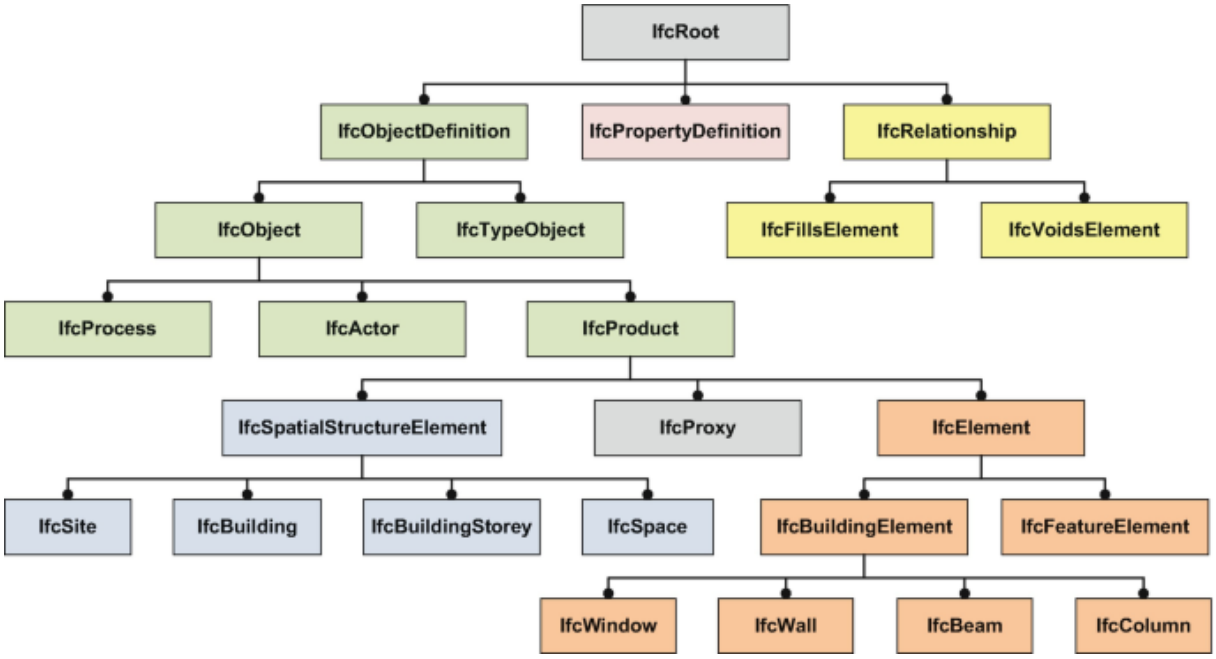


Figure 6: Part of the IFC Data Model (Borrmann, Beetz, et al., 2018)

3.2 LINKED DATA

Before data can be used for optimising processes, context needs to be connected to the data, only then will the data be useful to generate insights. These connections can be made in diverse ways. For example, in relational databases connections are made between different instances, giving context to the instance. The IFC schema also gives context to objects. However, recently there has also been a lot of research into the use of semantic web technologies for this purpose.

3.2.1 Semantic Web Technology and Linked Data

In 2001, Berners-Lee et al. (2001) proposed the concept of the semantic web. This would transform the World Wide Web (WWW) or the Web of Documents, to a Web of Data. This is done by giving semantics to information and making it machine-readable. The result is a web of data that can be understood by both humans as well as computers (Berners-Lee et al., 2001). The WWW is home to various documents such as web pages, images, and pdf-files. Users can explore the internet by entering a URL or clicking on hyperlinks that interconnect documents on the web. The semantic web transforms it into a web that is no longer only a collection of human-readable documents, but rather a vast, decentralised database of machine-readable data.

Linked Data in the Semantic Web Stack

The semantic web is based on using semantic web technologies (SWT). A number of these technologies have been standardised by the World Wide Web Consortium (W3C) in the semantic web stack. This describes the underlying technologies of the semantic web. Linked data is a subset of these technologies that can be used to create linked data graphs (Pauwels et al., 2017) (Pauwels et al., 2022). The result will be a knowledge graph where relationships between data objects are defined. Within this network of relations, every object will have a unique identifier or Uniform Resource Identifier (URI). The URI is a similar concept to the URL, the latter identifies things that exist on the web, such as a web page, while the former identifies things that exist using the web, such as an object in the real world. According to Tim Berners-Lee (2006), all linked data should be based on four principles: (1) use URIs as names for things, (2) use HTTP URIs so that people can look up those names, (3) when someone looks up a URI, provide useful information, using the W3C standards and (4) include links to other URIs so that they can discover more things. Besides URIs, linked data also uses the Resource Description Framework (RDF), a formal language for describing information, the Resource Description Framework Schema (RDFS) and the Web Ontology Language (OWL), both base ontologies used to define semantics (meaning) of other ontologies, and SPARQL, a language for creating queries on the graphs.

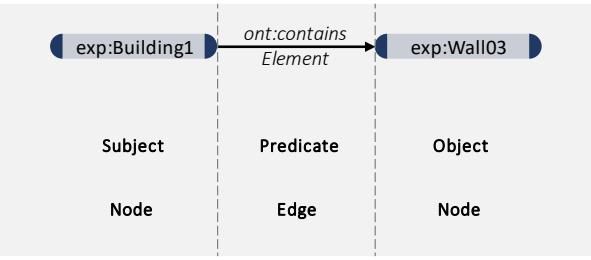
Berners-Lee (2006) also created the 5-star model of linked open data, providing a system to determine the maturity of linked data that is shared under an open licence. 1-star data is available on the web using any format with an open license as only requirement. 2-star data adds the requirement that the data should be structured, for example, an Excel file. Achieving 3-star data is done by only using non-proprietary formats, using CSV instead of Excel. For 4-star open data, W3C open standard and URIs should be used. 5-star open data should also be linked to other datasets. Not all linked data has to be open, however, creating open data creates more opportunities to link to other open datasets.

3.2.2 Linked Data Technologies

Resource Description Framework (RDF)

In the semantic web, the Resource Description Framework (RDF) is used to express knowledge. It consists of 'triples', triples are very simple statements that consist of three parts, a subject, predicate and object (Cyganiak et al., 2014). The directional relationship is expressed with the predicate, it is directed from subject to object. This creates data that can be understood by machines as well as humans. A graph can be visualised using nodes and edges, where two nodes (subject and object) are connected using an edge (predicate), creating a triple. There are various syntaxes to represent RDF triples, such as N-triples, XML, JSON-LD, and Turtle being the most widely recognised ones. The different serialisations or formats each have their own advantages. During the remainder of this thesis, the Terse RDF Triple Language (Turtle) will be used, as it is commonly used and relatively easily humanly readable.

Figure 7 shows an example of the structure of an RDF triple both visually and in Turtle. Every individual data object in this example has a URI. Most URIs include an address that links the data object to a webpage where the definition can be found. Prefixes are used to abbreviate the complete URIs into a structure where a prefix (*ont:)* represents a web address (*http://www.example.org/#)* and is only followed by the name *(ont:contains)* of the data object (*http://www.example.org/#contains*). This makes both visualisations and Turtle files better to read.

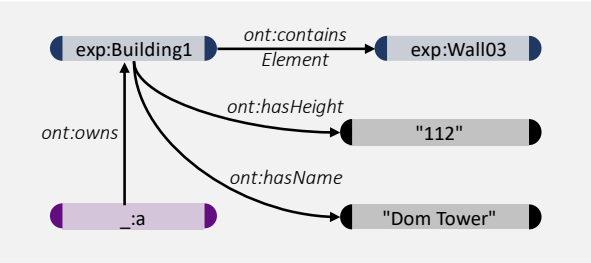



```
#Prefixes
@prefix ont: <http://www.example.org/#> .
@prefix exp: <http://www.mydata.org/#> .

#RDF Triple (Subject → Predicate → Object)
exp:Building1 ont:containsElement exp:Wall03 .
```

Figure 7: Examples of Linked Data Technologies: RDF Triple

In addition to URIs, a graph has literals that represent basic values such as strings, booleans, integers, and various other data types. However, literals can only be used in the position of an object in the triple. The third type of RDF term, next to URIs and literals, is the blank node. This can be used to express that a certain relationship exists, without explicitly naming it. This creates a node, subject or object, that can only be identified within a certain graph or file. Figure 8 shows an example of the usage of literals and blank nodes. The graph states that the height (*112*) of the building (*exp:Building ont:hasHeight*) is of the datatype *xsd:integer*. The name of the building is given in a special language-tagged string, denoting the language of the specific string, English in this example. The blank node is used to define that something or someone owns the building, without identifying the data object with a specific URI.




```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix ont: <http://www.example.org/#> .
@prefix exp: <http://www.mydata.org/#> .

exp:Building1
            ont:containsElement exp:Wall03 ;
            ont:hasHeight "112"^^xsd:integer ;
            ont:hasName "Dom Tower"@en .

_:a ont:owns exp:Building1 .
```

Figure 8: Examples of Linked Data Technologies: RDF Literals and Blank Nodes

Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL)

Data in itself is not structured by default. Within the semantic web, ontologies are used to give data structure and meaning with semantics, making it interpretable for a computer. An ontology can be described as a *'formal, explicit specification of a shared conceptualization'*, which is a definition by Studer et al. (1998). In other words, it should be a machine-readable abstract model description that is agreed upon by a group. Information can be structured using different ontologies, each covering a specific domain. There can also be relations between different ontologies, in this way, knowledge from different domains can be connected to each other.

The Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL), both part of the Semantic Web Stack, can be used to build ontologies in the RDF triple structure. Creating comprehensive ontologies through RDFS and OWL allows for the expression of meanings that machines and software can understand and utilise. SWTs typically operate under the Open World Assumption (OWA), which means that if something is not known to be true, it is not necessarily false but is unknown (Pauwels & Terkaj, 2016). This approach is logical for the SW since the goal is to publish all information on the web, which is never finished. In contrast, traditional software applications, including BIM authoring tools, often adopt the Closed World Assumption (CWA), which assumes that if something is not known to be true, it is false. The concept of ontologies in a semantic web context is often explained using the concept of the Terminology Box (TBox) and Assertion Box (ABox). The TBox (terminology layer) is used to define classes in a domain and is used to set rules and restrictions on the properties that are used as a predicate. The ABox (data layer) contains instances that follow the ontology defined using the TBox, see Figure 9.

RDFS provides the basic elements for describing an ontology, such as classes, subclasses, comments and datatypes. Figure 9 shows a few examples of the use of the *rdf:* and *rdfs:* ontologies. In the example, *exp:Building1* is classified as a *bot:Building* using *rdf:type* as the predicate. The property *rdf:type* is often abbreviated to *a*, as can be seen in the Turtle file. The property *rdfs:subClassOf* is used to point to which super-class the *ont:StoneWall* belongs. This hierarchical structure of classes can be used to make statements about a super-class, these will automatically hold for all its subclasses. In the figure the *rdfs:range* and *rdfs:domain* properties are used to define the classes that the instance subject and object should have when using the *ont:containsElement* property. Because *ont:StoneWall* is a subclass of *ont:Element*, it can also be used on *exp:Wall03*.

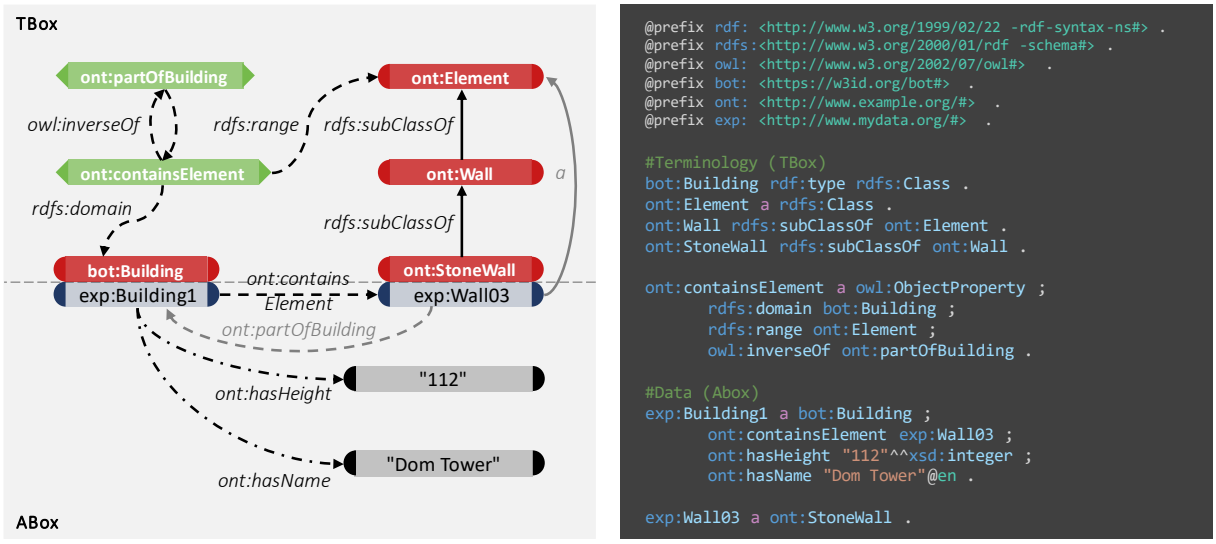


Figure 9: Examples of Linked Data Technologies: RDF Ontologies

OWL introduces constraints related to cardinality, property characteristics, and equality. Figure 9 shows two examples of the use of OWL. In the Turtle is stated that the *ont:containsElement* property is of the class *owl:ObjectProperty*. This statement defines that the property relates URIs to URIs, whereas an *owl:DatatypeProperty* relates URIs to a literal. The second example is the use of the *owl:inverseOf* property. The statement defines that the *ont:containsElement* property has an inverse in the *ont:partOfBuilding* property. This means that two instances that have the *ont:containsElement* relation also have a *ont:partOfBuilding* relation in the other direction. This also introduces the concept of inference, the ability to identify relationships while not explicitly stated. Because of the inverse relationship between *ont:containsElement* and *ont:partOfBuilding,* it can be inferred that the relationship between the building and the element is bi-directional. Based on the class hierarchy, the following statement can be deduced: *exp:Wall03 rdf:type ont:Element.*

SPARQL

SPARQL Protocol and RDF Query Language (SPARQL), which is the standardised query language by W3C, can be utilised to query any set of RDF triples that consist of both the data layer and the terminology layer (W3C, 2013). SPARQL queries can be categorised into two main types: read queries and update queries. The type of query can be determined by the keywords used. Read queries use keywords such as SELECT, CONSTRUCT, DESCRIBE, or ASK, while update queries use the keywords INSERT and/or DELETE. Most queries include a WHERE clause, which outlines a specific graph pattern that permits the use of variables for the subject, predicate, and/or object positions. This enables the representation of a portion of the query as an RDF graph that contains one or more unknowns (variables). Queries can be further specified using options such as LIMIT, FILTER and OPTIONAL. Subsequently, the query processing engine will attempt to identify results for the variables within the RDF dataset(s) that correspond to the entire pattern.

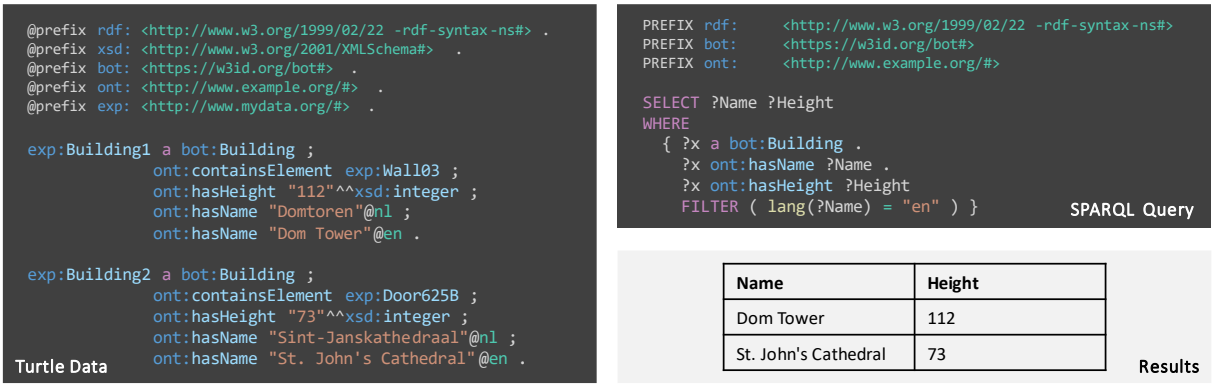


Figure 10: Examples of Linked Data Technologies: SPARQL

Update queries enable the editing of data by adding, removing or modifying triples in the source graph. Read queries can be used to return data. SELECT queries return results in a tabular format by default, while CONSTRUCT queries return RDF graphs that correspond to the graph pattern in the WHERE clause or a new pattern using matched variables. DESCRIBE queries return an RDF graph describing a specific resource, and ASK queries return a boolean value indicating the existence of the pattern in the WHERE clause. Figure 10 shows an example of a SELECT query. The data file show different statements that are made about two instances of a *bot:Building*. In the query *?Name* and *?Height* are defined as the desired return values. The WHERE clause is used to determine the pattern the query should follow. In the first step, it looks at all instances that are of the *rdf:type bot:Building,* at these instances the query looks at the *ont:hasHeight* and *ont:hasName* predicates to find the *?Name* and *?Height*. Finally, a FILTER option is applied, which results in only the instances that have a *?Name* in the English language being included in the results.

SHACL

The Shapes Constraint Language (SHACL) is a W3C standardised language for defining constraints. It gives features to express conditions that constrain the values, types, ranges, patterns and combinations of properties in RDF graphs (Knublauch & Kontokostas, 2017). SHACL also defines an RDF vocabulary to automatically create detailed validation reports as RDF graphs. SHACL can be used to validate RDF graphs that are obtained by any means, such as from files, HTTP requests, or datasets. SHACL Core can be used to define the syntax and structure of shapes and constraints and link them to data nodes. SHACL-SPARQL extends SHACL Core with SPARQL queries. Validating data using shape languages, such as SHACL, can ensure that exchanged data conforms to the expected structure and constraints of receiving applications. If the data violates constraints it can be rejected and a list of violations can be forwarded to the sender or a repairing application. Figure 11 shows an example of using SHACL. In the SHACL shape is defined that every instance of a *bot:Building* should at least have the *ont:hasHeight* and *ont:hasName* properties by setting the *sh:minCount* on 1. Additionally, a requirement is set that *ont:hasName* objects should be in either English or Dutch. It finally uses an OWA, otherwise, all other statements in the data graph would be violations. The result gives one violation of the shape, namely, there is no statement given with the *ont:hasHeight* at *exp:Building1.*

```
@prefix rdf: <http://www.w3.org/1999/02/22 -rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>   .
@prefix bot: <https://w3id.org/bot#>   .
@prefix ont: <http://www.example.org/#>   .
@prefix exp: <http://www.mydata.org/#>   .

exp:Building1 a bot:Building ;
              ont:containsElement exp:Wall03 ;
              ont:hasName "Domtoren"@nl ;
              ont:hasName "Dom Tower"@en .

exp:Building2 a bot:Building ;
              ont:containsElement exp:Door625B ;
              ont:hasHeight "73"^^xsd:integer ;
              ont:hasName "Sint-Janskathedraal"@nl ;
              ont:hasName "St. John's Cathedral"@en .
```

Turtle Data Graph

```
@prefix sh:  <http://www.w3.org/ns/shacl#>   .
@prefix rdf: <http://www.w3.org/1999/02/22 -rdf-syntax-ns#> .
@prefix bot: <https://w3id.org/bot#>   .
@prefix ont: <http://www.example.org/#>   .
@prefix exp: <http://www.mydata.org/#>   .

exp:Shape01
    a sh:NodeShape ;
    sh:targetClass bot:Building ;
    sh:property [
        sh:path ont:hasHeight ;
        sh:minCount 1 ;
    ] ;
    sh:property [
        sh:path ont:hasName ;
        sh:minCount 1 ;
        sh:languageIn ( "en" "nl" ) ;
    ] .
    sh:closed false .
```

Turtle Shape Graph

```
[           a sh:ValidationResult ;
            sh:resultSeverity sh:Violation ;
            sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
            sh:sourceShape _:n236 ;
            sh:focusNode exp:Building1 ;
            sh:resultPath ont:hasHeight ;
            sh:resultMessage "Less than 1 values" ;
] .
```

Turtle Validation Report

Figure 11: Examples of Linked Data Technologies: SHACL

3.2.3 Other Linked Data Standards in Construction

SKOS

SKOS (Simple Knowledge Organisation System) is a W3C-recommended ontology for representing and linking knowledge organisation systems such as taxonomies, thesauri, and classification schemes in a machine-readable format (*SKOS Simple Knowledge Organization System Reference*, n.d.). SKOS provides a standard way to express the structure and semantics of these systems, making them easier to share and reuse across different applications and domains. SKOS is based on RDF, which allows SKOS vocabularies to be integrated with other linked data sources. It provides a set of concepts and relationships that can be used to describe the relationships between concepts in a knowledge organisation system, such as broader/narrower relationships, related concepts, and preferred labels. SKOS is widely used by various organisations, such as libraries, museums, archives, government agencies and research institutions, to publish and manage their data online (*SKOS | Forum Standaardisatie*, n.d.).

NEN 2660

NEN 2660 is a Dutch standard that describes terminology and general rules for an information system for the building field. The standard lays down rules for entities, attributes and models that can be used to represent information about buildings and their components (*NEN 2660-2*, 2022). The standard consists of two parts: NEN 2660-1 and NEN 2660-2. NEN 2660-1 defines the basic concepts and principles of information modelling for the built environment. NEN 2660-2 gives an implementation of NEN 2660-1 using a linked data approach. One of the things the NEN 2660-2 defines is the relationship between functional and technical entities. A functional entity is a concept that describes the function of a system or process, while a technical entity is the physical implementation of that function. A functional entity could be the design of a building, while a technical entity would be the physical construction of that building. The design of the building describes the external behaviour of the system, while the physical construction is the implementation of that design. The technical entity implements the functional entity.

ICDD

As discussed before, a standard for information exchange in the construction industry is described in ISO 19650. The contents of an information container, in the standard, may be organised according to a specific structure. For example, with an Information Container for linked Document Delivery (ICDD), described in the EN ISO 21597-1:2020, it is possible to describe a set of interlinked construction documents in a structured way using semantic web technologies (ISO/TC 59/SC 13, 2020). An ICDD consists of three main components in a zipped folder structure, an ontology folder which describes the schema used in the files, a payload folder containing the documents and a link set folder describing relationships between the documents (Senthilvel et al., 2020). In an ICDD semantic links can be created between both RDF graphs and non-RDF files, such as images, spreadsheets and IFC files. Links can also be made to the sub-documents level, such as GUIDs within an IFC or pixel zones of an image (Werbrouck et al., 2019). Hagedorn et al. (2023) recognise the ICDD concept as a way to provide a structured data package, which can be handed over at the end of a construction project, without limitations on data type. However, they do see it only as an intermediate step to a fully linked data workflow without using files.

3.2.4 Developing Ontologies

Donkers et al. (2022) compared several ontology development methodologies and concluded that many similarities can be found in the different approaches. The first step is to determine the scope and purpose of the ontology, for example by formulating competency questions. Next, domain knowledge is acquired through a literature review, leading to the specification of classes and relationships. These are then used to build the ontology in an ontology editor such as Stanford Protégé[1]. The resulting ontology is then evaluated through various use cases and SPARQL queries to test the competency questions. The ontology was then integrated with existing standards and ontologies. Finally, the ontology should be published including documentation. The same research also suggests evaluating the ontology against a set of criteria, namely: query efficiency, practical applicability, pattern efficiency and extensibility. Three of the aspects that are important to consider in this regard are discussed below.

[1] *https://protege.stanford.edu/*

Taxonomy and Topology in Ontologies

An ontology created with RDFS and/or OWL provides a mechanism for defining a shared conceptualisation of the semantic representation of data instances. It focuses on two fundamental components: (1) classification, the identification of the instances, and (2) properties, attributes providing details about the instance through relations with other instances. This also introduces the concepts of taxonomy and topology in the context of ontologies. Taxonomical elements in an ontology focus on the hierarchy of different classes within the ontology. Topological elements in an ontology describe how different classes can be related or connected in a certain domain. To ensure flexibility and extensibility, one should take a modular data structures approach for representing building information. This concept of modularity holds for both the depth as well as the width of the domain. The former focuses on extending existing ontologies with specific classifications and properties for a subdomain, and the latter on creating relations between different ontologies.

Property Levels

Property data can be stored in various ways, with Level 1, Level 2, and Level 3 property modelling being popular distinctions based on the number of edges between the product and its corresponding property value (Rasmussen et al., 2018). As the level increases, so does the complexity of the property model, affecting the query and pattern efficiency, see Figure 12. A level 1 property can primarily be used for relatively simple instances and their properties are represented in the form of strings, whereas Level 2 and 3 properties are employed for storing additional metadata and state information. At Level 1, describing metadata is not feasible, although the unit could still be assigned using a custom datatype or default unit description for the property value. Adding more metadata is challenging at this level. In contrast, Level 2 properties enable the addition of metadata, such as the property's originator, and the creation of a PropertySet to enhance querying capabilities. For example, a SPARQL query could leverage this PropertySet to find all properties associated with a specific subject (Donkers et al., 2022). Level 3 properties offer a means to capture the complete historical evolution of the property value, including state changes over time. Rasmussen et al. (2018) proposed the Ontology for Property Management (OPM) to enable Level 3 properties. It is always possible to simplify all Level 3 properties to Level 2 or Level 1, this would however result in data loss.

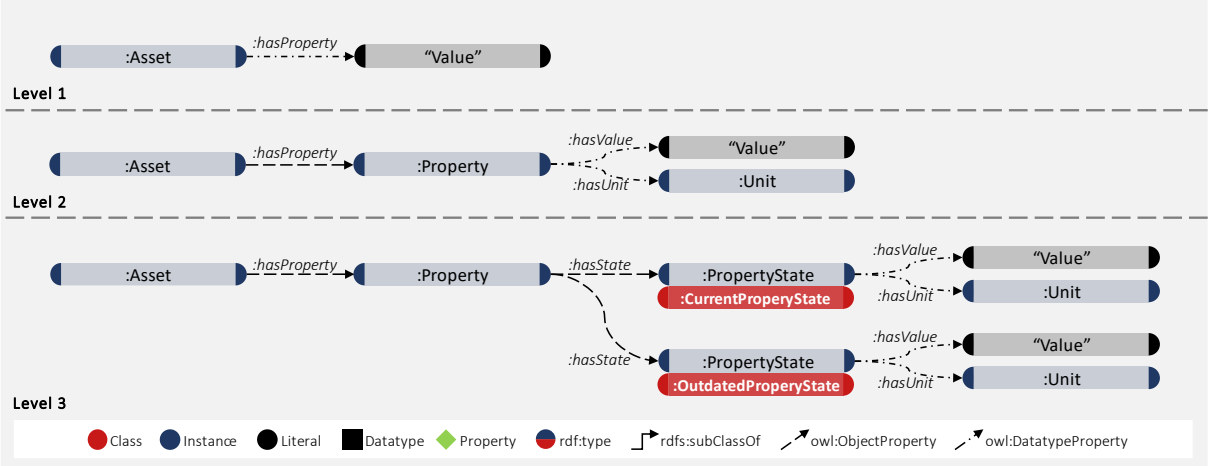


Figure 12: Property Levels

Property Naming

The proper naming of properties is an important aspect of creating ontologies. A clear and consistent naming convention ensures that the ontology is easily understandable and usable for both humans and machines. However, the position of the property name has not yet been standardised for each of the three property levels. Bonduel (2021) describes five possible approaches for modelling property names, each with its advantages and drawbacks. These approaches include using a hierarchy of (1) rdf:Property, (2) owl:AnnotationProperty and a combination of (3) owl:DatatypeProperty and owl:ObjectProperty. These approaches are possible for every level of property modelling, in contrast to the approach using (4) owl:Class or (5) skos:Concept, as these link to the intermediate node that only exists in level 2 property modelling or higher. The choice of approach depends on the specific requirements and practical application of the ontology and the reasoning engines used.

3.3 CONNECTED DATA

The use of linked data technologies in combination in BIM workflows has the potential to achieve a BIM maturity level 3. This would include a shift from file-based collaboration to data-centred collaboration using RDF, directly linking data instead of referring to other documents, see Figure 13.

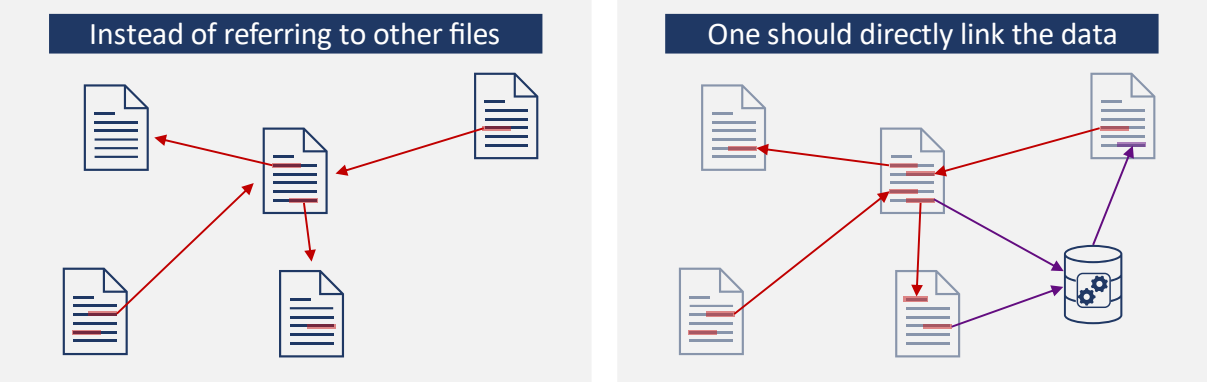


Figure 13: Towards a Web of Data

However, a complete transition to a workflow for structuring and visualising based on linked data would be an enormous challenge for the industry. Therefore, an intermediate step would be desirable, as is also done in the exchange of data with the ICDD standard (Hagedorn et al., 2023). This thesis proposes the use of linked data technologies in combination with BIM models and other files as a hybrid approach for structuring and visualising data, thereby enabling automation. This combination is defined as connected data. In other words, connected data is the combination of linked data technologies and files as a hybrid approach towards BIM maturity level 3, see Figure 14.

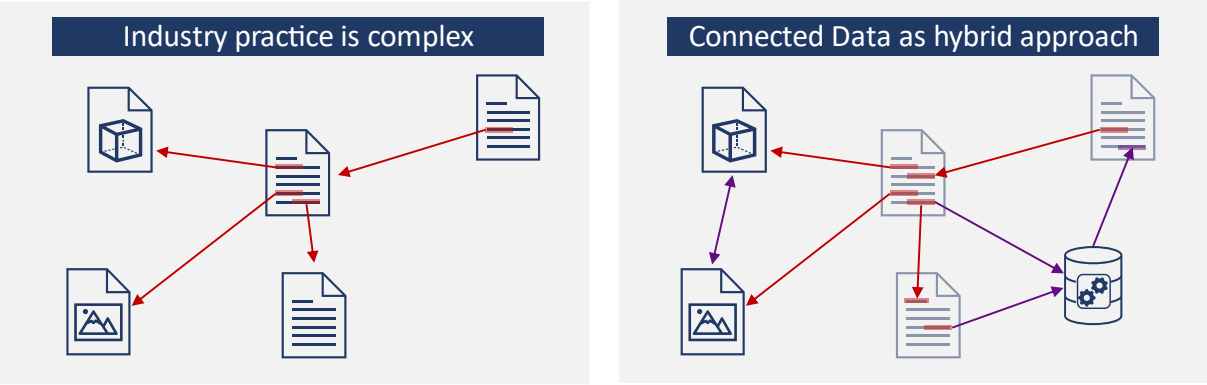


Figure 14: Connected Data as a step towards a Web of Data

3.4 Existing Ontologies for Natural Stone Restorations

This section highlights various ontologies and data structures that enable the representation, classification, and relationship modelling of historical buildings, their components, damages, inspections, interventions, geometry, and additional properties. These ontologies can be used as the basis for the data structure of a natural stone restoration project. All prefixes and corresponding namespaces can be found in Appendix I.

3.4.1 Building Registration

Objects in restoration are part of existing buildings in the real world, as there are also linked data developments in other domains there is already existing data on buildings. One example is the Dutch cadastre, which maintains a registry of all buildings and addresses (BAG) in the Netherlands. The BAG, together with other registries from the cadastre are published in a knowledge graph. Another relevant set of linked data is published by the Dutch Cultural Heritage Agency (RCE). They also made their registry of national monuments together with other cultural historical objects available in linked data.

The RCE has published their thesauri terms in linked data using SKOS (CHT), an ontology describing cultural heritage (CEO) and a full knowledge graph of cultural historical objects (CHO). Figure 15 shows some data that is available in these data structures, using the Dom Tower in Utrecht as an example. This building is classified as a national monument (*ceo:Rijksmonument*). Using the property about its legal status (*ceo:heeftJuridischeStatus*) a relation is made to a term from the thesaurus. This *skos:concept* has also defined it as a national monument. A similar construction is used to define the original function of the building (*ceo:heeftOorspronkelijkeFunctie*), however here an additional node is used. The bottom example shows a connection to other building registries (*ceo:heeftBasisregistratieRelatie*), which includes a link to a building (*bag:Verblijfsobject)* in the BAG in the knowledge graph of the cadastre.

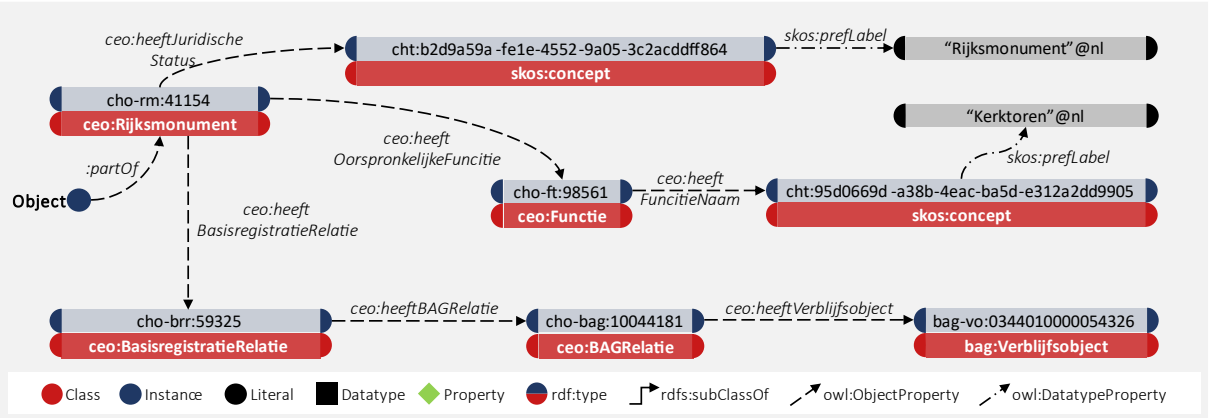


Figure 15: Dutch Cultural Heritage Agency Linked Data Structures: CHT, CHO, CEO

3.4.2 Object Topology

Several initiatives have aimed to develop web-based building data. To achieve this, several ontologies have been introduced to the Architecture, Engineering, and Construction (AEC) industry. One of the first initiatives was the creation of ifcOWL, which translated the IFC schema into an RDF-based schema using semantic web technologies (Beetz et al., 2009; Pauwels & Terkaj, 2016). However, as it mirrors the IFC EXPRESS schema, it inherits the same complex structure and size. Consequently, it consists of numerous classes and properties for describing building topology, categorising building components, geometry, and other building-related data. This then complicates understanding and usability of the ontology.

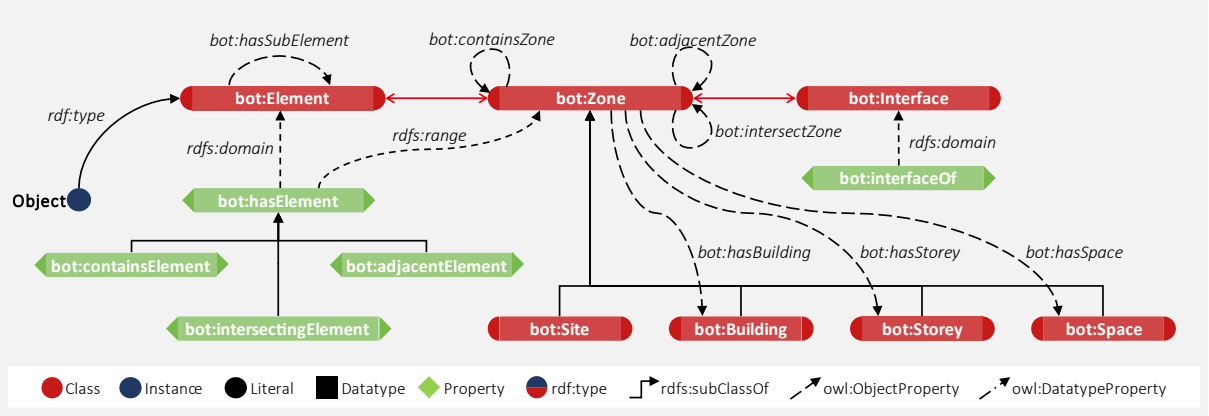


Figure 16: Building Topology Ontology (BOT)

The Linked Building Data (LBD) Community Group (CG) of the World Wide Web Consortium (W3C) was created with the goal to identify and align ontology development initiatives for building data. The approach they took was to create a set of interoperable, flexible, and open standards covering different aspects and domains (Rasmussen et al., 2021). In pursuit of these objectives, the LBD-CG developed the Building Topology Ontology (BOT), which is a flexible and lightweight ontology to define the relationship between elements within a building. Rasmussen et al. (2017) took the first step in creating the Building Topology Ontology (BOT) by introducing a basic ontology that describes building structures using the classes *bot:Building*, *bot:Storey*, *bot:Space*, and *bot:Element*. Over time, the BOT ontology has evolved, with the addition of multiple classes and object properties to the data model, which shows great promise in achieving BIM maturity level 3 with proper Linked Building Data (Rasmussen et al., 2021). The current version of BOT exists out of three main classes: *bot:Zone*, *bot:Element* and *bot:Interface*. Figure 16 shows the subclasses of *bot:Zone* and their relation to *bot:Element* using the three sub-properties of *bot:hasElement*. Instances of *bot:Interface* describe the relationship between some specific zones, elements or zones and elements in detail. W3C LBD-CG uses the BOT ontology as a core to develop new ontologies for the different domains and aspects of the built environment, following the interoperability and modularity of BOT.

Seeaed & Hamdan (2019) have developed the Stone Component Ontology (SCO), as an extension ontology of BOT, for semantically modelling natural stone facades. SCO, see Figure 17, introduces *sco:StoneComponent* as a subclass of *bot:Element* to classify the components in stone façades. Stones are created as instances of *sco:Stone*, while joints are classified using *sco:Joint*. The ontology also includes topological relations and simple geometrical data properties such as height, length and width (Hamdan, 2023).

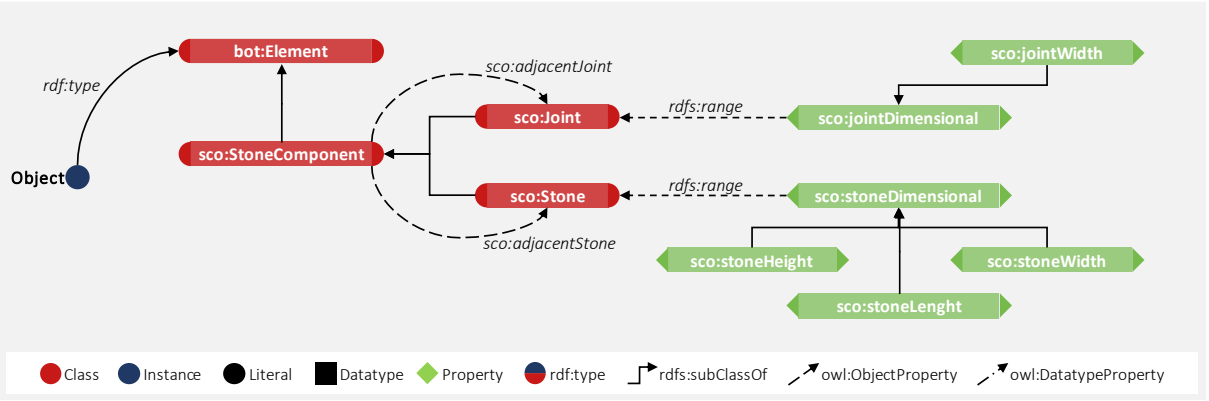


Figure 17: Stone Component Ontology (SCO)

3.4.3 Damage Topology

The Damage Topology Ontology (DOT) was developed using the principles of the LBD-CG, creating a core ontology for the representation of damages. DOT contains classes and properties that define topological relations between the damage and the affected building components (Hamdan et al., 2019). In DOT, the *dot:Damage* class is used to describe the concept of construction-related damages, see Figure 18. Damage can be modelled at different detail levels using *dot:DamageArea* and *dot:DamageElement*. The latter is used for specifying single damages at a high detail level, whereas the former allows for representing a collection of damages in an area at a lower detail level. The corresponding *dot:hasDamageElement* and *dot:hasDamageArea* properties can be used to link to building elements or zones. Adjacent damages, affecting multiple building elements can be grouped using the *dot:DamagePattern* class. Moreover, the cause of the damage can be specified using the *dot:Causation* class and linked to an instance of *dot:Damage*.

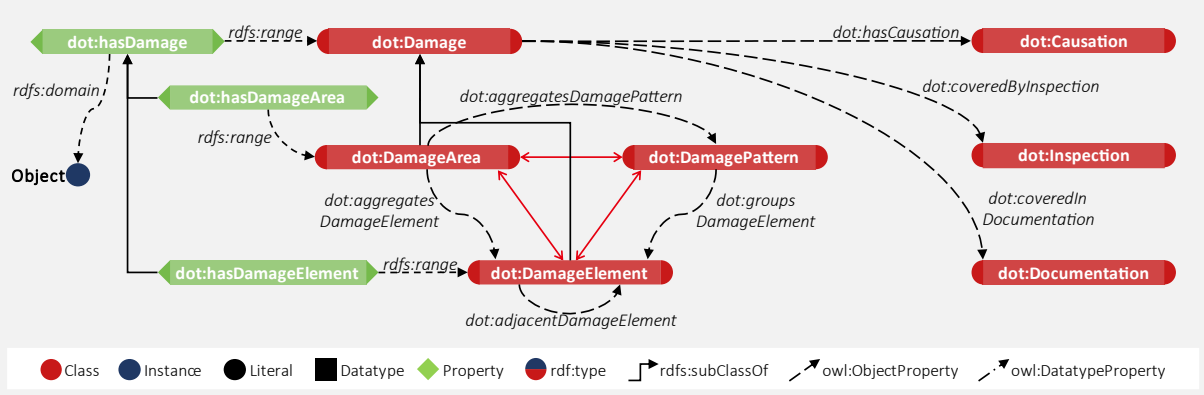


Figure 18: Damage Topology Ontology (DOT) 1/3

An important part of identifying damages is creating detailed documentation of the damage inspection. This introduces the *dot:Inspection* and *dot:Documentation*, they allow the linking of reports and various materials gathered during different inspections to individual instances of damages, see Figure 19. These documents, such as photos, sketches and inspection reports, can be classified as *dot:ExternalResource*. The data property *dot:filePath* can be used to connect to the URI of the file. Using the class *dot:Description* allows for the creation of literals describing the damage. This documentation can now be linked to the damage, the building element and the inspection. This inspection concept can have a specific person conducting the inspection, it also is used to describe other relevant information, it will therefore be explored further in the following section.

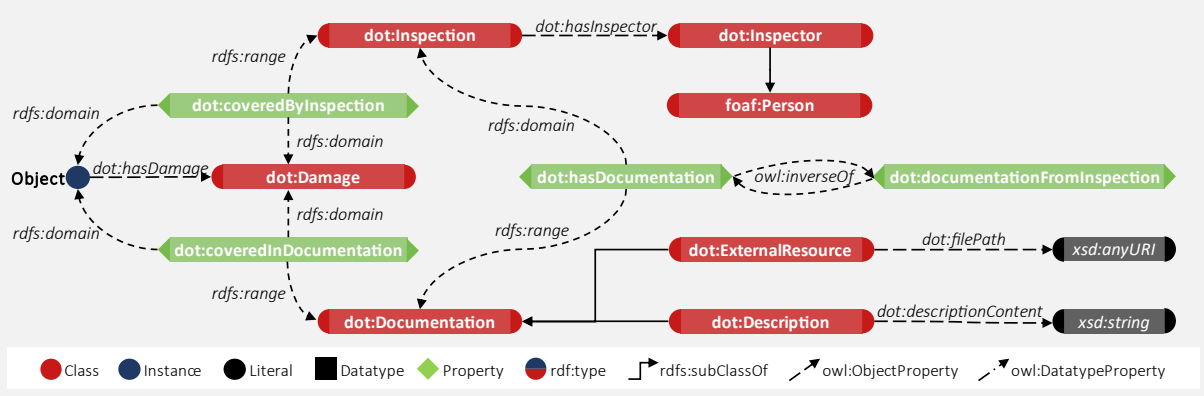


Figure 19: Damage Topology Ontology (DOT) 2/3

Based on the same principles as BOT, DOT also supports the creation of extensions that can describe damages in a specific domain. This could include adding other classes for classifying damages, such as more specific and additional properties that are relevant to these specific damages. Some relevant examples are the Natural Stone Damage Ontology (NSD) and the MDCS damage atlas Ontology (MDCS-O), see Figure 20. The NSD consists of a taxonomy of different types of natural stone damages based on the ICOMOS natural stone damage descriptions (Hamdan, 2023; Seeaed & Hamdan, 2019). The top class NSD, *nsd:StoneDamage* is a subclass of *dot:ClassifiedDamage* enabling domain-specific classification while using DOT. Bonduel (2021) used a similar approach to converting the MDCS damage atlas into an ontology that extends BOT in the same way. Another relevant type of available damage description is the thesauri terms the RCE has published in linked data using SKOS (CHT).

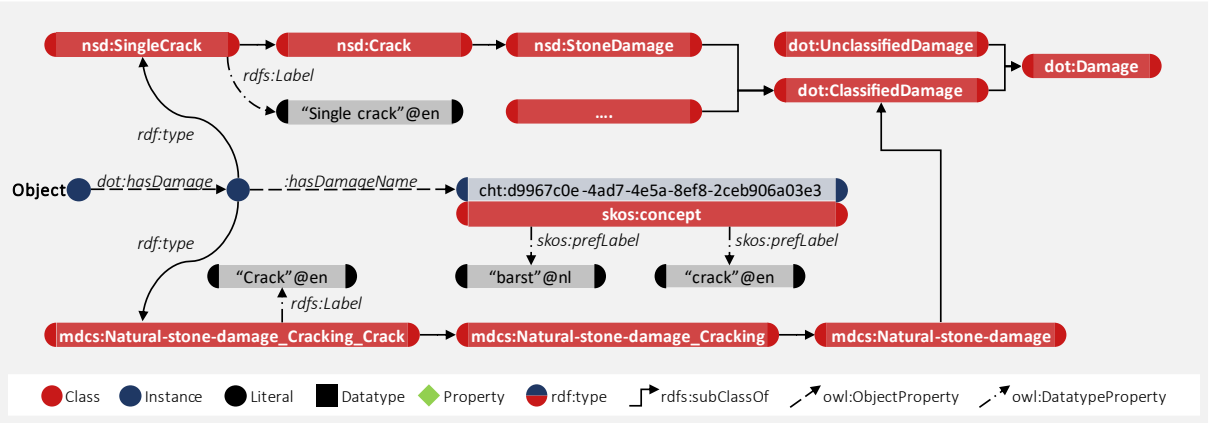


Figure 20: Damage Topology Ontology (DOT) 3/3

3.4.4 Inspection and Intervention Topology

Inspections and interventions are (a series of) tasks related to damages. Bonduel (2021) proposed the Construction Tasks Ontology (CTO) to provide terminology for construction tasks, see Figure 21. In an example a *dot:Inspection* is defined as a *cto:TaskContext*, providing context to a specific task. CTO defines five different types of tasks: instalment, removal, modification, repair and inspection. Each of these tasks can be related to either a *bot:Element*, a *bot:Zone* or a *dot:Damage*. CTO also introduces the concept of *cto:TaskContext*, which can be used to group tasks based on some common feature, such as being the result of the same inspection or being part of the same intervention. It also contains a *cto:TaskMethod* for adding some description to a task. The object property *cto:afterFinishedTask* can be used to add a sequence to a series of tasks.

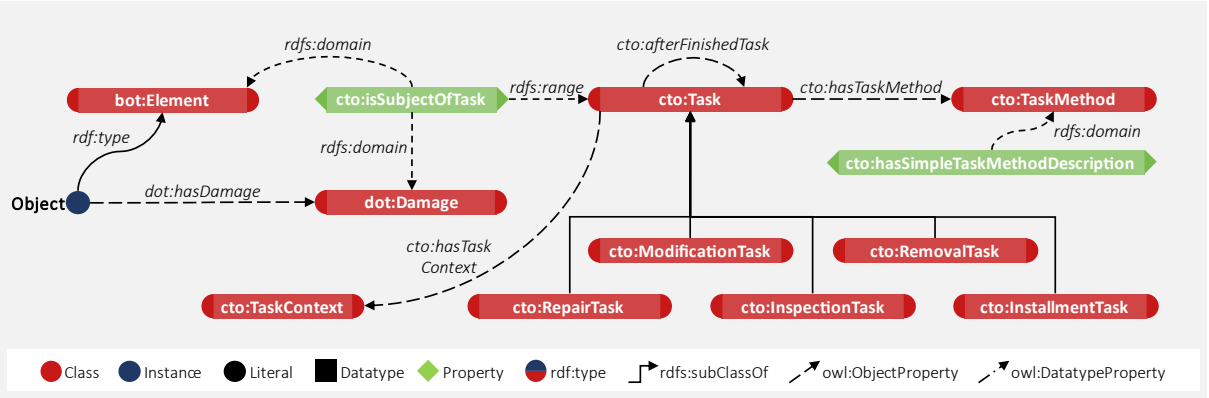


Figure 21: Construction Tasks Ontology (CTO)

Similar to BOT and DOT, extensions can be made on CTO. The Stone Renovation Measures Ontology (SRMO) extends *cto:RepairTask* into a taxonomy of different subclasses of *srmo:StoneRepairTask*, that on itself a *rdfs:subclass* if of *cto:RepairTask* (Hamdan et al., 2021). Similar to damages, the thesaurus of the RCE that is published in linked data using SKOS, also includes terms that define different types of interventions and tasks.

3.4.5 Object and Damage Geometry

Wagner et al. (2020) identify four distinct approaches for geometry description in a semantic web context. These are RDF-based geometry formats, JSON-based geometry as RDF graphs, RDF literals embedding non-RDF geometry files and RDF literals referencing external non-RDF geometry files. Various ontologies provide properties for connecting geometry descriptions to non-geometric data, including geometry ontologies, non-geometry ontologies, and geometry management ontologies. According to Pauwels et al. (2022), it is not efficient to directly transform these lists into knowledge graphs. Wagner et al. (2022) also describe the third idea of managing geometries as promising. This follows the fourth approach, where geometry is stored in its native format and semantic web technologies are used to link to the geometry. Two ontologies that can support this approach, the Ontology for Managing Geometry (OMG) and the File Ontology for Geometry formats (FOG), are explained below.

The Ontology for Managing Geometry (OMG) is an ontology that provides generic concepts for managing geometry descriptions in a semantic web context (Wagner et al., 2019). It focuses on general-purpose functionalities that allow the handling of single, multiple, and versioned geometry descriptions. OMG is inspired by the Ontology for Property Management (OPM), it thereby follows the same ontology design pattern as OPM (Pauwels et al., 2022). The ontology does not contain any information about a specific approach, but it serves as a generic way to describe single, multiple and versioned geometries. OMG can be extended by the File Ontology for Geometry formats (FOG), which extends the OMG to create file format, versioning and object identifier information for geometry data (Bonduel et al., 2019). See Figure 22 for an example of the structure of OMG and FOG using the property level 2 pattern.

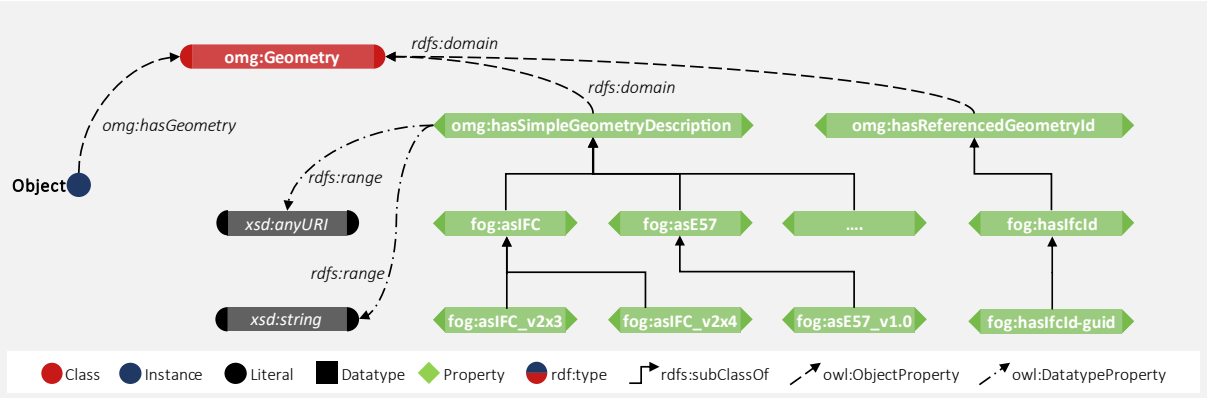


Figure 22: Ontology for Managing Geometry (OMG) with extension FOG

3.4.6 Additional Properties

Objects - Material

The cultural history thesaurus of the RCE is available in linked data using the structure of SKOS, this includes the materials described in the ERM-URL 4007. Hamdan (2023) created the Building Material Ontology (BMAT) to define building materials used in construction. Different building materials can be assigned to building elements using the object property *bmat:hasBuildingMaterial*. BMAT currently only exists out of a limited amount of materials with the superclass *bmat:BuildingMaterial*. Bonduel (2021) uses a similar approach using the Construction Properties taxonomy (CP) to assign materials to building elements (*cp:hasMaterialPart*). However, here two additional ontologies are created, the ConTax ontology (ConTax) and the AAT-MAT taxonomy (AAT-MAT). The ConTax defines the class *contax:Material* as superclass, while the AAT-MAT contains a huge set of subclasses to classify construction materials. The AAT-MAT was created by transforming the material section of the Getty Art & Architecture Thesaurus (AAT) from a SKOS structure to an RDFS structure.

Objects - Cultural Historical Value

The Built Heritage Properties (BHP) taxonomy is a classification system for built heritage properties that was developed by Bonduel (2021). The BHP taxonomy is used to describe and classify built heritage and includes several properties, such as building status, value, architect, function, style, etc. The *bhp:heritageValue* is a property that is used to describe the heritage value of a built heritage property. It can be used to describe the value of a property in terms of its association with significant events or people, its architectural style or design, or its historical context. BHP suggests the use of predefined property values such as existing concepts in thesauri. Bonduel (2021) suggests using concepts from the Flemish heritage organisation. In a Dutch context concepts in the RCE thesauri could be used.

Time

The PROV Ontology includes various properties, classes, and relationships that can be used to describe activities, entities, agents, and their relationships (Lebo et al., 2013). The *prov:generatedAtTime* property can be used to indicate the time at which an entity was created or generated. The *prov:startedAtTime* and *prov:endedAtTime* property is used to indicate the time at which an activity has started or ended. This information can be useful for understanding the context in which the dataset was created and how it has evolved over time.

3.5 DIGITISATION IN RESTORATION PROJECTS

As previously discussed, utilising Building Information Modelling and linked data technologies can aid in managing the difficulties that arise from the wide range of data formats and standards encountered in information documentation. This section discusses some important related concepts in a heritage conservation context.

In the literature on BIM for heritage projects, it is often referred to as HBIM, meaning Heritage or Historical BIM (Bonduel, 2021). Literature about BIM in historical building projects, mainly focuses on the geometry aspect of the models, such as Murphy et al. (2009) who refer to HBIM as the creation of a digital 3D model of a historical building, based on data from a laser scanner combined with digital cameras. This technology can be used to document the building's current state, plan restoration works, and monitor the progress of the project. Historic England takes a broader definition of HBIM, referring to it as the process of assembling intelligent objects into a virtual representation of a built asset (Antonopoulou & Bryan, 2017). This should include data about 2D and 3D geometry, created in a coordinated model, ensuring consistent output. It should however also include non-geometric information which refers to a wide range of parameters that describe the objects in the model. Finally, Historic England states that a HBIM should include the linking of data and external files to the model, by integrating other data in the model such as a point cloud or a link to an external document describing product specifications. Overall, working with HBIM requires a 3D geometry representation of the project, yet the level of detail of this geometry and all the other data depends on the use case. This research will contribute to the creating of this virtual representation but will refer to it as the connected object data.

Scan to BIM and photogrammetry are two technologies that are increasingly used in restoration projects to create accurate 3D models of existing buildings or structures (Rocha et al., 2020). Scan to BIM involves using laser scanning technology to capture detailed measurements of a structure, which are then used to create a 3D model of the space. This technology is particularly useful for restoration projects because it allows engineers to create accurate models of the existing geometry, which can be used to plan and design restoration work. Photogrammetry, on the other hand, involves using photos to create 3D models of a space. This technology is often used in conjunction with laser scanning to create even more detailed models (Alshawabkeh et al., 2021). Photogrammetry also allows to capture detailed information about the existing structure, including the condition of the building materials and any damage that may need to be repaired. Overall, both Scan to BIM and photogrammetry are valuable tools in restoration projects because they allow for the creation of accurate 3D models of existing structures. Creating a BIM model of an existing building is, in most cases, a very resource-intensive task that requires a lot of specific knowledge, which is why fully integrated models are not often used (Volk et al., 2014). This research will contribute to the value of creating these models, thereby improving the feasibility of using these resources.

The use of SWT in the sector is still limited in practice, however, more and more researchers investigate the opportunities of SWT. As no singular technology covers every use case, technologies such as BIM and SWT should be used in combination with each other. Some existing examples in research are using BIM and SWT to add additional information to a 3D model when creating a BIM model based on a point cloud (Werbrouck et al., 2020). Another example is the usage of SWT to describe information that does not fit in the BIM exchange schema IFC, for instance describing information about damages on objects while linking to the actual object in the IFC schema (Hamdan et al., 2019). The adoption of SWT for the description of heritage buildings could help in integrating the different data sources into a full virtual representation.

3.6 CONCLUSION

In conclusion, Building Information Modelling (BIM) has emerged as a significant collaborative process in the Architecture, Engineering, and Construction (AEC) industry, enabling the creation of digital representations of building structures that are data-rich and object-oriented. BIM models facilitate improved decision-making, design, construction, operation, and maintenance of structures while reducing information loss throughout different project phases. To advance BIM maturity and enhance collaboration, the industry requires standardised and vendor-neutral exchange formats, such as the Industry Foundation Classes (IFC). Restoration projects present unique challenges due to the need to respect the existing nature and historical significance of the building. Using BIM in historical building projects provides a promising framework for integrating historical data, architectural documentation, and digital modelling techniques to support the decision-making process in restoration projects. However, creating complete BIM models requires lots of resources and specific knowledge. Models can be created with increasing efficiency using techniques such as Scan-to-BIM and photogrammetry. Yet adding and linking all the information to the model during the process often falls short due to the amount of work this requires, creating multiple sources of data in different places.

The concept of connected data, including the use of semantic web technologies and linked data, holds great potential for optimising processes in the entire AEC industry. Linked data, based on semantic web technologies, facilitates the creation of knowledge graphs, and establishes relationships between data objects, providing context and enhancing the understanding of information. Through the combination of BIM principles, linked data and the specific requirements of heritage projects, data from different sources can be connected and structured according to the relevant standards in restoration projects. It can be seen as the next step to advance BIM maturity in the sector. It should enable professionals to make informed decisions, ensure the preservation of historical and cultural significance, and efficiently manage the complex challenges of restoring historical structures. The various ontologies discussed in this chapter can be used to structure all this data in a standardised way. The Building Topology Ontology (BOT) has been identified as the primary ontology used in other best practices and will be used as the primary ontology for the remainder of this thesis.

4 METHODOLOGY

This chapter details the methods used to design a prototype for structuring and visualising connected data. Furthermore, this chapter includes the requirements for the functionalities of the system and the resulting conceptual system architecture, which were defined in dialogue with industry experts.

4.1 RESEARCH APPROACH

The literature discussed in the previous sections provides an overview of the knowledge that lays the foundation for the creation of a system that can be used by heritage professionals to make better decisions. The aim of this research is to create a set of connected data of a restoration project and visualise this data in a way that is vendor independent, extendable and flexible. The choice has been made to focus on a specific type of restoration project, namely the restoration of natural stone facades of buildings. This allows the focus to be on a specific process and the information needed to support it.

The research uses a methodology that can be described as a system engineering approach, which uses a systematic approach to engineering a system. This involves reflecting on the outcomes at every iteration of the system design. These reflections are subsequently used to acquire further insights and enhance the upcoming iteration of the system under development. This method is visualised in Figure 23. The approach uses the following four steps: (1) defining the requirements, (2) designing a system architecture, (3) developing a prototype and (4) validating the prototype. In the first step, the process is analysed, and user requirements are defined using competency questions and use cases. A system architecture is developed that can help answer these questions by combining linked data with 3D models and other files. Based on this system architecture, a prototype of this system is developed using a specific case project. Finally, the system is validated against the requirements. The development and validation of different use cases is an iterative process where different functionalities are created and validated through three main iterations, where the desired competency questions are being answered and the prototype performance is being improved. Throughout the process, industry experts have been involved to ensure that the outcome addresses issues that are currently relevant in practice. This was done by using their input in defining the requirements, but also in the development and validation process to ensure that questions were being answered by the system in such a way that would add value to current practice.

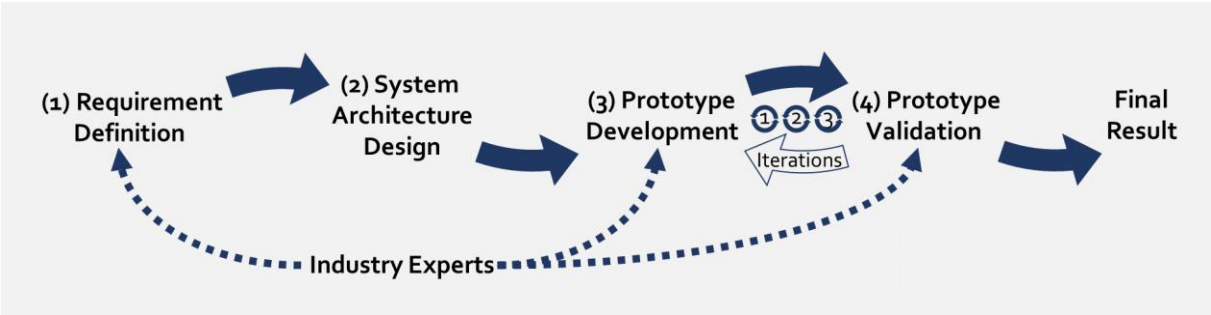


Figure 23: Research Methodology

4.2 REQUIREMENTS DEFINITION

During the research, various discussions were held with industry experts about the information problems stakeholders currently face. To translate these problems into functionality that should be available in the system, a three-step approach was used. First, the existing process on the site was analysed to define what steps are taken and what information is generated in the process. The resulting process schema was then used to discuss with stakeholders what information is needed during the different stages of a natural stone restoration. These information challenges were used to establish competency questions. Finally, these questions were translated into a number of different use cases that are used in the development of the system.

4.2.1 Process

The process of natural stone facade restoration consists of a series of tasks depending on the type of intervention that needs to be performed. To visualise this process, a BPMN schema was created, see Figure 24. As described in the literature, the ERM guidelines specify that information should be documented throughout the conservation cycle of a heritage property. The data column in the process schema shows where this data is collected throughout the process. This gives an insight into where information is generated, so that it can be associated with the parts of the process where that information is needed.

The schema does not show every possible workflow. However, it gives a clear overview of the main steps that are part of the process and the different types of workflows that are possible. Firstly, an inspection of each object in the building is carried out in two steps, the first is aimed mainly at gathering information and the second at deciding what intervention is needed. This information is then used to produce a cost estimation, including the type of replacement material. Once the interventions and costs are known, the actual work on the construction site begins. There are three types of workflows that are used in the execution of the interventions. Some interventions can be carried out in-situ, creating a simple process with only one task: the repair. However, it is often necessary to carry out an intervention in the masonry workshop, which results in a series of additional tasks. In this case, the object must be dismantled from the building and then transported to the masonry workshop. Once the object has been repaired or reconstructed, it can be transported back to the site where it is reassembled in the building. Dismantling an object will have the effect of requiring some of the surrounding objects to be dismantled as well. These are then transported to a location for temporary storage and, when the time is right, transported back to the site and used in a reassembly task. When all the work has been carried out and all the objects are back in place, a check is performed to ensure that the result meets the desired quality.

As mentioned, these are the standard steps and, in some cases, there are other tasks performed in the process. One example is when an object also requires the work of a sculptor, in which case the reconstruction task is followed by transport to the sculptor before being transported back to the site. Another example is when additional quality checks are carried out after repairs or reconstructions have been made to the masonry. Statuses are used to monitor progress during the process. After the inspection has been created, the object is given the status 'open', when the intervention on the construction site starts, it moves to 'in progress' and is set to 'ready' when the object is ready for inspection. Finally, when the object meets all the requirements, it is 'approved'.

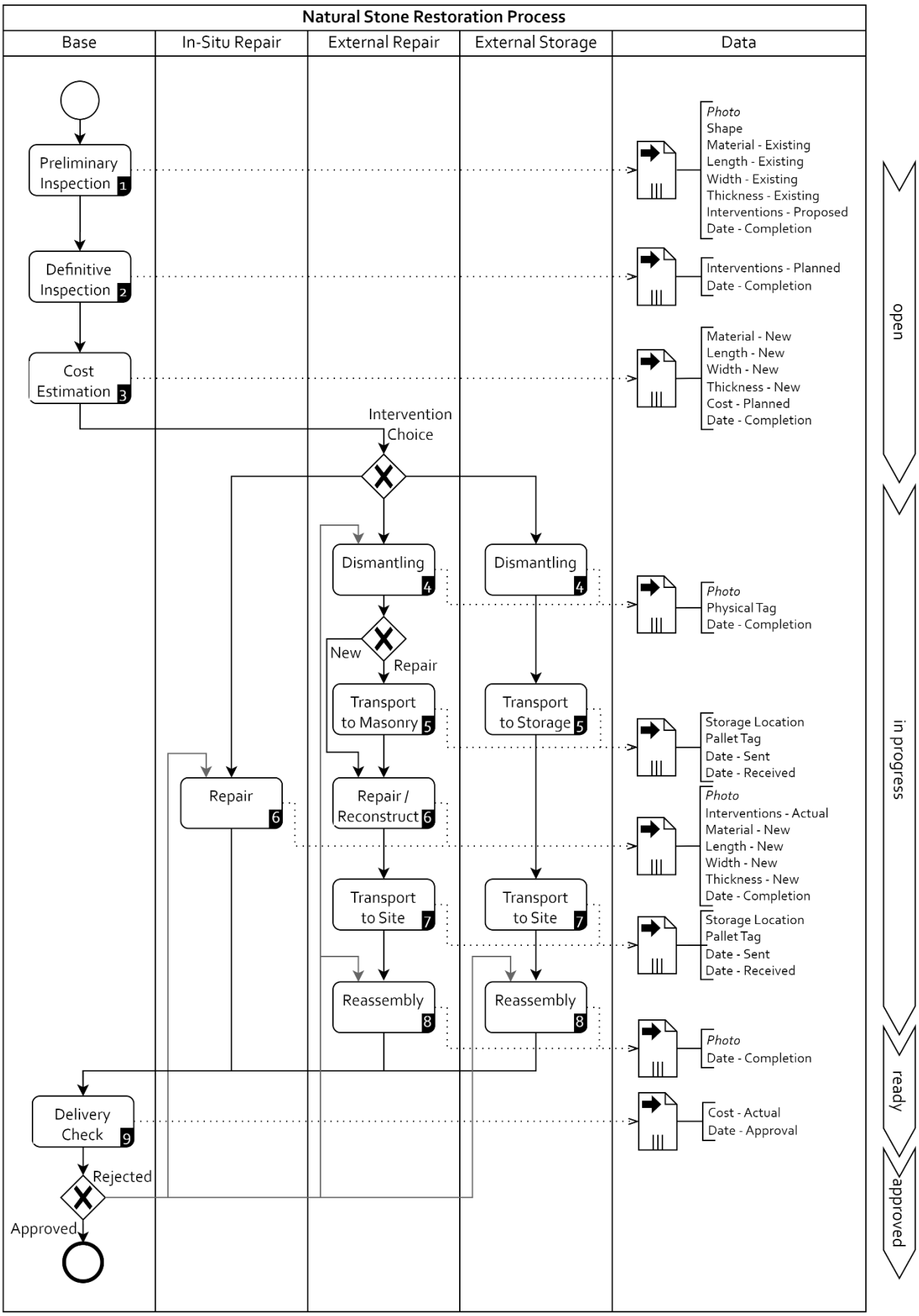


Figure 24: Process Schema of the Natural Stone Restoration Process

4.2.2 Competency Questions

The next step in creating requirements included discussions with different stakeholders about what information they needed throughout the process. The stakeholders included a project manager, a project planner, a cost expert, and an architect that all had a lot of experience in these kinds of projects. During a series of conversations, several competency questions were identified. Table 1 describes these questions and refers to the corresponding steps in the process illustrated in Figure 24.

Table 1: Current Competency Questions

CQ 1	**What is the location of the object in the building?** *Step 1-9*	Throughout the process, all stakeholders should be able to see where an object is in the building. For example, during cost estimation, it is relevant to have insight in the relation between the object and surrounding objects. Another example is during tasks on the construction site, here there is a need to quickly find an object to execute the desired task.
CQ 2	**What photos are available of a specific object?** *Step 2, 3 & 9*	During the process photos are taken to ensure complete project documentation. These photos of the inspection are used to create an accurate cost estimation, this includes the checking of the material and shape of the object.
CQ 3	**What is the volume, material, and shape of the object?** *Step 3 & 6*	During cost estimation, these aspects influence the cost of the intervention. If an object needs the be replaced, the dimensions of an object are needed to determine the volume of the new required material.
CQ 4	**What interventions are conducted on an object?** *Step 3, 6 & 9*	During an intervention task, there is a need to be able to get an overview of all the interventions that need to be conducted on a specific object.
CQ 5	**How to give an overview of all relevant information on an object?** *Step 3, 6 & 9*	Traditionally all the questions above are answered using different spreadsheets. However, when connection data opportunities arise for creating a better overview of this information, by combining it with other functionalities. For example, linking to photos or the object in the model.
CQ 6	**What is the height of the object relative to ground level?** *Step 3, 4, 6 & 8*	The height of the object relative to the ground level influences the cost of doing an intervention, mostly depending on the vertical transport facilities that will be required to reach the object.
CQ 7	**What is the progress status of every object in the building?** *Step 1-9*	For most of the parties involved in the project, there is a need for an overview of the progress of the project. Traditionally this is done by manually keeping and changing the status of each object and using different colours on one 2D drawing on the building site to keep track of the progress of each object.
CQ 8	**How to check if all objects include the required information?** *Step 3 & 9*	Finally, there is a need within projects to validate whether all data that should be collected on an object is recorded. As certain data needs to be available to make effective decisions.

In order to validate the competency questions, a specific object that reflects the existing process in a case project should be discussed with industry experts. Using this example object, industry experts can reflect on the way the competency questions are being answered throughout the process.

4.2.3 Use Cases

As a final step before creating a system architecture, the various competency questions were translated into use cases to be implemented as functionalities in the system. These functionalities are developed during the different iterations of the prototype development. To provide a clear objective for each iteration, a three-step approach was adopted, where in each step a set of functionalities is developed for the respective use cases, see Figure 25. Each use case either supports another use case or provides an answer to a competency question from Table 1. Multiple types of users are considered in the use cases being developed. The use cases for the beginner user focus on recognisable output, extended with new possibilities that introduce the user to the system. A more advanced user is provided with functionalities to find traditional information through the system. Finally, a pro user is also provided with the ability to create queries based on their extensive knowledge of the data structure. The following iterations are used:

- **Iteration 1 | Combining the Model with Other Data:** The first iteration focuses on the creation of a model viewer where a 3D model can be viewed. Within this model, objects should be selectable. Based on the selected object or user input, one should be able to find information about the properties, interventions, and photos of objects.
- **Iteration 2 | Extending Model Viewer Functions:** The second iteration focuses on extending the functionality of the model viewer to allow users to find objects in the model based on the unique identifier (GUID) of that element. In addition, functionality will be added to facilitate the possibility of taking measurements in the model.
- **Iteration 3 | Connecting to Existing Workflows:** In the final iteration of prototype development, the more traditional workflow is combined with the developed system. Creating a traditional spreadsheet export from the database with enhanced functionality, such as links directly to the model viewer and the data within it.

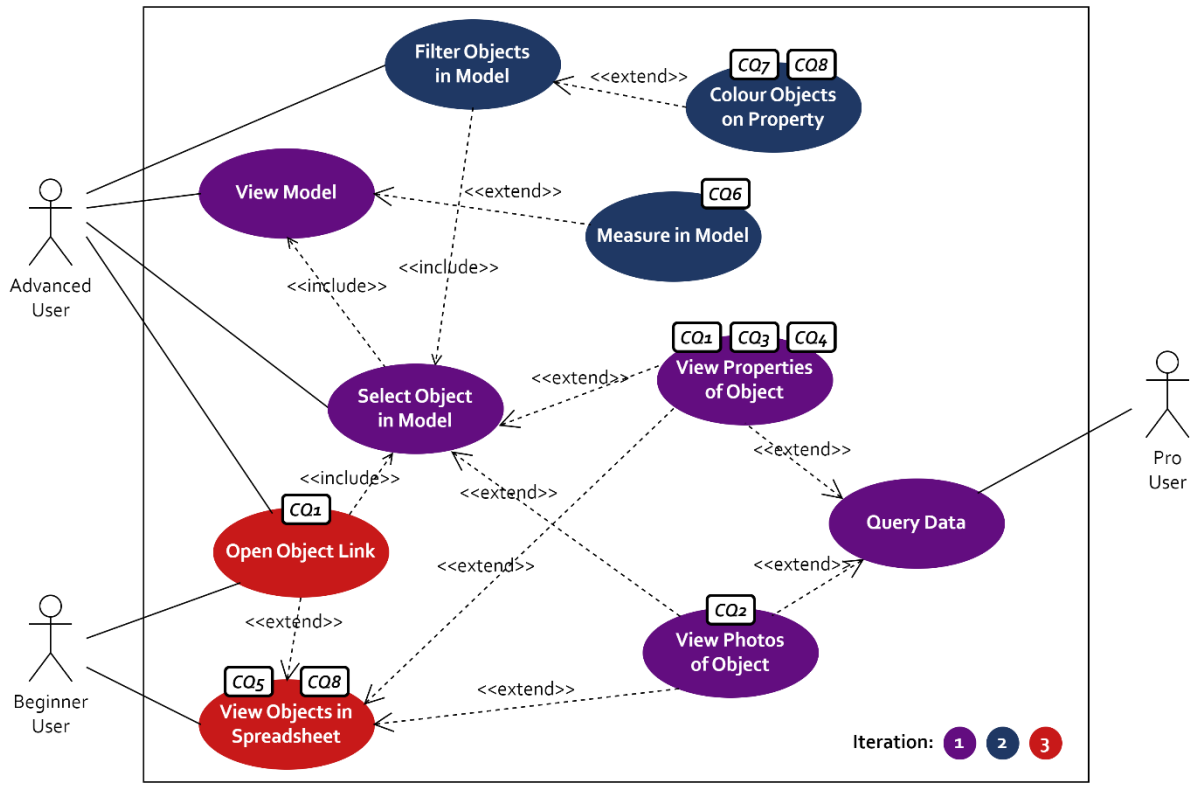


Figure 25: Use Case Diagram of proposed System

4.3 System Architecture Design

Before starting the prototype development, a conceptual system architecture was established, see Figure 26. It was inspired by best practice from other research. The architecture defines the components independently of the selected solutions, focusing on the responsibilities of these different layers in the system. Implementation of the system architecture must include considerations for communicating between different layers, currently indicated by the arrows in Figure 26.

Table 2: Layers of the proposed System Architecture

Resources - Data	Data is distributed across several different sources. This system component contains all the data that needs to be structured and connected. These sources include both documents and integrations with existing systems.
Resources - Structure	In addition to all the sources of data, there is another very important source, namely the structure of the data, which allows for the creation of structured and connected data. This should include a standardised and modular approach for connecting different information domains.
ETL	The ETL (Extract-Transform-Load) layer focuses on extracting, transforming, and loading data into the database. In this layer, all data sources are processed and transformed into structured and connected data that is then loaded into a database.
Connected Data	This component contains the databases that collectively comprise the connected data structure. This research proposes the use of a knowledge graph created with linked data to connect data and files such as a 3D model.
Services	The services component establishes the connection between the databases and the user interfaces, providing capabilities to ask questions about the connected data.
User Interface	Finally, the user interface provides a way for users to interact with the data through various functionalities implemented based on the described use cases.

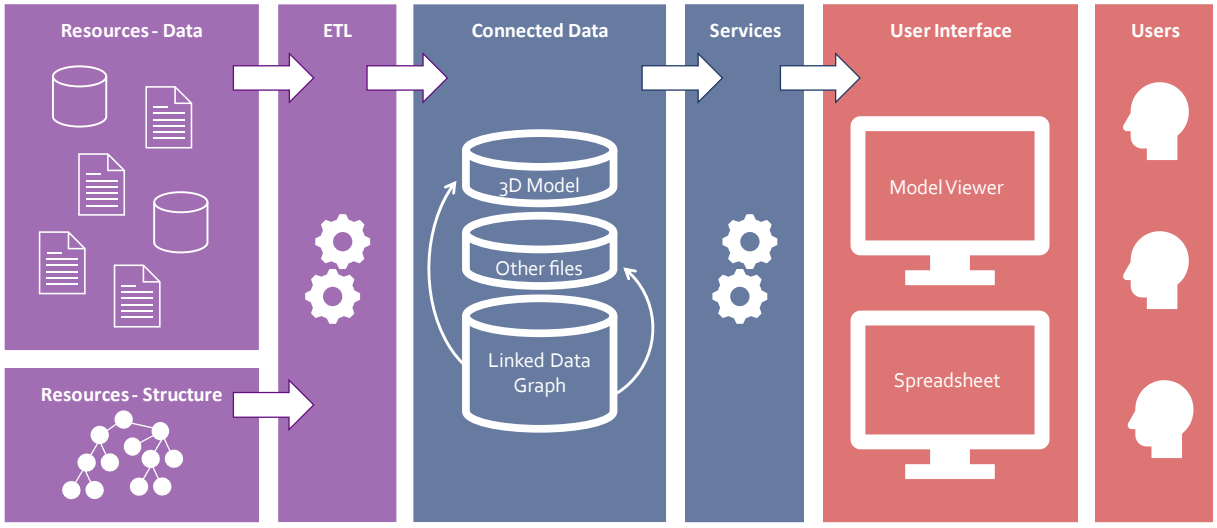


Figure 26: Proposed conceptual System Architecture

5 PROTOTYPE DEVELOPMENT

Having defined a conceptual system architecture, a prototype implementation of the system has been developed using a case project. This chapter outlines the different stages of the prototype development process and demonstrates the various functions that have been created within the visualisation tool, the Heritage LBDviz. However, the chapter begins with a description of the case project that has been used.

5.1 CASE PROJECT

For the proposed research, it is essential to have datasets that can be used to develop and validate a prototype of the proposed solutions. The restoration of the National Monument on Dam Square was chosen as a case project, see Figure 28. In this project, the restoration company Nico de Bont was responsible for dismantling and rebuilding the monument during a restoration in 2022. The National Monument in Amsterdam is a monument to the commemoration of World War II in the Netherlands. The monument is the centre of the annual National Remembrance Day ceremony on 4 May at 8 pm, when two minutes of silence are observed throughout the country in memory of Dutch war victims. This ceremony at Dam Square is traditionally attended by the Head of State and various representatives of the Council of Ministers of the Kingdom of the Netherlands and the Dutch Parliament.

The monument consists of a pylon with statues and a memorial wall with reliefs. In front of the monument, two lion figures stand as sentinels. The twenty-two-metre-high pylon, that is made of concrete, the memorial wall and the lions are cladded in travertine, a marble-like porous limestone from Tuscany, Italy. The last restoration took place in 1997. In 2022, 25 years later, the National Monument needed to be repaired or restored in some places. During the restoration, the travertine elements were dismantled and transported to the stonemasonry workshop, see Figure 27, where they were cleaned, repaired, and replaced where necessary, minimising disruption to the Dam Square.

The dataset provided does not include all the data described in the process schema, but the project offers a realistic representation of the current way in which the company operates and the data it collects. The following data sources were provided by the company during the research: a Revit model and PDF elevations to give an insight into the geometry of the building objects; the Autodesk Build system used to carry out the inspection, which contains data on object properties and photographs; the actual set of photographs taken during the process; and finally some data documented in several spreadsheets, such as the interventions carried out on the elements and the dimensions of all the objects.

Figure 27: Restoration of the National Monument

Figure 28: National Monument on Dam Square

5.2 System Architecture Implementation

Figure 29 shows the implementation of the system architecture used in the prototype development, focusing only on the data available in the case project. The system architecture contains the same layers as the conceptual design, except for the resources, which are grouped together in a single layer. This layer contains both the data sources and the data structure in the form of ontologies. In addition, a semi-transparent display at the bottom shows how the most important input data was created with Autodesk Build and extracted from it.

The following sections explain the implementation of the different layers in three parts. The first two focus on the preparation of the data that is then used to create a RDF graph, which can also be described as the data pipeline of the system. The third part focuses on the visualisation of the data to implement the use cases defined earlier. The whole gives an impression of how the proposed system architecture can be implemented in practice.

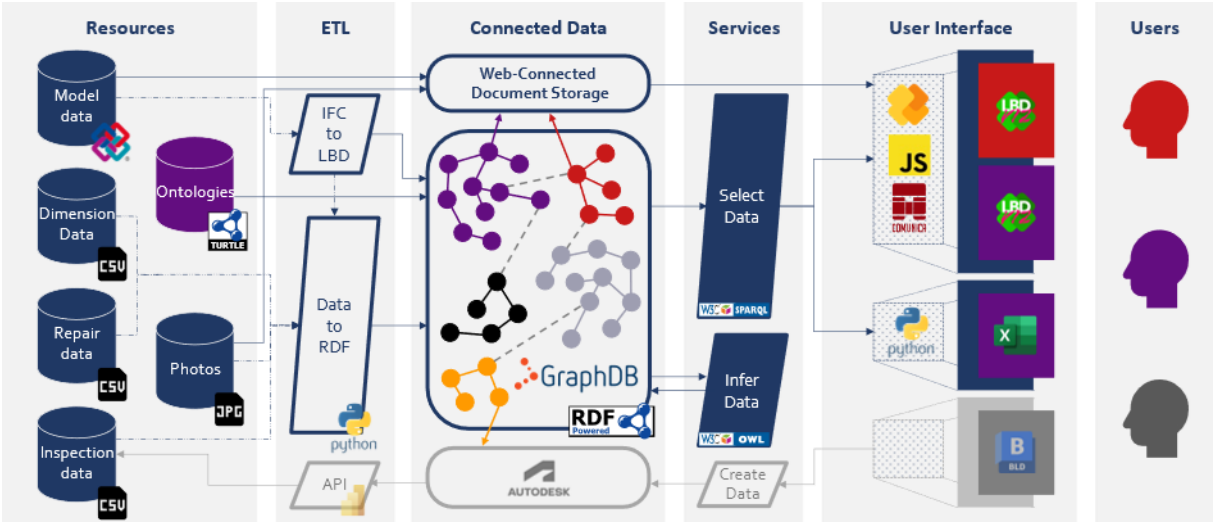


Figure 29: System Architecture of Prototype

5.3 Data Preparation

As current sources were created using the traditional process, many sources require some preparation before they can be used as input data.

5.3.1 Resources | Data

One of the essential parts of showing the use cases is the creation of the data graph. As explained before, in the current process data is spread out over multiple sources. For the proposed use cases, three main types of data are used, an IFC file that describes the location of the different objects in the building, the photos of all the objects that were made in the field, a set of CSV files that contain all kinds of data about the objects, their properties, damages, related tasks, and interventions.

Model data – Revit / SketchUp

During the execution of the project, a Revit model was made by the architect based on a point cloud of the monument. This model was mainly used to generate 2d drawings as output. The labelling of the specific objects was done on the 2d sheets, meaning that the IFC export from Revit only generated solid geometries that included multiple objects. So, to get a model that was useful for the desired use case some adjustments had to be made.

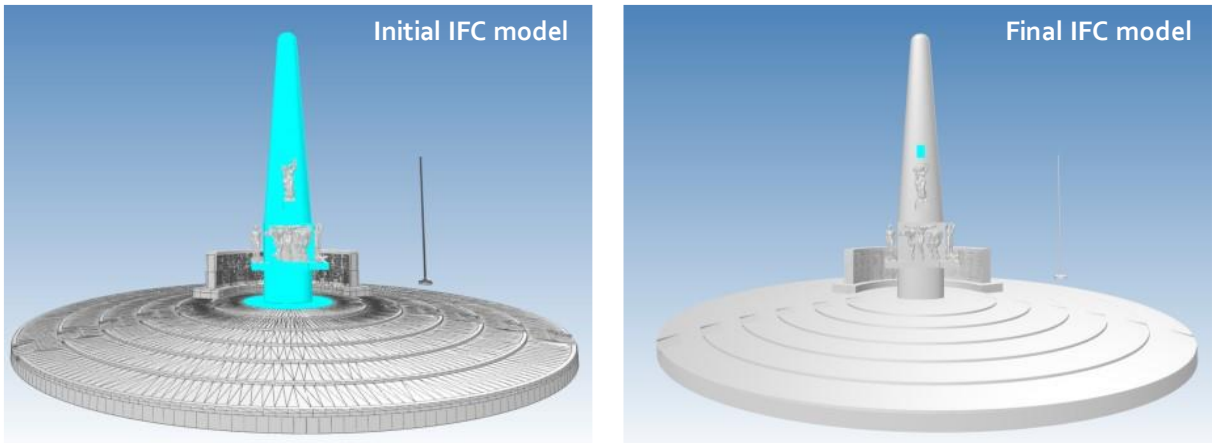


Figure 30: Difference between IFC Models before and after editing in SketchUp

The Revit model was loaded into Sketchup, where the model was edited to make sure every object could be identified by an individual (set of) surface(s). This was done by cutting the solid objects, that were created in Revit, into new surfaces. The geometry of these surfaces was based on the information that could be derived from the original Revit model and the 2D output that was used in the project. These surfaces were then combined into an IfcElement and given a name using the IFC Manager add-on for SketchUp (Brouwer, 2022). This model was then exported to a new IFC model using the exporter function of the same add-on. The reason for using this exporter instead of the one part of SketchUp itself is that the standard exporter is very limited in its functions and output. The resulting IFC model now included an individual IfcElement for every natural stone object that is part of the monument, see Figure 30.

CSV data – Autodesk Build

As visualised in the bottom part of the system architecture, inspection data is currently created in Autodesk Build. This data is stored on Autodesk servers, limiting the availability of this data to the capabilities of this proprietary software. To still be able to export desired data from the platform, the Power BI connector was used. This allowed to explore the different tables of created data that were available using Microsoft Power BI. Power BI was subsequently used to export the desired data to a CSV file. The consideration was made not to transform the data in Power BI but use the tables that are provided in the data structure used by Autodesk. The main reason for this is that this should make it easier to automate this process in the future, as the Autodesk Construction Cloud Platform APIs are based on the same data structure.

The Power BI connector has been used to extract two tables, namely the *'issues_issues'* and the *'issues_attachments'*. The first one contains information on the 'issue' that was created, this contains data on every issue that was created for every object in the building, most importantly the labels that were given to every object and the corresponding unique ID it has in the Autodesk database. The second describes the relationship between photos and the respective tasks that were recorded within the issue. This furthermore includes a similar unique ID for every photo and the file name of the photo. As in the current data structure only one 'issue' is created for every object, no information is stored within an issue about the progress of tasks on the object, other than giving the object as a hole a status. This however means that there is no detailed information about what photo was made in which task.

Photos – Autodesk Build

As part of the inspections that are done through Autodesk Build, photos are made. Extracting these photos from the Autodesk servers has historically been quite a challenge. Because of this reason, during the delivery of the 'Dam' project, an effort has been made to download the photos one by one and store them as local files. Automating this process in the future could be done by directly using the API provided by Autodesk. However, for the purpose of this prototype all the photos are available in a local folder structure.

CSV data – Excel

Some of the desired data was currently only available in several Excel files. This data was transformed into a CSV, assuming the process of the desired workflow where the data is collected in a more systematic approach. This method was used for a file that contained the dimensions of every object, one that stated every intervention that was conducted on every object in the structure and one containing the object type and material.

CSV data – Python

As explained before, the photos could not be linked to a specific task based on the information in Autodesk Build. To make sure there was data that could be used, a Python script was created using the folder with all the photos as input. This script was used to loop over all files and subfolders in the folders. For every file, if it was a photo, data was captured on the name of the photo, when the photo was created and what object was displayed on the photo according to the folder structure. Finally, a final iteration was done over the results to calculate the task in which the photo was most likely taken. This was done by comparing the dates the photos were taken for every object. Assuming that the first photo that was taken as part of the inspection and including all that were taken within 10 days as well. As the intervention were done by a sub-contractor these were updated manually to the platform and hence did not contain metadata on when they were created. Finally, adding all the other photos to the reassembly task. As the project only included a limited number of tasks where photos were made, the resulting dataset was quite reliable, see Listing 1. This is however far from ideal, as the desired process should use tools that already produce this data.

```
Folder Name,File Name,ID,DateTime,Task
W4.1,20220516_072319.jpg,29488c56-f9b8-46dd-a5a3-7044ca0d77ee,2022-05-16T07:23:19,Opname
W4.1,20220516_072320.jpg,acf5c7bf-ae82-4f31-af5d-2b3b115cbf98,2022-05-16T07:23:20,Opname
W4.10,20220516_072338.jpg,13cb6b3e-a757-40c5-b5c7-d8da0c299a25,2022-05-16T07:23:39,Opname
```

Listing 1: Fragment of the CSV File of the Photos

Sources → ETL

The communication between the sources and ETL layers in the system architecture can be explained quite simply, as all the resulting data is file-based, these files are just used as input for the processes in the ETL layer. However, as processes get more structured and standardised, this step could be automated to a great extent, as the sources could then directly be used as input for the ETL layer.

5.3.2 Resources | Data Structure

As described in the literature, and based on best practices, it was decided to use a standardised and modular approach to connect different information domains. Several ontologies were used to achieve this. Most of these ontologies were identified during the literature review of best practices, where the Building Topology Ontology (BOT) was identified as the primary ontology used. Based on all the available data, a mapping was done to find all the classes for the case project data in the BOT ontology and all its extensions, such as CTO, DOT and OMG. Figure 31 shows all the ontologies used, the domain for which they are used and their main relationship with the other ontologies. However, some of the relationships present in the resource data were not available in these existing ontologies. The solution created, the RII ontology, is explained below. All the ontologies themselves are stored in a Turtle serialisation format, these files are then used as input for the other layers.

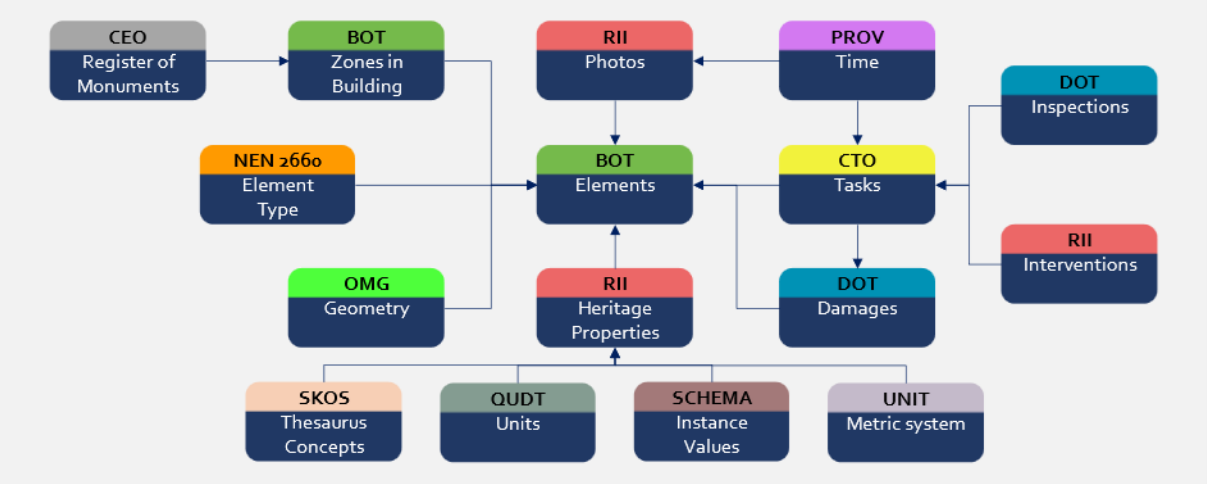


Figure 31: Overview of used Ontologies

Restoration Intervention Information (RII) ontology

As mentioned above, the Restoration Intervention Information (RII) ontology was created to define a set of classes and relationships that were not directly available in the other BOT related ontologies identified. As a solution, the RII ontology was created in Protégé. The ontology focuses on the interventions that are carried out on the heritage assets and the different element properties that are used as information in this process.

First, the ontology describes classes of interventions based on the conservation ladder, defining the class *rii:Intervention* and its subclasses *rii:Preservation*, *rii:Repair*, *rii:Reconstruction*, see Figure 32. These interventions are then linked to the *cto:InspectionTask* for which they were selected and the *dot:Damage* they are intended to cover. Finally, the *cto:hasTaskContext* is used to group the various *cto:RepairTask* instances that are performed to carry out the *rii:Intervention*.

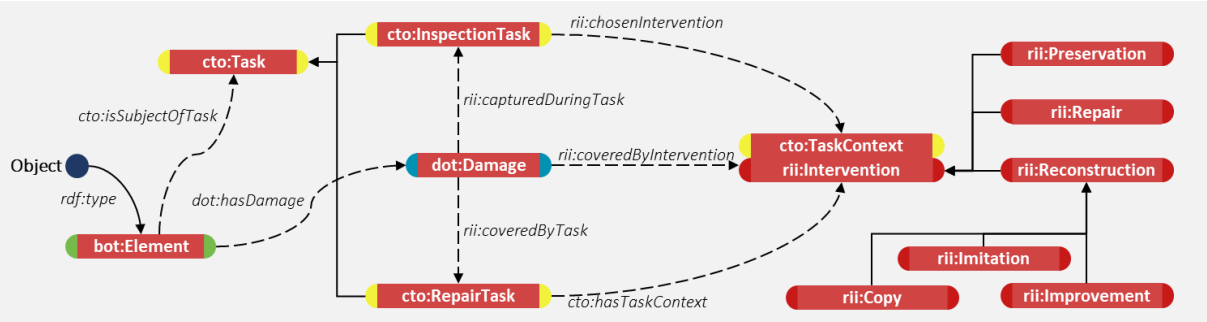


Figure 32: Restoration Intervention Information (RII) ontology - Interventions

In addition to interventions, the ontology is also used to add external resources to the graph, specifically photos. The DOT ontology already defines a *dot:ExternalResource*, but this is specifically described as documentation that should only be damage related. As not all information is damage related, it was decided to use the class *rii:ExternalResource* with the subclass *rii:ExternalImage*, which is used in combination with the data property *rii:filePath* to link to externally stored images, see Figure 33. The ontology also includes an OWL property chain based on *cto:isSubjectOfTask* and *rii:capturedDuringTaks*, creating the relation *rii:hasDescriptionInResource* with the inverse *rii:describesElement*.

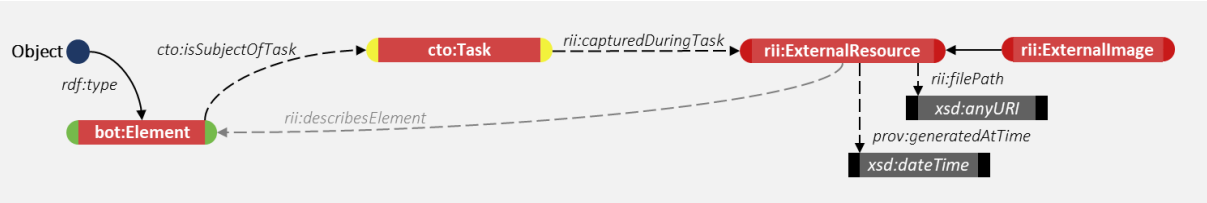


Figure 33: Restoration Intervention Information (RII) ontology - Images

Finally, the ontology was also used to define element properties such as length and material. These and most of the other properties can be found in some other ontologies. However, property levels and naming are often handled in different ways, so the relevant properties were added to the RII ontology to ensure that there was a consistent ontology design pattern in the graph, making querying more manageable. Figure 34 shows the chosen approach, which combines a property level 2 approach with a combination of property naming approaches 3 and 5 described in section 3.2.4. This ensures that user-defined instance values can be used, but also that thesaurus concepts can be linked via a *skos:Concept*. Each property is a subclass of the superclass *rii:ElementProperty*. Using the same approach, all relations between the element and the properties, such as *rii:hasWidth* and *rii:hasMaterial*, are subclasses of *rii:hasElementProperty*, ensuring that properties can be queried easily. Versioning of different properties can be done using the PROV ontology, enabling queries to be filtered by the most recent date. Similar to the photos, a property chain has been created to connect the element directly to subclasses of the dimension properties.

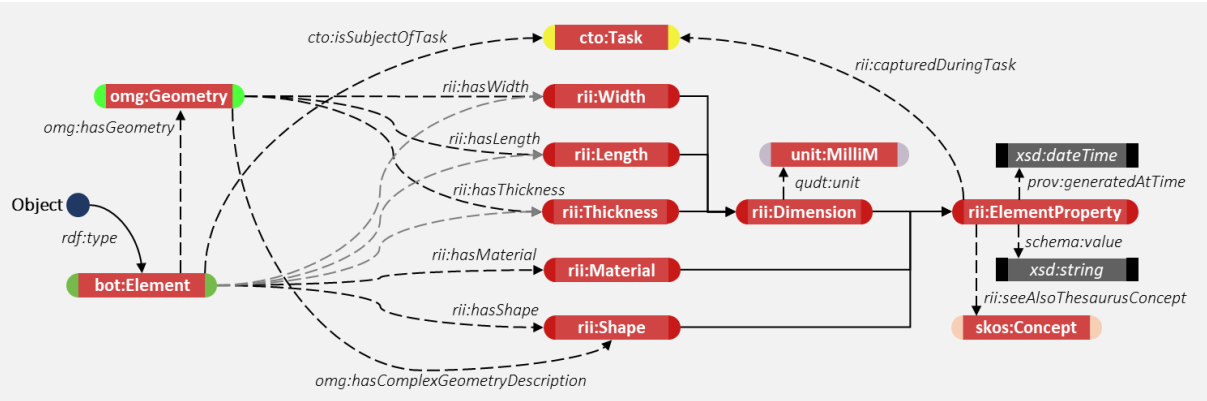


Figure 34: Restoration Intervention Information (RII) ontology - Properties

The current ontology focuses only on the needs of the case project and may not include all the information needed in other projects. However, the approaches that are used can be scaled, especially for the way properties are described.

5.4 GRAPH CREATION

5.4.1 ETL | Transforming IFC to RDF

An existing IFCtoLBD converter was utilised for the creation of RDF data from the IFC file (Oraskari et al., 2023). The conversion process converts the building data from its original format in STEP Physical File to Turtle, a serialisation of the RDF data model. As part of the conversion process, the geometrical data is excluded from the other data, resulting in a Turtle file that specifically describes objects, their classes, properties, and the corresponding property values using existing ontologies such as the BOT ontology. This resulted in a Turtle file that describes the buildings, stories and objects that are on the site, and the relationships between them. As can be seen in a sippet of the result in Listing 2, the resulting data only includes minimal information, the type, globalID and name of every object.

```
inst:site_d06994b3-1585-4417-9e47-4b620eef7fff
        bot:hasBuilding  inst:building_66679340-b5a3-4581-8b49-cf11ebb7bcd4 ;
        rdf:type         bot:Site ;
        props:nameIfcRoot_attribute_simple  "Dam" .

inst:building_66679340-b5a3-4581-8b49-cf11ebb7bcd4
        bot:hasStorey  inst:storey_0af8e765-ec68-45d9-b164-712eff91365b ;
        props:nameIfcRoot_attribute_simple  "Pyloon" ;
        rdf:type       bot:Building .

inst:storey_0af8e765-ec68-45d9-b164-712eff91365b
        bot:containsElement  inst:buildingelement_b2a3a850-0ad0-4c9d-ba00-b9dd3f692581 ;
        props:nameIfcRoot_attribute_simple  "13" ;
        rdf:type             bot:Storey .

inst:buildingelement_b2a3a850-0ad0-4c9d-ba00-b9dd3f692581
        props:globalIdIfcRoot_attribute_simple  "2oewXG2j1CdRe0kTq$QIM1" ;
        props:nameIfcRoot_attribute_simple  "P1.182" ;
        rdf:type                bot:Element .
```

Listing 2: Fragment of the Turtle File from the IFCtoLBD Converter

5.4.2 ETL | Transforming CSV to RDF

This paragraph focuses on the process of transforming a CSV file into RDF using Python. The process involves two main steps: importing the CSV file into Python and creating an RDF graph based on the data. Furthermore, additional methods such as SPARQL Insert queries and SPARQL Select queries are also mentioned, highlighting their usage for inserting data and retrieving additional input data from the graph, respectively.

Create RDF Graph from CSV

The first step of the script is to import the respective CSV file into Python, see Listing 3. This is done by using the CSV module. The script opens the CSV file, skips the first line of the file as this includes the headers and then appends every other row of data to a list.

The next step is to create RDF data based on this list of data. This is done by using the 'rdflib' module. This is a Python library for working with RDF. The library contains a Graph class that represents an RDF graph. It also contains parsers and serialisers for almost all the known RDF serialisations, such as RDF/XML, Turtle, N-Triples, JSON-LD, and provides functions for declaring graph namespaces. The code creates an RDF graph using the library and imports namespaces from the library and the _Namespaces file that was created to define several other namespaces. It then iterates through the list and collects specific values. For each element, it adds RDF triples to the graph with information such as its type, label, and relationship with other entities in the graph. Statements are made using

the predefined namespaces, the literal class and also full URIs using the URIRef class. Finally, it serialises the graph into a Turtle file and also creates a serialised version in N-Triples format that will be used later in the process.

In some cases, the creation of new triples is done in a more complex way using several conditions. For example, two sets of data about photos were used as source, these were both loaded into the same Python script. Based on these two lists, several if statements were made before creating a certain RDF statement. For example, the task in which the photo was made determined whether the photo was linked to an inspection and damage or a repair and damage.

```
IMPORT rdflib and csv libraries

LOAD "issues_issues.csv" into variable issues_issues using csv
CREATE empty Graph using rdflib

FOR each row in issues_issues:
    DEFINE variable project_id containing the 3rd value in row
    DEFINE variable task_id containing the 12th value in row
    DEFINE variable element_id containing the 29th value in row

    ADD statement to Graph: (inst:[element_id] rdf:type bot:Element .)
    ADD statement to Graph: (inst:[element_id] rdf:label "[element_id]" .)
    ADD statement to Graph: (inst:[element_id] cto:isSubjectOfTask inst:[task_id].)

SERIALIZE Graph in Turtle format to file "issues_issues.ttl"
SERIALIZE Graph in N-Triples format to variable issues_Graph
```

Listing 3: Pseudocode Fragment of Python Script - Creating RDF Graph

ETL → Data Layer

The resulting graph is communicated to the data layer using the 'SPARQLWrapper' library. This makes it possible to push the graph directly into the graph database 'GraphDB' that was used. Listing 4 show the used code, first the specifics of the endpoint of the database are stated, then it defines the URI of the specific graph to which the data will be uploaded. This graph is first cleared to make sure it does not contain any data. Finally, it executes a SPARQL query that inserts the data to the named graph. Here the N-Triples are used as input for the query, as this serialisation does not contain prefixes such as Turtle, that would give errors during the insert query.

```
IMPORT SPARQLWrapper library

DEFINE variable endpoint as "http://localhost:7200/repositories/Dam-Test/statements"
DEFINE variable GraphURI as "http://example.org/BIM360/issues_issues"

CREATE SPARQL_engine at endpoint using SPARQLWrapper

DEFINE SPARQL_query: "CLEAR GRAPH <{graph_uri}>"
RUN SPARQL_query using the SPARQL_engine

DEFINE SPARQL_query: "INSERT DATA { GRAPH <{graph_uri}> { {graph_NT} } }"
RUN SPARQL_query using the SPARQL_engine
```

Listing 4: Pseudocode Fragment of Python Script - Insert Data into Graph

Additional method: SPARQL Insert data directly using Python

For some smaller datasets another approach was used, namely instead of pushing the full graph at once to GraphDB it is also possible to do a lot of separate insert queries. Using this method, for every row in the imported CSV a query is made using the values of that specific row. The advantage of this method is that the Where clause can be used with specific values from the row, linking the data to a specific place in the graph. For example, in Listing 5 a link is made to an instance of a *cto:InspectionTask* without explicitly naming it. There are however also disadvantages, because of the number of queries the speed in which the script is executed is also greatly reduced.

```
DEFINE variable EG_URI as "https://nicodebont.nl/element-geometry/ModD/"

FOR each row in Borderellist:
    DEFINE variable Length containing the 1st value in row
    DEFINE variable Name containing the 2nd value in row
    DEFINE variable Element_URI containing "<https://nicodebont.nl/element/ModD/{Name}>"

    DEFINE SPARQL_query: "
PREFIX bot:  <https://w3id.org/bot#>
PREFIX cto:  <https://w3id.org/cto#>
PREFIX omg:  <https://w3id.org/omg#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX nen2660:  <https://w3id.org/nen2660/def#>
PREFIX rii:  <https://janssensteenberg.com/rii#>

INSERT {
    GRAPH <{graph_uri}>{
        {Element_URI} omg:hasGeometry <{EG_URI}+{X}-IHW_geometry> .
        <{EG_URI}+{X}-IHW_geometry> a omg:Geometry ;
            omg:complementsGeometry ?ifcGeometry ;
            rii:capturedDuringTask ?inspectiontask ;
            rii:hasLength <{EG_URI}+{X}-geometry_length> .
        <{EG_URI}+{X}-geometry_length> a rii:Length ;
            <https://schema.org/value> "{Length}"^^xsd:integer ;
            <https://qudt.org/schema/qudt/unit> <http://qudt.org/vocab/unit/MilliM> .
                }
        }
WHERE {
    {Element_URI} a nen2660:TechnicalEntity , bot:Element ;
        rdfs:label ?label ;
        cto:isSubjectOfTask ?inspectiontask .
    ?inspectiontask a cto:InspectionTask .

    ?ifcObject a nen2660:FunctionalEntity ;
        rdfs:label ?label ;
        omg:hasGeometry ?ifcGeometry .
    } "
    RUN SPARQL_query using the SPARQL_engine
```

Listing 5: Pseudocode Fragment of Python Script - Insert Query for every Row

Additional method: SPARQL INSERT WHERE Query

The SPARQL Insert Where query is also used in a more general sense, adding statements based on whether another (series of) statement(s) exists. For example, in Listing 6 a statement is added to the graph for every subject that has a certain predicate.

```
DEFINE SPARQL_query: "
       PREFIX props: <http://lbd.arch.rwth-aachen.de/props#>
       PREFIX fog: <https://w3id.org/fog#>

       INSERT {
           GRAPH <{graph_uri}> {
              ?ifcObject fog:hasIfcId-guid ?GUID .
           }
       }
       WHERE {
             ?ifcObject props:globalIdIfcRoot_attribute_simple ?GUID .
       } "
RUN SPARQL_query using the SPARQL_engine
```

Listing 6: Pseudocode Fragment of Python Script - SPARQL INSERT WHERE Query

Additional method: SPARQL Select for additional input data from Graph

As explained before, sometime the creation of certain statements is dependent on conditions. In most cases these could be created based on the values is the CSV files. However, in some cases this data was not available in the file but was already part of the graph. For these cases a SPARQL Select query was created, see Listing 7 for example. Using this method, a list is created in Python based on the results from the query, this list can then be used to create conditions for the data that needs to be added. For example, link the objects that have the label.

```
DEFINE variable endpoint as "http://localhost:7200/repositories/Dam-Test"
CREATE SPARQL_engine at endpoint using SPARQLWrapper

DEFINE SPARQL_query: "
    PREFIX bot: <https://w3id.org/bot#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX nen2660: <https://w3id.org/nen2660/def#>
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

    Select ?p ?label
    WHERE {
        ?p rdf:type bot:Element .
        ?p rdf:type nen2660:TechnicalEntity .
        ?p rdfs:label ?label .
    } "

SET result format of SPARQL_engine to CSV
RUN SPARQL_query using the SPARQL_engine
STORE query results in variable results

FOR each row in results:
    SPLIT row at ','
    ADD row values to variable Results_List
```

Listing 7: Pseudocode Fragment of Python Script - SPARQL Select Query as additional Input Data

5.4.3 Data Layer | RDF Graph

In this thesis, Ontotext GraphDB[1] has been selected to be used as the RDF database, since this solution has been used in other research and is considered best practice. GraphDB is a graph database and knowledge discovery tool for storing and querying semantic data. It is based on standards such as RDF, OWL and SPARQL. As explained earlier, data is uploaded using the SPARQL endpoint of the database. This research uses the free version of the product, running as localhost.

The resulting graph combines all instance data and is structured by the ontologies used. Figure 35 shows an example of how the data is structured in GraphDB. All data is centred around the element, which is described both functionally and technically using the NEN2660 ontology. The functional representation focuses on the object as a function in the building and the associated requirements. In this case, the implementation is limited to the relationships of the element to the location and geometry in the building. The technical entity focuses on the physical representation of this functional element, the object in the real world, which in some cases is replaced. This object has relations with the properties and damages identified during different tasks, the interventions performed in other tasks to mitigate these damages and, finally, the photos taken to create documentation throughout this process.

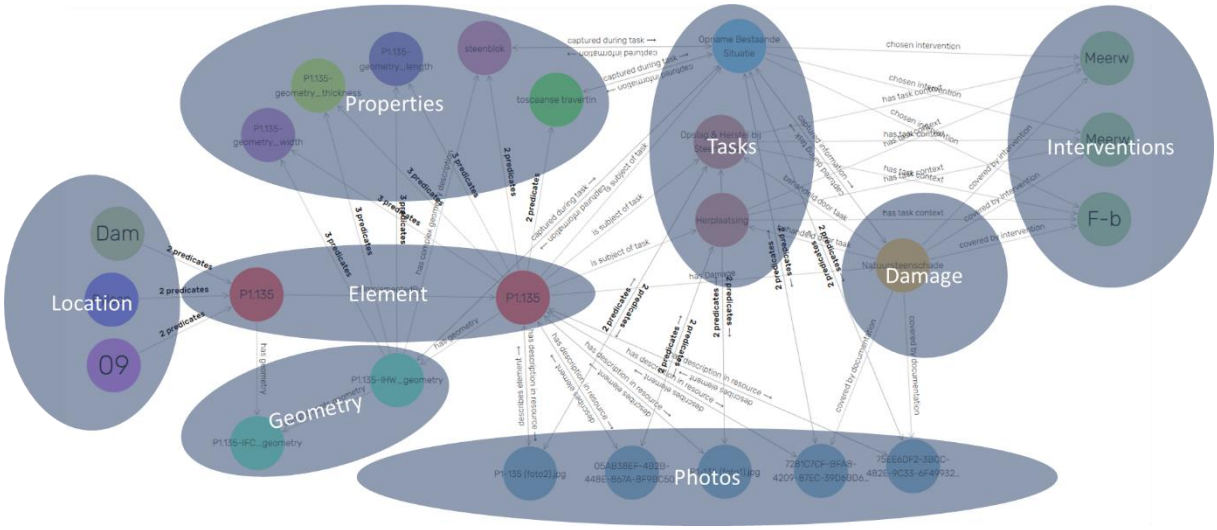


Figure 35: Example of Data in GraphDB

5.4.4 Data Layer | Web-Connected Document Storage

One of the goals of this research is to combine RDF with non-RDF data. To make sure the several non-RDF files can be linked to the graph, they are made available through a local live server through Visual Studio Code, this gives every file a URI that is added to the graph. This simulates the situation where all non-RDF are also available through an URI. Furthermore, there were also URIs added to the graph to link to tasks in the Autodesk Build system, enabling interaction between the two databases.

Data Layer → Services

Different services are used to communicate with the data in the graph database. This is primarily done by using the SPARQL endpoint of the database, which is used to execute different SPARQL queries on the data.

[1] *https://graphdb.ontotext.com/*

5.5 Data Visualisation

There is no point in connecting a lot of data between databases if users will not be able to interact with it. Therefore, one of the main results of the research includes ways to visualise the connected data. However, before the data can be visualised the user interfaces should be able to interact with the databases using services.

5.5.1 Services

Interacting with the RDF database can be done using SPARQL, the prototype only focuses on visualising the data, creating data is not part of the user interfaces developed. Nevertheless, the database makes use of data inference to create new statements. Both are explained below.

Select Data: SPARQL Querying

The SPARQL query language is used to create select queries on the graph database. Throughout the research many SPARQL queries have been created. The result of these queries is used as input data for visualisation purposes in the user interface. Listing 8 shows an example of a query that was created. The query requests for a list of unique combinations of elements and their tasks. The WHERE clause is used to define the location of this data.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cto: <https://w3id.org/cto#>
PREFIX bot: <https://w3id.org/bot#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX nen2660: <https://w3id.org/nen2660/def#>
PREFIX rii: <https://janssensteenberg.com/rii#>

SELECT DISTINCT ?Element ?Task
WHERE {
    ?functionalElement rdf:type bot:Element , nen2660:FunctionalEntity ;
                rdfs:label ?Element .
    ?functionalElement nen2660:isImplementedBy ?technicalElement .
    ?technicalElement cto:isSubjectOfTask ?task .
    ?task rdfs:label ?Task .
}
ORDER BY ?Element
```

Listing 8: Example SPARQL

Infer Data: OWL

In the database, semantic inference based on OWL is enabled, which results in new statements that can be derived from existing statements in the graph. This creates a situation where there are both explicit and implicit statements in the graph. For example, Figure 35 shows some relations that have multiple predicates because the original predicate is a subclass of another. There are also some predicates that have relations in both directions. These were not defined in the data but inferred from the respective ontologies. All this reduces the work involved in creating data and makes querying more convenient.

Data Layer → User Interface

The output of SPARQL queries is used in the user interfaces. The results of a SPARQL query can be serialised in various formats. This depends on the tools used to execute the SPARQL queries. SPARQL allows the creation of different queries that can return the same results in different formats, depending on the needs of the user interface.

5.5.2 User Interface | Heritage LBDviz

To demonstrate the possibilities of combining RDF and non-RDF data in the context of restoration projects, a front-end web application has been developed, creating an easily accessible application that a user can visit through their browser. HTML, CSS and JavaScript are used to create the content of the page, giving the application structure, style and logic.

The decision was made to build on an existing application that already combined RDF graphs with an IFC viewer in a different context, see Figure 36. This application, the LBDviz, is a single page web application that has a full page IFC viewer with several menu windows on top of the viewer (Donkers et al., 2023). To display a model in IFC format, LBDviz uses IFC.js. This is a JavaScript library that allows users to load, view and edit IFC models directly in the browser. It is the official IFCLoader of Three.js, which allows the Three.js geometry of IFC models to be generated in JavaScript. It also parses the file so that you can interact with it through JavaScript. One of the menu windows has the ability to query RDF data with SPARQL using the Comunica JavaScript library. This is a query engine that can be used to execute SPARQL queries against multiple SPARQL endpoints or RDF files simultaneously.

LBDviz already had several ways to execute SPARQL queries. One could use the SPARQL tab in the Queries window to specify a query. When the query is executed, it searches all the RDF files linked to in the graph locator. The results are displayed in the Results window. However, as not every user will have the ability to create SPARQL queries, the tool also includes buttons with predefined SPARQL queries. The last feature uses interaction with the model as additional input for the query. When a specific element is selected, the GUID of that element is extracted. For specific query buttons, the GUID is inserted into the SPARQL query and the results for that specific element are printed.

This thesis extends the functionalities of LBDviz. The resulting Heritage LBDviz adds several new features to address the use cases that exist for this type of restoration project. However, they could also be used in other use cases. These functionalities mainly focus on extending the possibilities of interaction between the model and the corresponding RDF graph. The full code of the Heritage LBDviz can be found in Appendix II.

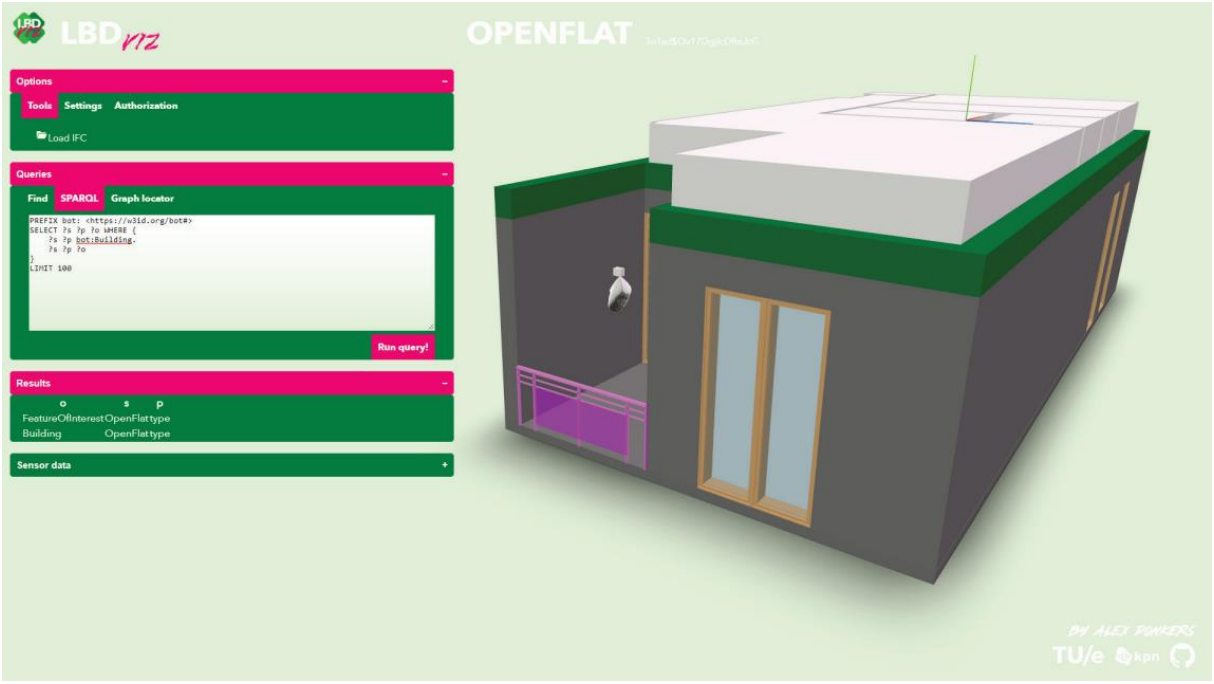


Figure 36: LBDviz by Donkers et al. (2023)

Finding Object Label

The tool has been enhanced in various ways. One of the simplest examples is the automatic query that takes place when an object is selected in the model, which looks for the label given to the object in the graph and displays it in the top right corner of the viewer, see Figure 37. This allows the user to instantly find the name of the selected object.

Figure 37: Heritage LBDviz - Finding Object Label

This function is created by querying the selected GUID and displaying the result on the page, see Listing 9. First the QueryEngine is loaded from Comunica, which allows querying RDF graphs. A query engine is created using the files in the graph locater as input for the RDF graph. Then the SPARQL query is executed against the RDF graph using the IFC GUID of the currently selected element as input. Finally, the results are streamed, the code extracts the value of the label and updates the HTML element to display that name.

```
IMPORT QueryEngine from comunica library

DEFINE function queryComunicaElementName
       CREATE new QueryEngine
       GET value from HTML element and SPLIT into array of graphs
       GET value of selected GlobalID and STORE in variable GUID

       DEFINE SPARQL_Query:
         SELECT ?label WHERE {
           ?ifcElement <https://w3id.org/fog#hasIfcId-guid> {GUID} ;
             rdfs:label ?label .
         }
         LIMIT 1 , {
       RUN SPARQL_Query on graphs using the QueryEngine

       STREAM query result
              GET value of label
              ADD label value to HTML element
```

Listing 9: Pseudocode Fragment of JavaScript Code - Finding Object Label

Built-in IFC.js Viewer Functionalities

One of the advantages of working with IFC.js is that much of the functionality for the viewer has already been developed and is therefore easy to implement. For example, LBDviz already included the ability to create sectional views in the model viewer using a set of hotkeys. During the development of the prototype, the standard dimensioning tool available in the ifc-web-viewer part of IFC.js was also added to the viewer. This makes it possible to measure distances between elements in the model being viewed, see Figure 38. Activation of this tool is also placed under a hotkey and an options button, making it available on command.

Figure 38: Heritage LBDviz - Built-in IFC.js Viewer Functionalities

Showing Images in Viewer

A further feature added was the ability to display images in the results of a query button, see Figure 39. As the graph contained URIs of the locations of the images on the live server, these could also be queried. When a particular item is selected in the viewer, the GUID of that item is added to the query. The result stream of the SPARQL query is stored in a list. This list of resulting URIs is then processed in JavaScript to create an anchor and image element within the HTML, see Listing 10. The images resulting from the query are displayed in a custom Photos window.

Figure 39: Heritage LBDviz - Showing Images in Viewer

```
RUN SPARQL_Query on graphs using the QueryEngine

STREAM query results
       CREATE list PhotoList
       FOR every result in results
              GET value of PhotoURI
              ADD PhotoURI to PhotoList

WHEN STREAM ends
       FOR every PhotoURI in PhotoList
              CREATE HTML image element with PhotoURI as source
              ADD PhotoURI as hyperlink that opens on new page to the element
              ADD HTML image element to Gallery
       ADD Gallery to HTML page
```

Listing 10: Pseudocode Fragment of JavaScript Code - Showing Images in Viewer

Highlighting Objects in IFC Viewer with Query Button

All the functions so far work with the model in one direction, selecting a particular object and retrieving some information about that object from the RDF graph. However, the following function takes the opposite approach, querying the RDF graph and displaying the results by automatically selecting the objects in the viewer, see Figure 40. To achieve this, several steps are taken, see Listing 11. First, a SPARQL query is executed, the results of which include a number of GUIDs. These results are then displayed in the results window and processed into an array of GUIDs. In the background, a mapping table is created that maps each GUID in the model to its corresponding ExpressID, as ExpressIDs can be used in the functions of IFC.js. This mapping table is then used to create an array of ExpressIDs based on the GUIDs. These ExpressIDs are then used to select these objects in the viewer and create a 'subset', which is an IFC.js function allowing a specific array of objects to be given a colour.

This function was later extended to allow the sets of objects to be given different colours based on an additional property of the specific object, filtering them by a particular type. Figure 41 illustrates how this works. This currently works based on a list of pre-defined colours, so that if the additional value type next to the GUID is part of this list, the object is automatically added to the appropriate subset with its specific colour. This could be extended to randomly colour the items where the colour is undefined.

Figure 40: Heritage LBDviz - Highlighting Objects in IFC Viewer with Query Button

```
IMPORT QueryEngine from comunica library
IMPORT IFCviewer from IFCjs library

RUN SPARQL_Query on graphs using the QueryEngine

STREAM query results
	CREATE ResultsArray
	FOR every result in results
		GET value of GUID
		GET value of label
		GET value of type
		ADD [GUID, label, type] as row in ResultsArray

WHEN STREAM ends
	REMOVE every subset in IFCviewer
	GET all elements in IFCviewer
	CREATE MappingArray
	FOR every element in elements
		GET value of GUID
		GET value of ExpressID
		ADD [GUID, ExpressID] as row in MappingArray

	FOR all GUID values with the same type value in ResultsArray
		CREATE GUIDsArray with all values of GUID
		CREATE ExpressIDsArray
		FOR every GUID in GUIDsArray
			GET value of GUID
			FIND value of matching ExpressID in MappingArray
			ADD ExpressID as to ExpressIDsArray
		CREATE new subset in IFCviewer
		SET colour of subset based on type
		ADD all elements in ExpressIDsArray to subset

	CREATE new HTML table
	ADD new row to table as headerRow
	ADD cell to headerRow containing "Label"
	ADD cell to headerRow containing "Type"
	FOR all result values in ResultsArray
		ADD new row to table
		GET value of label from result
		ADD cell to row containing label value
		GET value of type from result
		ADD cell to row containing type value
	ADD table to HTML page
```

Listing 11: Pseudocode Fragment of JavaScript Code - Highlighting Objects in IFC Viewer with Query Button

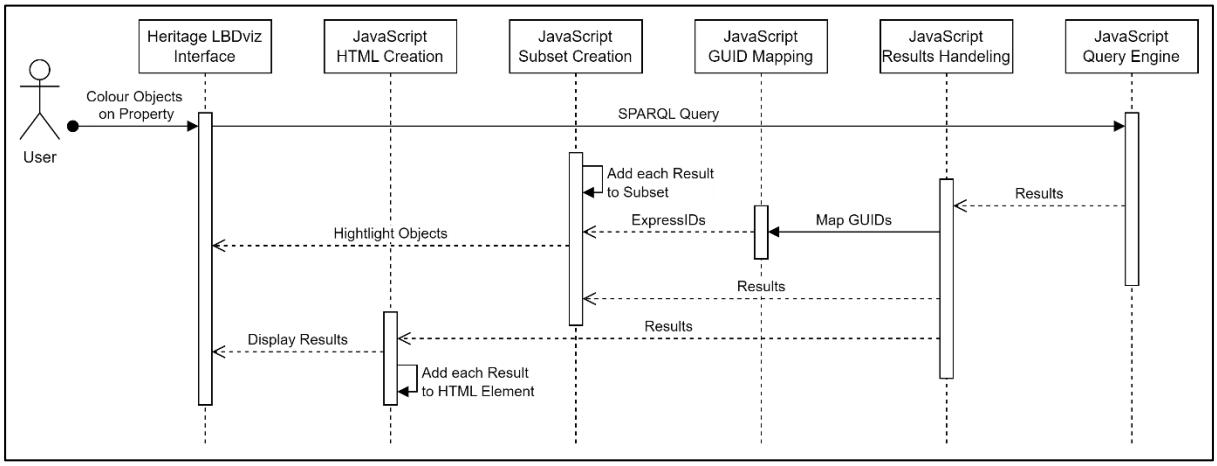


Figure 41: Sequence Diagram - Highlighting Objects in IFC Viewer with Query Button

Highlighting Object in IFC Viewer from Results

To allow more interaction between the model and the results, a similar implementation has been created for the results. When the result of a query consists of many items, it can be difficult to find a specific item that has the parameter one is looking for. Clicking on the object ID in the results selects the object in the viewer and centres the view around the object, see Figure 42. This allows the user to see where the object is in the building. This function is created in a similar way to the previous one, using the GUID in the mapping table to find the ExpressID and using that ID in the standard IFC.js function to select an object by ID. This function also provides the ability to focus and centre on the selected object. It automatically changes the value of the last selected GUID in the model to the GUID resulting from the query. This ensures that the selected object is also named in the top right corner.

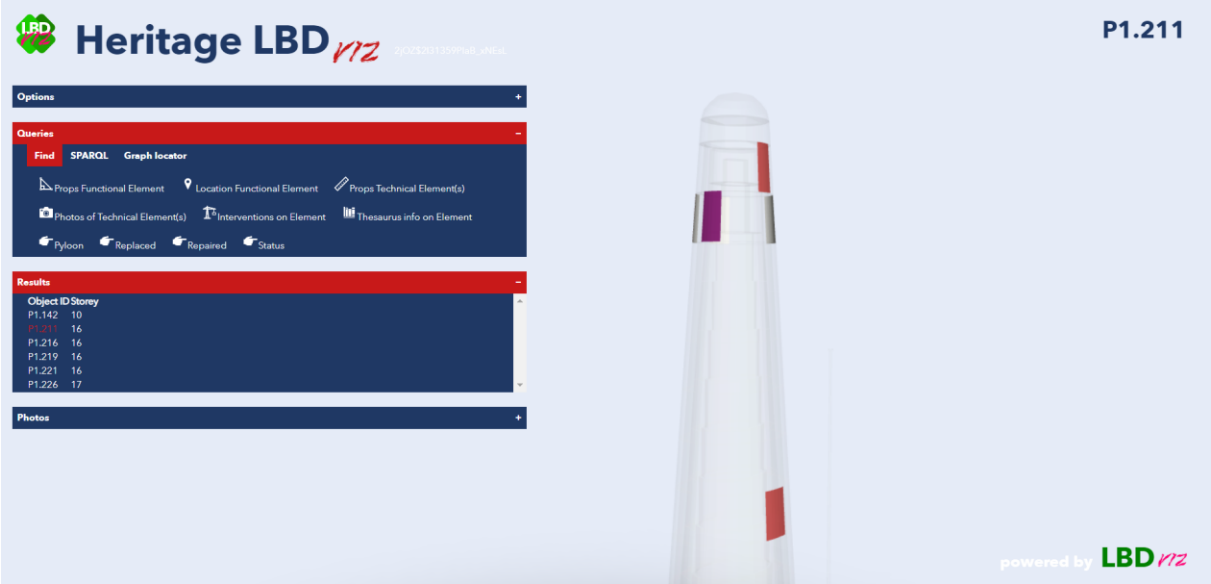


Figure 42: Heritage LBDviz - Highlighting Objects in IFC Viewer from Results

Creating Links to Objects in IFC Viewer

The last major feature added to LBDviz is the ability to handle URIs that, when opened in a browser, will directly load the correct model and automatically select the desired object in the IFC viewer. This works as follows, behind the web address where the Heritage LBDviz is located is a hash followed by a GUID, for example ".../Heritage-LBDviz/#1$PRPlVFT2oe_DY_uNLz7K". When the page is launched, the web application will automatically check if a GUID is defined after the hash and store it in a variable. The original LBDviz required the IFC model to be manually loaded into the viewer. However, to create this feature, a change was made to automatically load the predefined IFC model of the case project. If the viewer would be used for multiple projects, the GUID could also be used as input for a SPARQL query to find the corresponding IFC model location in the diagram, as GUIDs are unique. However, the prototype contains a hardcoded link to the IFC model on the live server. When the IFC is loaded into the viewer, a check is made to see if a GUID variable has been defined. If so, the GUID in the variable is used to select the object in the IFC viewer by first mapping the GUID to an ExpressID. This allows users to quickly find an object using Heritage LBDviz, see Figure 43.

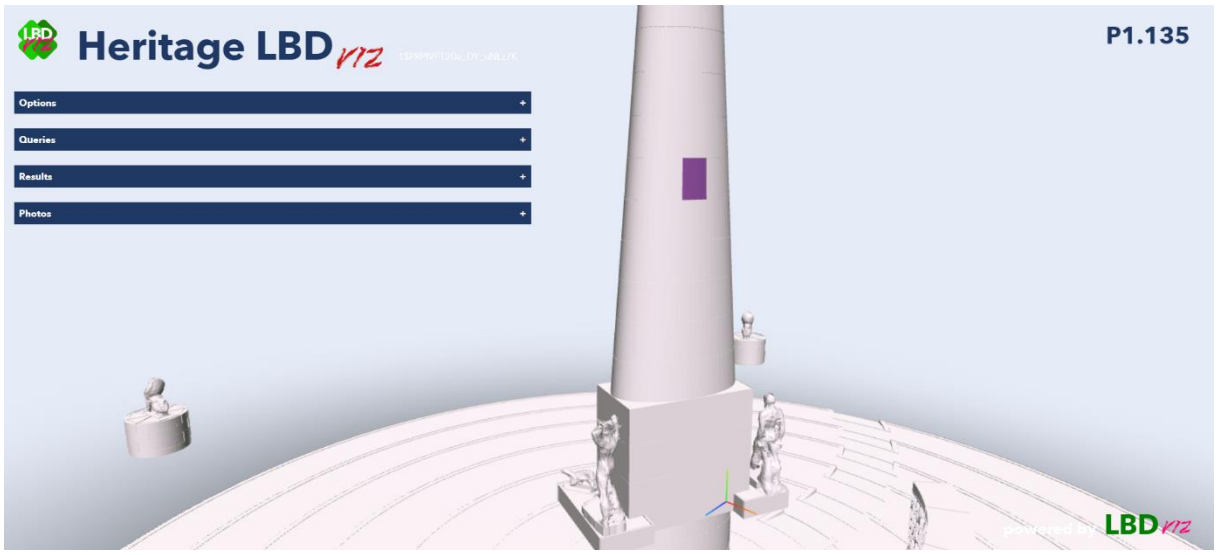


Figure 43: Heritage LBDviz - Creating Links to Objects in IFC Viewer

5.5.3 User Interface | Spreadsheet Export

In addition to the Heritage LBDviz, the graph database was also used to automatically create an enhanced version of the traditional spreadsheet used in the current process. This export was created using the combination of the SPARQLWrapper and pandas libraries in a Python script, see Listing 12. The SPARQL query, which can be found in Appendix III, contains several more complex SPARQL functions as it combines data from two technical objects with their functional representation. The properties of the existing and, if applicable, the new version of a natural stone object are combined in one row in the results. Figure 44 shows the result of the export script. The values shown in red have a hyperlink underneath them to open a photo or use the Heritage LBDviz to open the viewer directly on that specific element, extending the traditional spreadsheet with new possibilities that arise from using the linked data structure.

```
IMPORT SPARQLWrapper and pandas libraries

DEFINE variable endpoint as "http://localhost:7200/repositories/Dam-Test"
CREATE SPARQL_engine at endpoint using SPARQLWrapper
DEFINE SPARQL_query: "...."
SET result format of SPARQL_engine to JSON
RUN SPARQL_query using the SPARQL_engine
STORE query results in variable results

CONVERT JSON results into dictionary
CONVERT dictionary into list

CREATE dataframe of list using pandas
CHANGE "LBDviz" and "Photo" values in dataframe into excel hyperlink formula

SAVE dataframe to excel file "SPARQL_output.xlsx"
```

Listing 12: Pseudocode Fragment of Python Script – Export Data from Graph

	A	B	C	D	E	I	J	N	O	P	Q	R
1	LBDviz	Building	Storey	label	Shape	Volume_ Existing	Material_ Existing	Volume_ New	Material_New	LastChangedTask	Photos_ Existing	Photos_ Repaired
152	Open in Model	Pyloon	16	P1.215	steenblok	0.088315	toscaanse travertin			Herplaatsing	Photo	Photo
153	Open in Model	Pyloon	16	P1.216	steenblok	0.088315	toscaanse travertin	0.090125	travertin alabastrino	Herplaatsing	Photo	Photo
154	Open in Model	Pyloon	16	P1.217	steenblok	0.088315	toscaanse travertin			Herplaatsing	Photo	Photo
155	Open in Model	Pyloon	16	P1.218	steenblok	0.088315	toscaanse travertin			Herplaatsing	Photo	Photo
156	Open in Model	Pyloon	16	P1.219	steenblok	0.088315	toscaanse travertin	0.060564	travertin alabastrino	Herplaatsing	Photo	Photo
157	Open in Model	Pyloon	16	P1.220	steenblok	0.088315	toscaanse travertin			Herplaatsing	Photo	Photo

Figure 44: Spreadsheet Export from Graph Database

6 PROTOTYPE VALIDATION

This chapter presents the validation of the prototype system in relation to the requirements developed with industry experts. The validation process initially focuses on assessing the system's ability to provide answers to the various Competency Questions. Next, a particular object is chosen to reflect the functionalities of the system throughout the process. Lastly, the findings are discussed.

6.1 COMPETENCY QUESTIONS

Throughout the different iterations, validations were performed to verify that the created interfaces provided answers to the competency questions defined in the methodology. This section describes for each Competency Question (CQ) which functionalities help to answer these questions.

CQ 1: What is the location of the object in the building?

Using the most traditional approach, this question can now be answered using the spreadsheet. If a user wants to know where a particular object is located, they can directly open the Heritage LBDviz where the specific object is selected. The same applies to the results of a button query. Clicking on the element in the results, selects the object in the model viewer, helping the user to find the object. In addition, the model viewer can be used simply to navigate and select objects in the building, whereupon the Heritage LBDviz instantly displays the name of the object in the top right corner. When an object is selected, a button query can be used to get more information about its location. Listing 13 shows the SPARQL query that selects information about the element in the building. The results of this query give more insight into the specific building or floor the object is part of.

```
SELECT DISTINCT ?Element ?Type ?PartOf
WHERE {
  ?ifcElement fog:hasIfcId-guid "1$PRPlVFT20e_DY_uNLz7K" ;
              rdfs:label ?Element .
  ?something  bot:hasElement ?ifcElement ;
              rdfs:label ?PartOf ;
              a ?Type .
  FILTER (?Type NOT IN (owl:Thing, bot:Zone))
}
```

Listing 13: SPARQL Query CQ1

CQ 2: What photos are available of a specific object?

The inclusion of links to the photos in the graph gives both the spreadsheet and Heritage LBDviz the capability to refer to the photos of an element. In the spreadsheet they are added as a hyperlink to the web location of the photo. In Heritage LDBviz this feature is complemented by a tab where the photos are displayed. Listing 14 shows the SPARQL query used to select the photos from the graph.

```
SELECT DISTINCT ?Element ?Photo ?Photo_Link
WHERE {
  ?ifcElement fog:hasIfcId-guid "1$PRPlVFT20e_DY_uNLz7K" ;
               nen2660:isImplementedBy ?element.
  ?element rdfs:label ?Element ;
           rii:hasDescriptionInResource ?photo .
  ?photo rdfs:label ?Photo ;
         rii:filePath ?Photo_Link .
}
```

Listing 14: SPARQL Query CQ2

CQ 3: What is the volume, material, and shape of the object?

Similar to the photos and the location of an object, some of the basic properties of an object can also be found using the spreadsheet and the Heritage LBDviz, see Listing 15. A filter is used to ensure that none of the super classes are included in the results. Volume is not part of the data but can be calculated using the dimensions of the object, see Listing 16.

```
SELECT DISTINCT ?element ?Element ?Property_Type ?Property_Value ?Created
WHERE {
  ?ifcElement fog:hasIfcId-guid "1$PRPlVFT20e_DY_uNLz7K" .
  { ?ifcElement nen2660:isImplementedBy ?element . }
  UNION
  { ?ifcElement nen2660:isImplementedBy ?elementNew .
       ?elementNew rii:replacesElement ?element .  }
  ?element rdfs:label ?Element ;
       rii:hasElementProperty ?property.
  ?property rdf:type ?Property_Type ;
       rdfs:label|<https://schema.org/value> ?Property_Value ;
       <http://www.w3.org/ns/prov#generatedAtTime> ?Created .
  FILTER (?Property_Type NOT IN (owl:Thing, rii:ElementProperty,rii:Dimension))
  }
```

Listing 15: SPARQL Query CQ3 1/2

```
SELECT DISTINCT ?element ?Length ?Thickness ?Width ?Volume
WHERE {
  ?ifcElement fog:hasIfcId-guid "2s4IjT4M5DQObqvIOMvgxP" .
  { ?ifcElement nen2660:isImplementedBy ?element . }
  UNION
  { ?ifcElement nen2660:isImplementedBy ?elementNew .
    ?elementNew rii:replacesElement ?element .  }
  ?element rii:hasLength ?length ;
    rii:hasThickness ?thickness ;
    rii:hasWidth ?width .
  ?length <https://schema.org/value> ?Length .
  ?thickness <https://schema.org/value> ?Thickness .
  ?width <https://schema.org/value> ?Width .
  BIND((?Length * ?Thickness * ?Width)/1000000000 as ?Volume)
}
```

Listing 16: SPARQL Query CQ3 2/2

CQ 4: What interventions are conducted on an object?

In the same way as the previous questions, a query can be done on the interventions that are conducted on an object. Returning a list of all interventions that are related to that specific object, using a similar approach as the query listed in Listing 15.

CQ 5: How to give an overview of all relevant information on an object?

The traditional spreadsheet is mostly focused on the different properties of each object in the building. However, once it is based on the graph, it can be extended with new features. This question is addressed by the creation of the spreadsheet export from the graph, combined with the ability to link to photos and objects in the Heritage LBDviz model viewer. The result is a tool that combines the familiar way of using information with the new possibilities offered by the linked data structure. In this way it also serves as a way to introduce beginners to Heritage LBDviz and its possibilities.

CQ 6: What is the height of the object relative to ground level?

The created prototype can answer this question in two ways. One can use the dimensions tool in the Heritage LBDviz model viewer to measure the distance of the object to the ground-level. Alternatively, a SPARQL query can be created that uses information about the height of the floor on

which an object is located defining the height of the object. However, it should be noted that this height is affected by the origin chosen in the IFC model, as the zero on the Z-axis is not always equal to the ground level. In this case project the Z-axis is equal to the Dutch standard NAP, which means that some amount must be subtracted from the result to get the actual distance.

CQ 7: What is the progress status of every object in the building?

In the traditional spreadsheet workflow, the status of an object is tracked manually. On one hand by updating the status in the spreadsheet, and on the other hand by highlighting the objects in different colours on an 2D drawing. Based on the process schema that was created with industry experts in the requirement definition, it was concluded that the statuses used were derivatives of the progress of tasks. Therefore, this research proposed a definition of status that is based on the latest completed task. As tasks and the completion thereof are part of the dataset, status can be automatically assigned to an object based on the data in the graph. Listing 17 shows the SPARQL query used, where for all tasks the completion time is compared to find the task that was last completed.

```
SELECT DISTINCT ?Element ?LastChangedTask
WHERE {
  ?ifcElement fog:hasIfcId-guid "2s4IjT4M5DQObqvIOMvgxP" ;
    rdfs:label ?Element .
  {    ?ifcElement nen2660:isImplementedBy ?element .
       ?element cto:isSubjectOfTask ?task . }
  UNION
  {    ?ifcElement nen2660:isImplementedBy ?elementNew .
              ?elementNew rii:replacesElement ?element .
              ?element cto:isSubjectOfTask ?task .  }

  ?task rdfs:label ?LastChangedTask ;
        prov:endedAtTime ?DateTime .

  {    SELECT  ?label (MAX(?dateTime) AS ?maxDateTime)
             WHERE {  ?element cto:isSubjectOfTask ?task ;
                              rdfs:label ?label .
                      ?task prov:endedAtTime ?dateTime . }
       GROUP BY ?label }
  FILTER (?DateTime = ?maxDateTime)
}
```

Listing 17: SPARQL Query CQ7

The result of this query can be showed in both the spreadsheet and the Heritage LBDviz as a value. However, an even more powerful combination can be created using the highlighting function that was created in the model viewer. This allows a user to visualise the statuses of every object in the building by colouring then in the model viewer of the Heritage LBDviz with the click on a query button.

CQ 8: How to check if all objects include the required information?

The use of more data in workflows also increases the reliance on complete and high-quality data. Linked data technologies offer several ways to validate data, including SHACL, which were not part of the scope of this prototype. However, visualisation of data can be a basic tool to check whether data is complete. For example, the spreadsheet export can be used to validate that all required values are defined. In addition, the highlighting capabilities of Heritage LBDviz allow the presence and values used for properties associated with an element to be visualised. For instance, this is shown in Figure 40, where the status is visualised with different colours. However, there are also several objects that remain white because no status could be defined, directly indicating to the user that some data is missing.

6.2 EXAMPLE PROCESS

In order to validate the value of the developed system in practice, a specific object has been selected in the case project that could be discussed with industry experts. This object was used to reflect on the value of the created system in answering the questions that arise in the process described earlier (Figure 24). For this purpose, the object 'P1.142' has been selected, as this object has been replaced and therefore contains most of the steps that are completed in the process. This subsection reflects on the developed system through every relevant step of the process that was described in Figure 24. Before going into specific steps in the process, some general findings are discussed.

6.2.1 General Findings

Combining Data

One of the common factors that influence the resources required to find information is the effort required to gather documents and open the necessary systems. To answer the different competency questions in the current workflow, a user often needs to combine information from different sources, such as the 2D drawings, spreadsheets and data in the Autodesk Build system. Assuming a beginner user of the proposed system, one would need the spreadsheet and open the Heritage LBDviz. Comparing the two workflows, one would need to find the spreadsheet in both cases, the 2D drawings would be replaced by the Heritage LBDviz tool which loads in a few seconds, and finally the user would no longer need to look at the Autodesk Build system as it would be found by the tool.

This requires some effort to create a 3D model of the building, but this effort can be compared to the creation of 2D drawings if a pragmatic approach is adopted that limits the level of detail to the basics, as in this case project using SketchUp. Finally, the graph database must be created, but this process can be almost completely automated. Taking all this into account, it was assumed that there would be no major difference in the resources required between the current process and the usage of the proposed system.

Highlighting Status in Model Viewer

Throughout the process, stakeholders want to monitor the progress of the project. Heritage LBDviz can be used to visualise the status of the P1.142 at any point in time using highlighting colours in the model viewer. This creates a solution that has the potential to completely replace the traditional 2D drawing used on site to monitor progress. This creates a situation where information on the status of objects is more widely available to stakeholders. This solution is also less prone to error, as the status is currently tracked manually, which has a higher error rate. As tasks are performed on object P1.142, the status resulting from the query to the graph database will automatically change as new task progress is uploaded to the graph database from, in this case, Autodesk Build. When a user highlights the status in the Heritage LBDviz, it will always show the current status of the graph. This eliminates the risk of using outdated information.

Querying Data

One of the limitations that was found in this process is the execution speed performance of the more complex SPARQL queries that are executed using Comunica query engine. The general rule is that as SPARQL queries are more complex, the time needed to generate a response increases. Therefore, an effort was made to develop queries that can generate a quick response. This was done by using the query engine in GraphDB in a process where queries were simplified as much as possible to optimise response speed. However, in some cases quite complex queries were required. In the case of the calculation of the statuses of every object, every task on an object must be compared to all other tasks to find the one that is the most recent. When executing this query in the GraphDB SPARQL query

engine it takes 5,3 seconds before a response is given. This is quite an increase compared to other simpler queries that often only take 0,1 or 0,2 seconds but would still give a user an answer in a reasonable amount of time. However, when executing the same query using Comunica the response time increases to 120,2 seconds, so more than two minutes. Therefore, effort should be made to improve this. However, for validation purposes the results have been hardcoded in the script to validate the functional value of this feature.

6.2.2 Restoration Process | Preliminary / Definitive Inspection

The current version of the system does not change the way inspections are done on the building site. The model viewer could however be used in locating object P1.142 in the building when it needs to be inspected. A future version could include a use case where it is also possible to create data within the Heritage LBDviz that is then added to the graph database. However, at the end of the inspection, the created prototype can be used to check the completeness of the data.

6.2.3 Restoration Process | Cost Estimation

Cost estimation combines all the information gathered during the inspections with expert knowledge to determine the cost of the interventions to be carried out on the object. The estimator uses the traditional spreadsheet to make the calculation. The properties of the object, such as its shape, material and dimensions, are used to estimate the cost of reconstructing P1.142. However, the total cost is also influenced by the context in which the object is located. In the current process, the estimator uses the 2D drawings and photos in Autodesk Build to check aspects such as accessibility, surrounding objects and the quality of the inspection data. When compared to the proposed system, there are several improvements in this process. The improved spreadsheet can be used as a single source for all the information required, as links to the model viewer and photos of the object are integrated. This saves a lot of time that would normally be spent searching for the corresponding 2D drawings and photos in the respective systems.

As the data is already linked in the database, it can also be used to automate some calculations. For example, the current version of the spreadsheet export automatically calculates the gross volume of P1.142, saving some manual work in the spreadsheet. A future implementation of this concept could be to add data on the cost per unit of interventions to the graph. This would provide the opportunity to further automate cost estimation as more parameters affecting cost are added to the graph.

6.2.4 Restoration Process | Dismantling / Reconstruction

Before the actual work starts on the construction site, the Heritage LBDviz can be used as an additional tool alongside the 2D drawings to find the location of the object to be dismantled. In the stonemasonry, the traditional spreadsheet is used to find the specifications of the reconstruction that needs to be created. However, as the current process involves multiple spreadsheets, there is a high risk of incomplete or incorrect information. In practice, this leads to cases where newly created elements have the wrong dimensions or are created twice due to errors in the spreadsheet. With the cost of reproducing an object such as P1.142 running into the thousands of euros, eliminating these errors can result in significant savings. By using the Heritage LBDviz, one can be sure that the latest information from the graph database is being used, thus avoiding these costs. The improved version of the spreadsheet still carries the risk of using outdated data, so it is important to ensure that a freshly exported version is used. A future implementation of the spreadsheet could be fully web-based, always displaying the latest data from the graph database.

6.2.5 Restoration Process | Transport to Site

After the reconstruction is completed at the masonry workshop, the object is stored there until it is requested for transport to the construction site. When requested through a spreadsheet, pallets of one or more objects are transported back to site. This sometimes creates situations where a part of the pallet has to be unstacked in order to reach the first object. The current version of Heritage LBDviz includes a model viewer that can assist in stacking the objects in the correct order. A future implementation of the Heritage LBDviz could provide functionality to directly request a transport based on the objects selected in the model viewer, including the correct reassembly order.

6.2.6 Restoration Process | Reassembly

Before the reassembly work can start, the pallet must be transported from the storage on the site to the correct location on the scaffolding. In the current process, the QR code that is attached to the object can be scanned through the Autodesk Build system. In this system this QR code is used to open a pushpin on a 2D drawing. This enables the user to find the location in the building of the respective element, after which it can be transported to the correct location on the scaffolding. The Heritage LBDviz can also be used as a tool to find the location of the object in the building. Attaching a QR code of the weblink that opens P1.142 in the Heritage LBDviz on the object allows a user to directly find the location of the object through the 3D model viewer.

6.2.7 Restoration Process | Delivery Check

After de reassembly of P1.142 was completed, the final delivery check could be executed. Part of this delivery is to make sure all required information was collected. In the current process, this requires an effort of collecting all the information from all different systems and files make sure the data is complete. This information is then manually combined in a report that can be handed over to the client. The proposed system can already provide a complete overview of all the relevant data, both through the spreadsheet and the Heritage LBDviz. A future development of the system could provide additional functionality to directly generate reporting for the client in a traditional PDF but also in approaches such as creating an ICDD based on the information in the graph database.

6.2.8 After Delivery

Together with some stakeholders, the value of the proposed system after delivery of the project was discussed. It was emphasised that having access to historical information is crucial for effective preparation of the next conservation cycle, as there will always be a next restoration. Therefore, there also arises the opportunity to connect to existing historical data in several archives. However, at present, this historical information can often only be referenced rather than directly accessed, as the information is dispersed and cannot be digitally shared due to copyright restrictions. An example of this would be the documenting the specific types of natural stone, their respective quarries, and their placement in the building. This information is essential at the start of the next cycle and now often stems from individuals, and in cases where information is lacking, external experts are be consulted. Creating connected data could help in making this information more findable and could allow for information retrieval at an object level.

6.3 Discussion

The created prototype data structure used the modular network of ontologies that was developed by the LBD-CG. This structure was extended by the Restoration Intervention Information ontology to define interventions that are conducted restoration projects. Furthermore, it helps defining specific extensions with properties that are used in natural stone restoration projects. This follows the principles that were established by different members of the LBD-CG that suggest a modular and flexible approach in creating ontologies (Bonduel, 2021; Hamdan et al., 2019; Rasmussen et al., 2021). The added ontology enables the use of the data structure on the data that was collected during the restoration of the Monument on Dam Square. This extends the ontology network of Bonduel (2021) with the integration of data from practice. The created classes and properties in this ontology focus only on the case project that was used for the prototype. However, due to its modular and flexible nature, the ontology could easily be extended for use in projects that have some additional needs.

The development of the Heritage LBDviz extends on the work of Donkers et al. (2023). Their research suggested that co-creating the LBDviz application with industry experts may reduce the barriers to using linked data in practice. This research shows that combining linked data with other files to create connected data does just that. The methodology used resulted in a set of requirements that mostly includes basic information needs that exist today, as it was often found that it is difficult for industry experts to determine what functionalities are needed based on the potential of connected data. The validation of the Heritage LBDviz prototype showed that demonstrating the possibilities of combining data helps industry to better understand the possibilities and value of creating connected data. Often, it is only after seeing the prototype that experts can imagine more complex questions that could be answered by connecting data. Based on the initial questions, the tool developed already includes several functions enabled by connected data that add value to the current process. However, the demonstration of this tool shows that there are still many future opportunities on the horizon for using connected data in a restoration project context.

The functionalities of the current prototype were mainly based on the requirements of the general contractor who provided the data of the case project. However, the connected data could be of value to several stakeholders during the execution phase of a restoration project. Sharing tool access with other stakeholders in the process may offer value by facilitating additional data connection and adding further functionalities, allowing all stakeholders to benefit from interconnected information.

The functionalities developed in the prototype mainly focused on visualising connected data to answer the information questions of the end user. However, as stated before, validating the quality of data is essential to make sure the end user will be provided with information that is reliable. The current prototype of the Heritage LBDviz and the spreadsheet provides basic functionalities to visually validate data. However, there the current prototype does not include automated data validation with SHACL as suggested by Van den Bersselaar (2022). Combining the two could provide a powerful workflow where data is connected, validated, and visualised.

At the end of the project, the data should be preserved since it holds value for subsequent cycles of conservation. The system proposed by Van der Pas (2022) aims to ensure that the data will remain interpretable, accessible, and easy to find during the operational phase of a building. Access to a complete dataset of past restorations could create many opportunities and be highly valuable, as the current restoration project will never be the last. However, the question remains about who should be responsible for maintaining this information after the project is completed.

7 CONCLUSION

This concluding chapter summarises the results of the research and explains the contribution of the study in scientific and societal contexts. Lastly, suggestions for further research and implementation are made.

7.1 SUMMARY OF RESEARCH

This research aimed to facilitate and improve the structuring and visualisation of data during the restoration of natural stone objects on historical buildings by using connected data structures that are vendor-independent, extendable, and flexible. This has been summarised in the following research question:

What value can be created by using linked data technologies and BIM for structuring and visualising a connected data model during natural stone restoration projects on historical buildings?

To help answer the main question, literature research has been conducted into the object information that is used during natural stone restoration projects. This research was combined with literature about how Building Information Modelling (BIM) and linked data could support in connecting this data. Additionally, industry experts gave input about their processes and the information requirements inherent to these processes. Combining literature and industry experts input a prototype was developed and validated. Four sub-questions were addressed throughout this process.

The first part of the literature study focused on answering: *Which object-related data is currently used in natural stone restoration projects*. The research found that Dutch building conservation is governed through self-regulation and guidelines established by the Dutch Cultural Heritage Agency (RCE) and the ERM Foundation. The ERM guidelines provide technical specifications, quality standards and information requirements for restoration work, ensuring the preservation of historical value throughout a continuous process. This "Conservation Cycle" involves inspections, damage diagnoses, intervention strategy using the "Conservation Ladder", on-site intervention works, and the monitoring of the building during operation. The choices made and information gathered during the entire conservation cycle should be documented to provide valuable information for future projects on the same or other buildings. In natural stone restoration projects, it is necessary to document aspects such as cultural-historical value, damage assessment, damage identification, material type, and shape. Currently, a wide range of standards and data formats are used to document these aspects, creating a situation with fragmented data that is difficult to manage.

The second part of the literature study focused on answering: *What semantic web technologies, linked data, and connected data are; and why they are promising for structuring and visualising a connected data model of a natural stone restoration project*. Building Information Modelling (BIM) has emerged as a significant collaborative process in the Architecture, Engineering, and Construction (AEC) industry. BIM enables the creation of digital representations of building structures that are data-rich and object-oriented. This thesis introduces the concept of connected data by combining BIM and linked data. Linked data, which is based on semantic web technologies, facilitates the creation of knowledge graphs, and establishes relationships between data objects, provides context and enhances the understanding of information. By combining BIM principles, linked data and the specific requirements of heritage projects, data from various sources can be connected and structured according to the relevant standards in restoration projects. This approach can be considered as the next step towards advancing BIM maturity in the sector. This approach should enable professionals to make informed decisions, ensure the preservation of historical and cultural significance, and efficiently manage the complex challenges of restoring historical structures.

Subsequently, industry expert input was used to answer: *What questions arise in the current process of a natural stone restoration project*. To answer this question a three-step approach was used. First, the existing process on-site was analysed to determine the necessary steps and data generated throughout the process. The process of conserving natural stone facades consists of a series of tasks depending on the type of intervention to be executed, as illustrated in Figure 24. The resulting process schema was then used to discuss with stakeholders what information needs exist during the different stages of a natural stone restoration. These information challenges were used to establish eight competency questions. These questions were translated into three sets of different use cases that were used to develop the system. The developed use cases considered multiple types of users. The use cases for the beginner user focus on providing recognisable output, while also introducing new possibilities to the user. More advanced users are provided with functionalities for finding traditional information through the system. Lastly, professionals are granted the ability to create queries based on their extensive knowledge of the data structure.

Finally, a prototype system has been developed to address: *What a suitable workflow is for using a connected data model during a natural stone restoration project*. This started with the establishment of a conceptual system architecture that could provide the defined use cases, see Figure 26. This system architecture was developed into a prototype using the case project of the Monument on Dam Square. As sources from the case project were generated using a traditional process, some preparation was necessary before their use as input data. This included preparing multiple spreadsheets and creating an IFC model where every object in the Monument was represented by an individual IfcElement. Furthermore, a data structure was formed by utilising various existing ontologies that were identified in the literature study. This was supplemented by a newly created RII ontology that follows a modular and flexible structure recommended by members of the LBD-CG. This data structure was subsequently used to construct an RDF graph of the case project data. In addition, to make sure the several non-RDF files can be connected to the graph, they are made available through a local live server, giving every file a URI that is added to the graph. To demonstrate the possibilities of combining RDF and non-RDF data in the context of restoration projects, a front-end web application was developed that addressed the defined use cases. The resulting Heritage LBDviz expands the LBDviz and includes various additional features to address the use cases that exist in natural stone restoration projects. In addition to the Heritage LBDviz, the graph database was also used to automatically generate an enhanced version of the traditional spreadsheet used in the current process, demonstrating the new possibilities that arise from utilising connected data.

To conclude and answer the main research question: *What value can be created by using linked data technologies and BIM for structuring and visualising a connected data model during natural stone restoration projects on historical buildings?* To answer this question, the developed prototype system was validated with industry experts based on the identified competency questions in the process. To validate the proposed system, a specific object was selected in the case project. This object was used to reflect, together with industry experts, on the value of the created system in answering the questions of stakeholders. During the validation, it was concluded that the developed prototype already provides significant savings in time and minimises the likelihood of errors. Furthermore, it enhances awareness of data value, and its vendor-neutral approach that makes a project less reliant on a specific software.

7.2 CONTRIBUTION

This research has contributed to both scientific and societal aspects. Furthermore, there are some areas where the research may have fallen short in achieving its objectives or where its findings may not be entirely accurate. Both are discussed below.

7.2.1 Scientific Contribution

1 | **System Architecture**: The research proposed a conceptual system architecture that can provide the defined use cases for natural stone restoration projects. However, this system architecture can be used in a much wider range of contexts, such as other implementations of the LBDviz and other types of applications, even in other domains, as long as there is a need to connect, structure and visualise data.

2 | **Value of Prototype creation**: The research revealed the importance of building applications, such as the Heritage LBDviz. The validation of the prototype system demonstrated that displaying the combinations of data helps the industry experts understand the potential and value of creating connected data and helps in determine future use cases.

3 | **Intervention Process**: The process of interventions in conserving natural stone facades comprises several tasks, depending on the specific intervention to be performed. The process, however, is generalisable and is also suitable for interventions on other objects. For instance, repairing or temporarily storing a door.

4 | **Network of Ontologies**: The developed ontology confirms the value of the modular and flexible ontologies, which are part of the network developed by members of the LBD-CG. The used ontology patterns demonstrate the potential of using both SKOS concepts and XSD values as classes to name property instances, while retaining a structure that is easy to query. The same principle applies to the versioning of these properties. Adding a date to each property allows for filtering of the most recent version when querying.

5 | **LBDviz Platform**: This study has contributed to the continued investigation of the potential use of linked data technologies in combination with BIM-models within the AEC industry, more specifically the conservation of historical buildings. The created prototype system incorporates concepts from different research and implements them through a case project. The creation of the Heritage LBDviz tool demonstrates that the LBDviz can serve as a platform for developing applications that serve different contexts, proving the potential of introducing different sub-sectors of the AEC industry to the value of linked data.

7.2.2 Societal Contribution

1 | **Heritage LBDviz**: The development of the Heritage LBDviz tool has already demonstrated that the practical implementation of connecting BIM and linked data enables significant potential for time and error reduction. Additionally, the Heritage LBDviz can serve as a platform to connect more workflows in natural stone restoration projects and pursue one of many opportunities to improve efficiency.

2 | **Scalable System**: The developed system concentrated on the process of natural stone restoration projects. Nonetheless, the functionalities developed in the Heritage LBDviz have a broad applicability. Tracking objects during restoration projects is not exclusive to natural stone objects; numerous restoration projects involve objects that require external repairs or temporary storage, such as doors, statues, flooring, and many others. Furthermore, the system has been developed entirely using open standards and vendor-neutral solutions, allowing for future development as new use cases emerge.

3 | **Standardising Data Structures**: The addition to the network of modular ontologies can serve as a resource for standardising data structures in practice. This already provides value as the current restoration project will never be the last because these buildings need to be preserved.

4 | **Data-driven Workflows**: Access to standardised structured datasets of current and past restoration projects significantly increases the findability of data and provides many additional opportunities for cross-project analysis. Working based on standards will also help stakeholders transition from thinking in specific functionalities of commercial software to a workflow that centres on the necessary data.

7.2.3 Limitations

The following limitations are currently present in this thesis:

- The development of the data structure has focused on processes and standards in a Dutch built heritage context.
- The tool development was limited to data that was available in the case project or could be realistically reproduced.
- The development has focused on a specific type of project.
- The tool speed was limited by the performance of the JavaScript Comunica library.

7.3 RECOMMENDATIONS

The recommendations made can be divided in two categories: further research and implementation.

7.3.1 Further Research

- **Query Engine**: More research should be done to improve Comunica query engine performance. Response time could be enhanced through various methods. One approach to achieve this is by enhancing the Comunica engine to improve the speed at which it produces results. Secondly, optimising the data structure used to ensure that queries can be simplified and therefore executed faster. For example, automating the detection of the most recent task and incorporating an additional statement declaring it as such.
- **Expanding Use Cases**: As stated before, the results of Heritage LBDviz give users inspiration for other questions they would like answered, which should be explored in further research. For instance, cross-project functionalities could be created. This feature could be helpful in identifying projects that use the same materials, determining the damages that have occurred and gaining understanding into the effectiveness of the interventions used for these damages.
- **Validation with SHACL**: The existing prototype of the Heritage LBDviz and the spreadsheet offers some basic functionality for visual data validation. Nonetheless, the linked data technology SHACL has the capacity to automate numerous data validation processes. Thus, it is advisable that future research explores the use of SHACL to validate data as it is imported into the graph and during the process of validating the completion of tasks.
- **Inferring Knowledge**: With increased data connections, the possibilities of deriving knowledge from the graph should be explored. As more data is gathered about interventions, the system should be capable of assisting in the execution of these interventions. It could, for example, give suggestion on how to execute a certain intervention according to the relevant standards and based on previous data point to specific risks or points of attention.
- **Non-IFC models**: At moment, the tool only works with an IFC model, but future versions should support other open-source 3D model formats, such as a photogrammetry-based model.

7.3.2 Implementation

- **Business Case**: Before implementing the prototype into practice, a business case should be created to compare costs and benefits. The contractor created all the data used in the project. As soon as other stakeholders begin contributing to the data, one should consider data ownership. Furthermore, when considering the benefits one can look at different areas, the direct benefit of time and cost reduction during the project, but also the fact that this data can now be used for purposes in other projects and think about future business models where data could be provided as a service to clients.
- **Data Quality**: Data quality should always be considered when using the proposed system since poor data quality leads to inaccurate results ("garbage in, garbage out"). Poor quality data cannot produce useful results for a user, therefore data validation is essential.
- **Ongoing Development**: The system was developed using data from an existing project as input. Further optimisation through development cycles will be necessary to implement the system. Using the system in a pilot project may reveal new issues and provide useful insights. These will need to be addressed and integrated into the tool before it can be implemented on a larger scale.

Thank you for reading!

REFERENCES

Alshawabkeh, Y., Baik, A., & Miky, Y. (2021). Integration of laser scanner and photogrammetry for heritage BIM enhancement. *ISPRS International Journal of Geo-Information*, *10*(5). https://doi.org/10.3390/ijgi10050316

Antonopoulou, S., & Bryan, P. (2017). *BIM for Heritage* (pp. 1–78). Historic England. https://historicengland.org.uk/advice/technical-advice/recording-heritage/

Beerda, E. (2022, November 1). Restauratie Domtoren is vooral ook een mega-digitaliseringsklus. *Cobouw*. https://www.cobouw.nl/308713/restauratie-domtoren-is-vooral-ook-een-mega-digitaliseringsklus

Beetz, J., Van Leeuwen, J., & De Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, *23*(1), 89–101. https://doi.org/10.1017/S0890060409000122

Berners-Lee, T. (2006). *Linked Data*. https://www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, *284*(5), 34–43. https://doi.org/10.1038/SCIENTIFICAMERICAN0501-34

Bew, M., & Richards, M. (2008). BIM maturity model. *Construct IT Autumn 2008 Members' Meeting. Brighton, UK*.

Bonduel, M. (2021). *A Framework for a Linked Data-based Heritage BIM* [KU Leuven]. https://lirias.kuleuven.be/retrieve/618662

Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M., & Klein, R. (2019). Including widespread geometry formats in semantic graphs using RDF literals. *Proceedings of the 2019 European Conference on Computing in Construction*, *1*, 341–350. https://doi.org/10.35490/ec3.2019.166

Borrmann, A., Beetz, J., Koch, C., Liebich, T., & Muhic, S. (2018). Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models. In *Building Information Modeling: Technology Foundations and Industry Practice* (pp. 81–126). Springer International Publishing. https://doi.org/10.1007/978-3-319-92862-3_5

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building information modeling: Why? What? How? *Building Information Modeling: Technology Foundations and Industry Practice*, 1–24. https://doi.org/10.1007/978-3-319-92862-3_1/FIGURES/9

Brouwer, J. (2022). *Release SketchUp-IFC-Manager 4.3.1 · BIM-Tools/SketchUp-IFC-Manager*. GitHub. https://github.com/BIM-Tools/SketchUp-IFC-Manager/releases/tag/v4.3.1

Cyganiak, R., Wood, D., & Lanthaler, M. (2014). RDF 1.1 concepts and abstract syntax. In *W3C Recommendation*. https://www.w3.org/TR/rdf11-concepts/

Donkers, A., Yang, D., de Vries, B., & Baken, N. (2022). Semantic Web Technologies for Indoor Environmental Quality: A Review and Ontology Design. *Buildings 2022, Vol. 12, Page 1522*, *12*(10), 1522. https://doi.org/10.3390/BUILDINGS12101522

Donkers, A., Yang, D., de Vries, B., & Baken, N. (2023). A Visual Support Tool for Decision-Making over Federated Building Information. *20th International Conference, CAAD Futures 2023*, 485–500. https://doi.org/10.1007/978-3-031-37189-9_32

ERM. (n.d.). *ERM Richtlijnen: Kennisbron en houvast voor de praktijk - Stichting ERM*. Retrieved February 23, 2023, from https://www.stichtingerm.nl/subpagina-bij-richtlijnen-en-kennis

ERM. (2020a). *BRL ERB 3000: Erkend Restauratie Bouwbedrijf* . Stichting Erkende Restauratiekwaliteit Monumentenzorg (ERM). https://www.stichtingerm.nl/kennis-richtlijnen/brl3000

ERM. (2020b). *URL 2005: Inspecties van monumentale gebouwen*. Stichting Erkende Restauratiekwaliteit Monumentenzorg (ERM). https://www.stichtingerm.nl/kennis-richtlijnen/url2005

ERM. (2021a). *RESTAURATIELADDER en GEDRAGSCODE - doelen en achtergronden* -. https://www.stichtingerm.nl/doc/Restauratieladder en gedragscode - samenhang (210928).pdf

ERM. (2021b). *URL 4007: Steenhouwwerk*. Stichting Erkende Restauratiekwaliteit Monumentenzorg (ERM). https://www.stichtingerm.nl/kennis-richtlijnen/url4007

Franken, V., & Meijer, S. A. (2013). Sense of history : capturing and utilizing immaterial values for sustainable heritage protection. *Livinggreen Scientific Conference, Delft, The Netherlands, 19 April 2013*, *April*, 1–24. https://repository.tudelft.nl/islandora/object/uuid%3Ac75b42fa-fe80-4140-9323-e8fe1497f805

Hagedorn, P., & König, M. (2021). Rule-Based Semantic Validation for Standardized Linked Building Models. *Lecture Notes in Civil Engineering*, *98*, 772–787. https://doi.org/10.1007/978-3-030-51295-8_53/FIGURES/9

Hagedorn, P., Liu, L., König, M., Hajdin, R., Blumenfeld, T., Stöckner, M., Billmaier, M., Grossauer, K., & Gavin, K. (2023). BIM-Enabled Infrastructure Asset Management Using Information Containers and Semantic Web. *Journal of Computing in Civil Engineering*, *37*(1), 04022041. https://doi.org/10.1061/(asce)cp.1943-5487.0001051

Hamdan, A.-H. (2023). *Ein ontologiebasiertes Verfahren zur automatisierten Bewertung von Bauwerksschäden in einer digitalen Datenumgebung*. Technische Universität Dresden.

Hamdan, A.-H., Bonduel, M., & Scherer, R. J. (2019). An ontological model for the representation of damage to constructions. *CEUR Workshop Proceedings*, *2389*, 64–77. http://ceur-ws.org/Vol-2389/05paper.pdf

Hamdan, A.-H., Katranuschkov, P., & Scherer, R. J. (2021). Ontology based anamnesis and diagnosis of natural stone damage for retrofitting. *CEUR Workshop Proceedings*, *3081*, 8–19. http://ceur-ws.org/Vol-3081/01paper.pdf

Heinemann, H., & Naldini, S. (2018). The role of monumentenwacht: 40 years theory and practice in the Netherlands. *Innovative Built Heritage Models - Edited Contributions to the International Conference on Innovative Built Heritage Models and Preventive Systems, CHANGES 2017*, 107–116. https://doi.org/10.1201/9781351014793-13/ROLE-MONUMENTENWACHT-40-YEARS-THEORY-PRACTICE-NETHERLANDS-HEINEMANN-NALDINI

Hendriks, L. and van der Hoeve, J. (2009). *Guidelines for Building Archeological Research*. https://www.cultureelerfgoed.nl/publicaties/publicaties/2009/01/01/guidelines-for-building-archeological-research

ICOMOS. (2008). *Illustrated glossary on stone deterioration patterns* (V. Vergès-Belmin, Ed.). ICOMOS-ISCS. http://www.icomos.org/publications/monuments_and_sites/15/pdf/Monuments_and_Sites_15_ISCS_Glossary_Stone.pdf

ISO 19650-1:2018. (2018). In *ISO* (p. 36). https://www.iso.org/standard/68078.html

ISO/TC 59/SC 13. (2020). *ISO 21597-1:2020 - Information container for linked document delivery. Exchange specification. Part 1: Container* (p. 41). https://www.iso.org/standard/74389.html

J. Paul Getty Trust. (n.d.). *About the AAT*. Retrieved March 1, 2023, from https://www.getty.edu/research/tools/vocabularies/aat/about.html

Knublauch, H., & Kontokostas, D. (2017). *Shapes Constraint Language (SHACL)*. W3C Recommendation. https://www.w3.org/TR/shacl/

Lebo, T., Sahoo, S., & McGuinness, D. (2013). *PROV-O: The PROV Ontology*. https://www.w3.org/TR/2013/REC-prov-o-20130430/

Lubelli, B., Pottgiesser, U., Quist, W. J., & Rextroth, S. (2021). *Dealing with Heritage: Assessment and Conservation* (S. Naldini, Ed.). BK Books. https://research.tudelft.nl/en/publications/dealing-with-heritage-assessment-and-conservation

Murphy, M., Mcgovern, E., & Pavia, S. (2009). Historic building information modelling (HBIM). *Structural Survey*, *27*(4), 311–327. https://doi.org/10.1108/02630800910985108

Naldini, S., Heinemann, H., & van Hees, R. (2018). Monumentenwacht and preventive conservation: Changes. *Innovative Built Heritage Models - Edited Contributions to the International Conference on Innovative Built Heritage Models and Preventive Systems, CHANGES 2017*, 117–124. https://doi.org/10.1201/9781351014793-14/MONUMENTENWACHT-PREVENTIVE-CONSERVATION-CHANGES-NALDINI-HEINEMANN-VAN-HEES

Naldini, S., & Hunen, M. van. (2019). Guidelines for quality of interventions in built cultural heritage in The Netherlands. In *Professionalism in the Built Heritage Sector* (pp. 87–93). CRC Press. https://doi.org/10.1201/9780429397912-11

Naldini, S., van Hees, R. P. J., & van der Grijp, E. (2021). Integrated Conservation Strategies in the Netherlands. In P. Roca, L. Pelà, & C. Molins (Eds.), *12th International Conference on Structural Analysis of Historical Constructions* (pp. 1583–1593). International Centre for Numerical Methods in Engineering, CIMNE. https://doi.org/10.23967/sahc.2021.137

NEN 2660-2. (2022). https://connect.nen.nl/Standard/Detail/3665979?compId=0&collectionId=0

NEN 2767-1+C1. (2019). https://connect.nen.nl/Standard/Detail/3611437

NEN-EN 12440. (2017). https://www.nen.nl/nen-en-12440-2017-en-240542

Oraskari, J., Bonduel, M., McGlinn, K., Pauwels, P., Freddy, P., Anna, W., Kukkonen, V., Steyskaland, S., & Lehtonen, J. (2023). *IFCtoLBD*. https://github.com/jyrkioraskari/IFCtoLBD

Pauwels, P., Costin, A., & Rasmussen, M. H. (2022). Knowledge Graphs and Linked Data for the Built Environment. In *Industry 4.0 for the Built Environment* (Vol. 20, pp. 157–183). Springer, Cham. https://doi.org/10.1007/978-3-030-82430-3_7

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, *63*, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003

Plasterk, R., & Cramer, J. (2009). *Beleidsbrief Modernisering van de Monumentenzorg (MoMo)*. https://www.tweedekamer.nl/kamerstukken/detail?id=2009Z17463&did=2009D46060

Preidel, C., Borrmann, A., Mattern, H., König, M., & Schapke, S. E. (2018). Common data environment. In *Building Information Modeling: Technology Foundations and Industry Practice* (pp. 279–291). Springer International Publishing. https://doi.org/10.1007/978-3-319-92862-3_15

Rasmussen, M. H., Bonduel, M., Hviid, C. A., Lefrançois, M., & Karlshø, J. (2018). *OPM: An ontology for describing properties that evolve over time*. https://www.researchgate.net/publication/326930517

Rasmussen, M. H., Lefrançois, M., Schneider, G. F., & Pauwels, P. (2021). BOT: The building topology ontology of the W3C linked building data group. *Semantic Web*, *12*(1), 143–161. https://doi.org/10.3233/SW-200385

Rasmussen, M. H., Pauwels, P., Hviid, C. A., & Karlshøj, J. (2017). Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*, 237–244. https://doi.org/10.24928/JC3-2017/0153

RCE. (n.d.-a). *BESTEKSTERMEN voor. A. Natuursteensoorten.pdf*.

RCE. (n.d.-b). *Zoeken in het Rijksmonumentenregister*. Retrieved March 3, 2023, from https://monumentenregister.cultureelerfgoed.nl/

RCE Erfgoedthesaurus. (n.d.). Retrieved March 1, 2023, from https://thesaurus.cultureelerfgoed.nl/

Rocha, G., Mateus, L., Fernández, J., & Ferreira, V. (2020). A scan-to-bim methodology applied to heritage buildings. *Heritage*, *3*(1), 47–65. https://doi.org/10.3390/heritage3010004

Scheffer, M., Mattern, H., & König, M. (2018). BIM project management. In *Building Information Modeling: Technology Foundations and Industry Practice* (pp. 235–249). Springer International Publishing. https://doi.org/10.1007/978-3-319-92862-3_13

Seeaed, M. K., & Hamdan, A.-H. (2019). BIMification of stone walls for maintenance management by utilizing Linked Data. *31st Forum Bauinformatik*, 165–172. https://www.researchgate.net/publication/337104176_BIMification_of_stone_walls_for_maintenance_management_by_utilizing_Linked_Data

Senthilvel, M., Oraskari, J., & Beetz, J. (2020). Common Data Environments for the Information Container for linked Document Delivery. *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2020* , 132–145. https://www.w3.org/community/lbd/

SKOS | Forum Standaardisatie. (n.d.). Retrieved May 8, 2023, from https://www.forumstandaardisatie.nl/open-standaarden/skos

SKOS Simple Knowledge Organization System Reference. (n.d.). Retrieved May 8, 2023, from https://www.w3.org/TR/skos-reference/

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, *25*(1–2), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

TNO, TU Delft, & RCE. (n.d.). *MDCS - Damage Atlas*. Retrieved March 1, 2023, from https://mdcs.monumentenkennis.nl/damageatlas

Torre, S. della, & Moioli, R. (2021). Chapter 15: Cultural heritage at work for economy and society. In *A Research Agenda for Heritage Planning: Perspectives from Europe* (pp. 175–185). Edward Elgar Publishing Ltd. https://doi.org/10.4337/9781788974639.00028

van Berlo, L., Beetz, J., Bos, P., Hendriks, H., & van Tongeren, R. C. J. (2012). Collaborative engineering with IFC: New insights and technology. *EWork and EBusiness in Architecture, Engineering and Construction - Proceedings of the European Conference on Product and Process Modelling 2012, ECPPM 2012*, 811–818. https://research.tue.nl/en/publications/collaborative-engineering-with-ifc-new-insights-and-technology

van den Bersselaar, E. (2022). *Automatic Validation of Technical Requirements for a BIM model using Semantic Web Technologies* [Eindhoven University of Technology]. https://research.tue.nl/en/studentTheses/091293fd-b928-4fdf-9bcf-078ba8bae594

van der Hall, B. (2023). *Digital building logbook for improved asset management: The value of a linked data approach* [Eindhoven University of Technology]. https://research.tue.nl/en/studentTheses/digital-building-logbook-for-improved-asset-management

van der Horst, A., Nauta, O., Nijboer, R., & Weemstra, J. (2022). *Beleidsdoorlichting Erfgoed - Eindrapport*. https://zoek.officielebekendmakingen.nl/blg-1050587

van der Pas, C. (2022). *Achieving the long-term preservation of building data via linked data combined with durable data storage measures* [Eindhoven University of Technology]. https://research.tue.nl/en/studentTheses/achieving-the-long-term-preservation-of-building-data-via-linked-

van Hees, R. P. J., & Naldini, S. (2020). MDCS - A system for damage identification and monitoring. *Preventive Conservation - From Climate and Damage Monitoring to a Systemic and Integrated Approach - Proceedings of the International WTA - PRECOM3OS Symposium*, 113–118. https://doi.org/10.1201/9781003004042-17

Vandesande, A., & van Balen, K. (2016). An operational preventive conservation system based on the monumentenwacht model. *Structural Analysis of Historical Constructions: Anamnesis, Diagnosis, Therapy, Controls - Proceedings of the 10th International Conference on Structural Analysis of Historical Constructions, SAHC 2016*, 217–224. https://doi.org/10.1201/9781315616995-28/OPERATIONAL-PREVENTIVE-CONSERVATION-SYSTEM-BASED-MONUMENTENWACHT-MODEL-VANDESANDE-VAN-BALEN

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. In *Automation in Construction* (Vol. 38, pp. 109–127). https://doi.org/10.1016/j.autcon.2013.10.023

W3C. (2013, March 21). *SPARQL 1.1 Overview*. W3C Recommendation. https://www.w3.org/TR/sparql11-overview/

Wagner, A., Bonduel, M., Pauwels, P., & Rüppel, U. (2020). Representing construction-related geometry in a semantic web context: A review of approaches. *Automation in Construction*, *115*, 103130. https://doi.org/10.1016/j.autcon.2020.103130

Wagner, A., Bonduel, M., Pauwels, P., & Uwe, R. (2019). Relating geometry descriptions to its derivatives on the web. *Proceedings of the 2019 European Conference on Computing in Construction*, *1*, 304–313. https://doi.org/10.35490/EC3.2019.146

Wagner, A., Bonduel, M., Werbrouck, J., & McGlinn, K. (2022). Geometry and geospatial data on the web. In P. Pauwels & K. McGlinn (Eds.), *Buildings and Semantics* (pp. 69–99). CRC Press. https://doi.org/10.1201/9781003204381-5

Werbrouck, J., Pauwels, P., Beetz, J., & van Berlo, L. (2019). Towards a decentralised common data environment using linked building data and the solid ecosystem. *Advances in ICT in Design, Construction and Management in Architecture, Engineering, Construction and Operations (AECO) : Proceedings of the 36th CIB W78 2019 Conference*, *September*, 113–123. http://hdl.handle.net/1854/LU-8633673

Werbrouck, J., Pauwels, P., Bonduel, M., Beetz, J., & Bekers, W. (2020). Scan-to-graph: Semantic enrichment of existing building geometry. *Automation in Construction*, *119*, 103286. https://doi.org/10.1016/j.autcon.2020.103286

Zijlstra, H., Nijhuis, S., Quist, W. J., Zijlstra, H., Quist, W. J., Thissen, P., Pegels, T., Nijhuis, S., Naldini, S., Lubelli, B., Ridder, A. C. de, Remøy, H., Clarke, N., Dobbelsteen, A. A. J. F. van den, Gommans, L. J. J. H. M., Dang, M. K., & Phoelich, L. (2021). *Karakteristiek Duurzaam Erfgoed in Gelderland: KaDEr-stellingen: Ervaringen en stellingnames na een vierjarige samenwerking tussen de Provincie Gelderland en de TU Delft faculteit Bouwkunde 2017 - 2021*. TU Delft Open. https://research.tudelft.nl/en/publications/karakteristiek-duurzaam-erfgoed-in-gelderland-kader-stellingen-er

APPENDICES

APPENDIX I: PREFIXES AND NAMESPACES

Prefix	*Namespace*
bhp	https://w3id.org/con-tax/properties/bhp#
bot	https://w3id.org/bot#
contax	https://w3id.org/con-tax#
cp	https://w3id.org/con-tax/properties/cp#
cto	https://w3id.org/cto#
dot	https://w3id.org/dot#
exp	*A fictive namespace for example data*
fog	https://w3id.org/fog#
inst	*A fictive namespace for case project data* | https://nicodebont.nl/
mdcs	https://mdcs.monumentenkennis.nl/damageatlas/ontology#
omg	https://w3id.org/omg#
ont	*A fictive namespace for example ontology*
owl	http://www.w3.org/2002/07/owl#
prov	http://www.w3.org/ns/prov#
qudt	http://qudt.org/2.0/schema/qudt#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
rii	*A fictive namespace for case project ontology* | https://janssensteenberg.com/rii#
schema	http://schema.org/
skos	http://www.w3.org/2004/02/skos/core#
unit	http://qudt.org/vocab/unit/
xsd	http://www.w3.org/2001/XMLSchema#

APPENDIX II: HERITAGE LBDVIZ

Complete code can be found on GitHub:

https://github.com/MarijnJanssenSteenberg/Heritage-LBDviz

APPENDIX III: SPARQL QUERY - SPREADSHEET EXPORT

```
PREFIX bot: <https://w3id.org/bot#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX nen2660: <https://w3id.org/nen2660/def#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rii: <https://janssensteenberg.com/rii#>
PREFIX omg: <https://w3id.org/omg#>
PREFIX schema: <http://schema.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX cto: <https://w3id.org/cto#>
PREFIX dot: <https://w3id.org/dot#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX fog: <https://w3id.org/fog#>

Select Distinct ?LBDviz ?Building ?Storey ?label ?Shape ?Length_Existing
?Thickness_Existing ?Width_Existing ?Volume_Existing ?Material_Existing ?Length_New
?Thickness_New ?Width_New ?Volume_New ?Material_New ?LastChangedTask (GROUP_CONCAT
(DISTINCT ?Photo_Existing; SEPARATOR="; ") AS ?Photos_Existing) (GROUP_CONCAT (DISTINCT
?Photo_Repaired; SEPARATOR="; ") AS ?Photos_Repaired)
WHERE {
    ?ifcElement rdf:type bot:Element , nen2660:FunctionalEntity ;
                rdfs:label ?label ;
                fog:hasIfcId-guid ?LBDviz .

    ?building a bot:Building ;
              rdfs:label ?Building ;
              bot:hasElement ?ifcElement.
    ?storey a bot:Storey ;
            rdfs:label ?Storey ;
            bot:hasElement ?ifcElement.

    {
    SELECT ?ifcElement ?label ?Shape ?Length_Existing ?Thickness_Existing ?Width_Existing
?Volume_Existing ?Material_Existing ?Photo_Existing ?Photo_Repaired
        {
        ?ifcElement nen2660:isImplementedBy ?elementExisting .
        ?elementExisting rdfs:label ?label .

        ?elementExisting rdf:type bot:Element , nen2660:TechnicalEntity ;
                rdfs:label ?label .

        ?elementExisting
                rii:hasShape ?shape ;
                rii:hasMaterial ?material_Existing .

        ?material_Existing rdfs:label ?Material_Existing .
        ?shape rdfs:label ?Shape .

        ?elementExisting
                rii:hasLength ?length_Existing ;
                rii:hasThickness ?thickness_Existing ;
                rii:hasWidth ?width_Existing .

        ?length_Existing <https://schema.org/value> ?Length_Existing .
        ?thickness_Existing <https://schema.org/value> ?Thickness_Existing .
        ?width_Existing <https://schema.org/value> ?Width_Existing .
        BIND((?Length_Existing * ?Thickness_Existing * ?Width_Existing)/1000000000 as
?Volume_Existing)
```

```
        ?elementExisting rii:hasDescriptionInResource ?photo_Existing .
        ?photo_Existing rii:resourceCapturedDuringTask ?inspection ;
            rii:filePath ?Photo_Existing .
        ?inspection a cto:InspectionTask .

        ?elementExisting rii:hasDescriptionInResource ?photo_Existing .
        ?photo_Existing rii:resourceCapturedDuringTask ?inspection ;
            rii:filePath ?Photo_Existing .
        ?inspection a cto:InspectionTask .

        ?elementExisting rii:hasDescriptionInResource ?photo_Repaired .
        ?photo_Repaired rii:resourceCapturedDuringTask ?repair ;
            rii:filePath ?Photo_Repaired .
        ?repair a cto:RepairTask .
        }
    }
    UNION
    {
    SELECT ?ifcElement ?label ?Shape ?Length_Existing ?Thickness_Existing ?Width_Existing
?Volume_Existing ?Material_Existing ?Length_New ?Thickness_New ?Width_New ?Volume_New
?Material_New ?Photo_Existing ?Photo_Repaired
        {
        ?ifcElement nen2660:isImplementedBy ?elementNew .
        ?elementNew rdfs:label ?label ;
                    rdfs:comment "Nieuw"@nl ;
                    rii:replacesElement ?elementExisting .

        ?elementExisting
                rii:hasShape ?shape ;
                rii:hasMaterial ?material_Existing .

        ?material_Existing rdfs:label ?Material_Existing .
        ?shape rdfs:label ?Shape .

        ?elementExisting
            rii:hasLength ?length_Existing ;
            rii:hasThickness ?thickness_Existing ;
            rii:hasWidth ?width_Existing .

        ?length_Existing <https://schema.org/value> ?Length_Existing .
        ?thickness_Existing <https://schema.org/value> ?Thickness_Existing .
        ?width_Existing <https://schema.org/value> ?Width_Existing .
        BIND((?Length_Existing * ?Thickness_Existing * ?Width_Existing)/1000000000 as
?Volume_Existing)

        Optional{
            ?elementNew rii:hasMaterial ?material_New .
            ?material_New rdfs:label ?Material_New .
        }
        Optional{
        ?elementNew
            rii:hasLength ?length_New ;
            rii:hasThickness ?thickness_New ;
            rii:hasWidth ?width_New .

        ?length_New <https://schema.org/value> ?Length_New .
        ?thickness_New <https://schema.org/value> ?Thickness_New .
        ?width_New <https://schema.org/value> ?Width_New .
        BIND((?Length_New * ?Thickness_New * ?Width_New)/1000000000 as ?Volume_New)

        ?elementExisting rii:hasDescriptionInResource ?photo_Existing .
        ?photo_Existing rii:resourceCapturedDuringTask ?inspection ;
            rii:filePath ?Photo_Existing .
        ?inspection a cto:InspectionTask .
```

```
        }
        }
    }

    Optional{

    { ?elementExisting cto:isSubjectOfTask ?task . }
    UNION
    { ?elementNew cto:isSubjectOfTask ?task .  }

    ?task rdfs:label ?LastChangedTask ;
           prov:startedAtTime|prov:endedAtTime ?DateTime .

    {
    SELECT  ?label (MAX(?dateTime) AS ?maxDateTime)
        WHERE {
            ?element cto:isSubjectOfTask ?task ;
                     rdfs:label ?label .
            ?task prov:startedAtTime|prov:endedAtTime ?dateTime .
        }
    GROUP BY ?label
    }

    FILTER (?DateTime = ?maxDateTime)
    }

}
Group By ?LBDviz ?elementExisting ?Building ?Storey ?label ?Shape ?Length_Existing
?Thickness_Existing ?Width_Existing ?Volume_Existing ?Material_Existing ?Length_New
?Thickness_New ?Width_New ?Volume_New ?Material_New ?LastChangedTask
Order By ?Storey ?label
```

[This page is intentionally left blank]

For the future of our **heritage,** preserving **information** is as important as the conservation of the **monument** itself

Contact

linkedin.com/in/marijn-janssen-steenberg

marijn@janssensteenberg.com

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY