Slava Ivanov

As-Built Modelling in Civil Construction

Metropolia University of Applied Sciences

Master of Engineering

Computing in Construction

Master’s Thesis

10 December 2025

Abstract

Author:	Slava Ivanov
Title:	As-Built Modelling in Civil Construction
Number of Pages:	141 pages + 0 appendices
Date:	10 December 2025
Degree:	Master of Engineering
Degree Programme:	Computing in Construction
Professional Major:	Name of the professional major
Supervisors:	Seppo Törmä, Principal Lecturer Mikail Kesenci, Project Manager Jari Pelkonen, BIM Manager

The purpose of this research was to explore the immediate applicability of the as-built models produced for Crown Bridges Light Rail civil construction project. This study combined practical applicability assessments with exploratory activities that employed modern technologies such as the Internet of Things and Artificial Intelligence. The as-built dataset used in this study was prepared in both LandXML and IFC formats, covering the primary open BIM formats used in civil construction in Finland. Major part of this research was dedicated to examining as-built models in terms of their readability across various platforms, including modelling, viewing, and construction site management applications. Additional research components included models' suitability for structure gauge analysis, their integration with sensor frameworks, their potential applications in machine learning workflows, and their compliance with relevant regulatory provisions. The analytical methods applied in this research included qualitative, comparative and exploratory approaches. The results demonstrated strong performance of the evaluated as-built deliverables across all of the above listed disciplines, while also identifying features requiring additional development and standardisation. This study provides applicable information on the value of as-built models for design, construction and maintenance professionals, as well as lays ground for future research and development.

Keywords: as-built model, BIM, digital twin, light rail, AI, infrastructure, design, construction, maintenance, land surveying, LandXML, IFC

Contents

List of Abbreviations

AI	Artificial Intelligence
AIM	Asset Information Model
API	Application Programming Interface
ATU	Structure Gauge (finn. Aukean Tilan Ulottuma)
BIM	Building Information Modelling
BREP	Boundary Representation
bSF	buildingSMART Finland
CAD	Computer-Aided design
CDE	Common Data Environment
CSG	Constructive Solid Geometry
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GT	GeoTime
HTML	HyperText Markup Language
IDS	Information Delivery Specification

IFC	Industry Foundation Classes
InfraBIM	Survey code list for surface layers
InfraRAK	Survey code list for elements (finn. Infra Rakentajakoodaus)
IM	Inframodel
IoT	Internet of Things
ISO	International Organisation for Standardisation
JS	JavaScript
JSON	JavaScript Object Notation
LAS	Laser
LiDAR	Light Detection and Ranging
LLM	Large Language Models
LOF	Local Outlier Factor
LOIN	Level of Information Need
MQTT	Message Queuing Telemetry Transport
PIM	Project Information Model
RAG	Retrieval Augmented Generation
RTK	Real Time Kinematic
RYHTI	Built Environment Information System (finn. Rakennetun ympäristön tietojärjestelmä)

RYTV	Built Environment Information Modelling Standardisation (finn. Rakennetun ympäristön tietomallintamisen vakiointi)
STEP	Standard for the Exchange Product model data
TIN	Triangulated Irregular Network
TS	TypeScript
URL	Uniform Resource Locator
XML	eXtensive Markup Language
YIV	Common InfraBIM requirements (finn. Yleiset Inframallivaatimukset)

1 Introduction

1.1 Study Background

The commissioning organisation and the primary beneficiary of this research is the Crown Bridges Light Rail project. It is one of the largest and most prominent infrastructure construction works currently underway in Finland, stretching across the capital city coastline and comprising approximately 400,000m^2 of newly built environment. The constructed assets include 15.5km of single-track rail, five bridges, 13 light rail stations, 430 poles and 300km of utility lines. A schematic view of the light rail route is shown in figure 1.

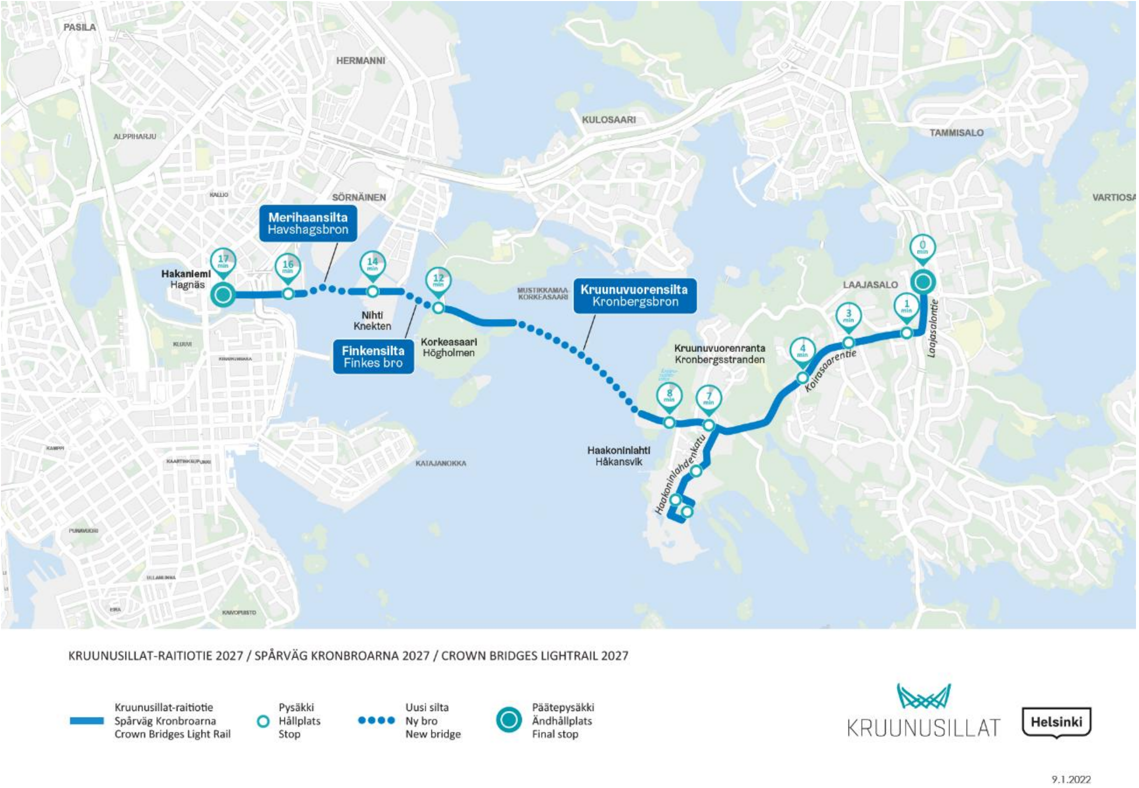


Figure 1. Crown Bridges Light Rail route [1]

Because of its spatial scope and large element quantities, the project carries significant urban impact. It is designed to ensure smoother traffic flow across Helsinki for the steadily growing population. In addition to the light rail network, the projects contribute substantially to the development of cycling and pedestrian routes, which, in turn, support Finland’s carbon neutrality goals. [1.]

At one of the initial phases of the project, the owners made a strategic decision to produce a 3D as-built model of the new structures, including both above and below-ground components. In earlier infrastructure projects of similar description, as-built documentation relied largely on 2D elements, with little to no metadata, mainly due to shortage of relevant experience and the absence of task-specific software solutions. Thus, Crown Bridges became the first large-scale infrastructure project in Finland to digitally replicate its as-built state in three dimensions. The successful completion of such an extensive modelling effort is expected to provide a robust foundation for further digitalisation of the city infrastructure. [2.]

The Crown Bridges project aims to enhance the existing workflows for model-based data governance generating insights that benefit the whole industry. The lessons learned and practices developed within this project have the potential to improve modelling practices in other infrastructure construction projects, regardless of their scale or complexity. Potential beneficiaries of this research include civil design professionals, production engineers, project managers, and facility operators. [2.]

1.2 Research Perspectives and Questions

The research perspectives reflect recognised gaps in Building Information Modelling (BIM) for infrastructure. They range from immediate challenges, such as initial data quality and model-based maintenance, to emerging needs, including legal governance, Digital Twins (DT) and Artificial Intelligence (AI)-driven workflows.

1.2.1 Initial Data Perspective

In order to design a piece of new civil infrastructure or plan the demolition of an existing one, specialists require a reliable and comprehensive initial data set. The managerial consortium of Crown Bridges Light Rail identified the applicability of as-built models as initial data as a priority research question. [2.]

To provide access to high-quality initial data, reduce design lead times, and minimise the need for corrective actions as-built models must be compatible with design and sketching software. Therefore, this research investigates whether the as-built models produced for Crown Bridges are compatible with the domain-specific design and viewing platforms, and what is the minimum accepted level of information for such models.

1.2.2 Operation and Maintenance Perspective

Once commissioned, an asset enters the operation phase, during which as-built models should be transitioned into model-based maintenance and operation. While the model-based approach to maintenance and operation has already been successfully implemented in various facilities, its use in civil infrastructure, such as roads, railways and utility lines, remains limited.

Confirming that as-built models are compatible with asset management platforms would facilitate an easier transition to model-based asset management. Consequently, this study examines whether the as-built models of Crown Bridges are suitable for operation and maintenance applications, and what is required to keep these models relevant and up to date after commissioning.

1.2.3 Regulatory Compliance Perspective

Recent advancements in Finnish legislation have introduced as-built model references in the legislation. Although the current definitions remain relatively broad, ensuring compatibility at an early stage is essential to prepare for more detailed requirements in future regulations. In addition to national law, standards and local regulations provide more specific definitions.

Verifying the compliance of as-built models with these frameworks is crucial for their validity in contexts involving higher degrees of legal obligation. Therefore, this study evaluates whether the Crown Bridges models conform to relevant legal provisions, industry standards, and whether they are capable of scaling to meet future requirements.

1.2.4 Auxiliary Applications Perspective

Beyond conventional applications, as-built models offer opportunities for less explored but highly promising ways to use the data, that could bring substantial benefits at both project and industry levels. One of them is structure gauge analysis, which is critical for safe operation of rolling stock but is still often performed manually. A model-based alternative would reduce costs and could be performed simultaneously with the construction, leaving time for corrective actions. Another relevant example is the digital twin. Although its definitions vary, it is typically understood as a 3D model view combined with sensor data streams. Digital twin solutions are particularly valuable during the maintenance and operation phase, though they may also enhance other phases of the asset’s life cycle. A third potential application is AI integration. Combining models with artificial intelligence algorithms can push process automation to a completely new level and help professionals address other emerging challenges.

Therefore, it is beneficial to investigate whether the Crown Bridges as-built models are suitable for the structure gauge examination, whether they qualify as digital twin or align with digital twin frameworks, and whether they can be effectively integrated into AI-driven workflows. Taken together, these perspectives indicate that the value of as-built models is not confined to conventional use cases but extends into innovative domains with significant long-term potential.

1.3 Structure and Scope

This thesis is organised into five chapters. The structure follows the research process, with the first chapter outlining the context of the host project and the research questions, and the final chapter presenting the conclusions, explicitly linking them back to the study objectives. Chapters 2,3 and 4 focus on the core study and thereby deserve a more detailed explanation.

Chapter 2 is dedicated to presenting the existing body of knowledge that directly or indirectly influences as-built modelling in Finland. It approaches the subject from both academic and technological perspectives. The academic perspective includes key definitions and applicable regulations, both mandatory and advisory. Technological perspective, in turn, describes existing technologies, applications, and data exchange formats.

Chapter 3 reviews previously done work. The selected reference case was deliberately chosen from a project similar to Crown Bridges, enabling direct comparison and highlighting potential workflow improvements.

Chapter 4 presents the research activities undertaken in this thesis in a step-by-step format. The research is categorised by model applications, progressing from the most conventional to the least explored. Each subchapter presents the data used, the selected framework, and the drawn conclusions.

The scope of this research is defined by the host project - The Crown Bridges. The project’s as-built models are predominantly land survey-based, although a fraction of them is derived from as-designed models. This distribution reflects the practical finding that land survey-based modelling is generally faster and easier to manage. In contrast, as-designed models often came with challenges related to format conversion, or schema updates. This research does not intend to compare as-built modelling workflows or software combinations, but rather to evaluate the existing as-built deliverables in ways directly relevant to the project.

Although this research addresses as-built models from a range of perspectives, it does not examine issues related to information security. Following careful consideration, it was concluded that information security practices are too substantial a topic to be treated as supplementary research here and instead deserve their own dedicated thesis.

The contribution of this thesis is twofold. First, it provides one of the first detailed analyses of infrastructure as-built models from multiple perspectives. Second, it develops and validates workflows that can be directly applied by project organisations and asset owners.

2 Theoretical Background and Existing Frameworks

2.1 Definitions of the Key Concepts

2.1.1 BIM

Understanding of the Building Information Modelling concept may vary depending on a country, organisation or construction domain; therefore, it is advised to always include the definition relevant to a specific project, organisation or research context [3]. The BIM dictionary offers a following definition of BIM: “a set of technologies, processes and policies enabling multiple stakeholders to collaboratively design, construct and operate a facility in virtual space” [4].

A frequent misconception is the association of BIM with a certain type of software or even hardware, such as land surveying tools. In reality, BIM is rather seen as a human-driven activity than a technological product. [5.]

Another factor contributing to the complexity of the BIM concept is the notion of BIM dimensions. As shown in figure 2, ten dimensions have currently been proposed. While 1D, 2D, and 3D dimensions are self-explanatory, higher dimensions require elaboration. The fourth modelling dimension (4D) refers to the 3D model integration with time and schedule data, and the fifth dimension (5D) stands for 4D combined with cost and budget information. Other BIM dimensions, such as facility management, sustainability, safety and lean construction are lacking international consensus and have not yet been widely adopted. [6.]

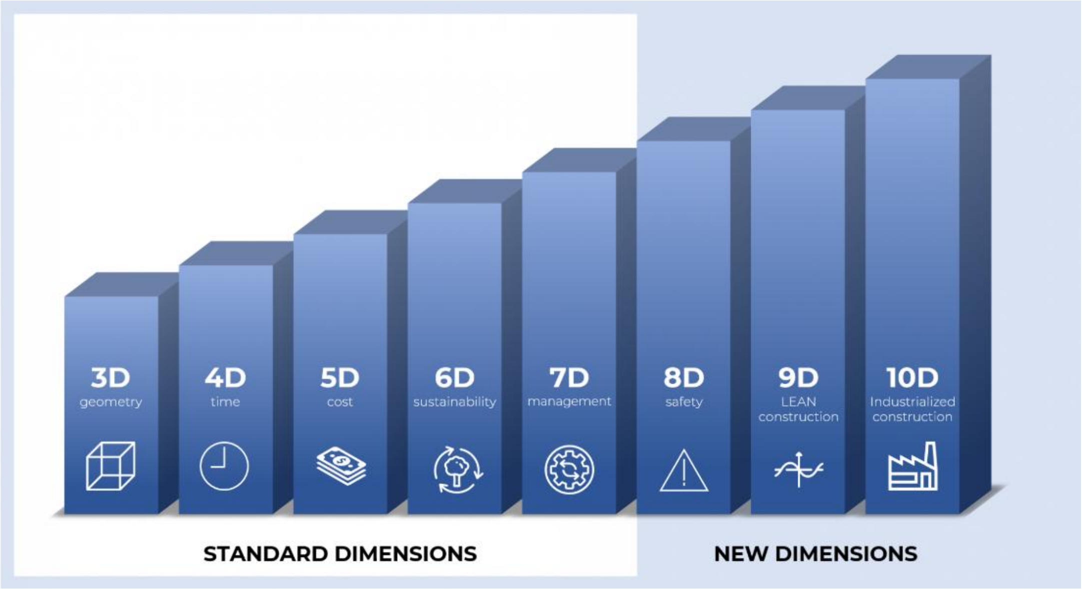


Figure 2. BIM Dimensions [6]

Within the scope of this thesis, BIM is understood as a human intelligence-driven process that involves creating and managing digital representations of real-world assets, whether they are planned, existing or demolished. BIM generally comprises three areas of activity.

The first area concerns model creation, which typically involves the development of initial drawings, shaping of volumetric geometries and integration of architectural, structural and engineering details. These tasks rely firmly on design software capabilities, sketching and sculpting instruments, parametric and computational design functionalities and the ability to render complex datasets.

The second area relates to model management, which may contain design coordination, model updating, validation, implementation, and distribution. These activities are often associated with the construction phase of an asset and depend on a variety of technological solutions. Model checking and validation require robust quality control algorithms and may occur in isolated stationary environments, whereas implementation and distribution require interaction with the on-site people and machinery.

The third area is model-based work, which may involve collaboration and decision making, construction and maintenance operations, progress tracking and situational awareness. These applications may not require modelling at all but are heavily influenced and supported by existing models. Properly modelled data enables efficient communication, task management, cost and schedule control, risk assessment.

2.1.2 As-Built Model

In this thesis, as-built modelling is approached from a civil construction perspective and is understood as a process of capturing the geographical, geometrical and metadata properties of the asset at the time of construction. BuildingSMART Finland defines as-built model as “an infrastructure information model that describes an infrastructure or system as it is actually constructed when entity-specific quality requirements are taken into account” [7]. Put simply, an as-built model must represent the asset as it is, once the required quality has been confirmed. An as-built model is essentially a BIM model, since as-built modelling is a part of Building Information Modelling domain [8].

One of the known benefits derived from as-built modelling is improved quality assurance, achieved when captured data represents the asset as constructed and confirms that its properties conform to the structural, geographical, and legal requirements [7]. Rendering data into a BIM model enables users to compare as-built condition against as-designed models and drawings.

Other key benefits are improved collaboration and enhanced risk management. As-built data supports situational awareness within the project, helps track progress and monitor schedule and costs. Detailed visualisations of the built structures support collaboration between on-site and off-site personnel and assist decision-making [9]. Moreover, they allow deviations to be identified earlier, reducing the likelihood of adverse events in later project phases and strengthening the overall risk management process.

As-Built information modelling is a relatively recent phenomenon, and work practices in this area are not yet fully established. Active ongoing research is constantly affecting the industry, therefore the as-built modelling workflow may vary depending on the region or the project [3]. From a technology perspective, as-built model production may require advanced solutions in order to reach the desired level of detail and precision. For example, land surveying, aerial photography, laser scanning, and machine automation are widely implemented in the as-built domain.

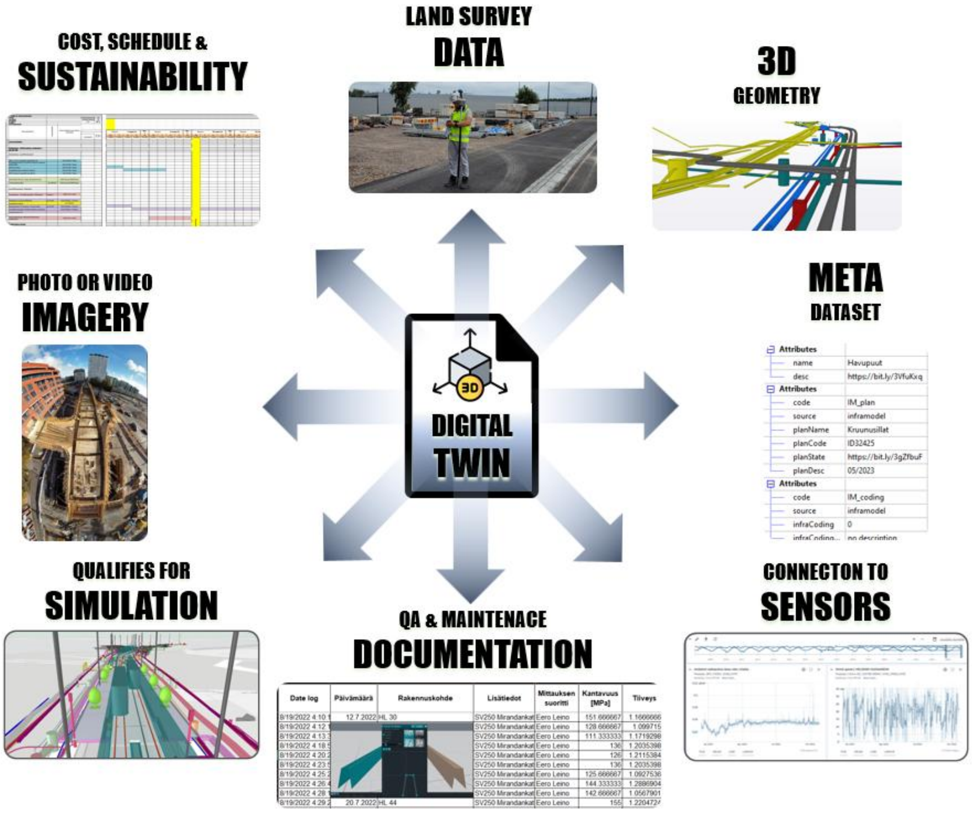
2.1.3 Digital Twin

The Digital Twin can be defined as “the information construct of the Physical Twin. The intent of the concept is that it can provide the same or better information than could be obtained by being in physical possession of the Physical Twin. The key assumption is that the type, granularity, and amount of information contained in the Digital Twin is driven by use cases” [10].

The original term was initially introduced by NASA in 2010 and adopted in the aerospace sector, although the concept itself originated in 2002 as a part of Product Lifecycle Management programme [11]. In the construction industry, DT is an emerging trend, attracting considerable global interest. While some suggest that a plain BIM model qualifies as a digital twin, others argue that the model must contain real-time sensor data - such as for example weather or occupancy information - to meet the definition. In reality, none of the currently available products can be classified as an actual digital twin [3].

Moreover, there are disagreements regarding the essence of a digital twin, which may lead to a misunderstanding of how it is used in this research. One popular opinion implies that a DT is actually a tool - either software, or a combination of software and hardware, such as a control board – that is remotely connected to the asset but does not necessarily require any 3D visualisation. Examples may include an indoor lighting control application or even Google Maps with its ability to display real-time traffic conditions. Another group of people sees digital twin as a computational algorithm or a process workflow that mirrors the state and condition of the asset, while keeping virtual and physical assets bilaterally independent [12]. Other experts state that the word Twin can only refer to an identical copy, and anything other than that should be named a digital replica but not a twin whatsoever [3].

For the purpose of this research, a digital twin is addressed from the civil construction and asset management perspective and is conceptualised as a composition of machine-readable information layers that represent an infrastructural asset in both a visual and a schematic way. Visual representation includes confirmed precise geospatial positioning and validated 3D geometry, suitable for simulation purposes, while schematic representation includes entity identification data, information on implemented materials, schedule records, a bill of quality assurance and potential connections to on-site sensors. Figure 3 illustrates the DT content structure as used in this research.



Attributes	
name	Havupuut
desc	https://bit.ly/3VfuKxq
Attributes	
code	IM_plan
source	inframodel
planName	Kruunusillat
planCode	ID32425
planState	https://bit.ly/3gZfbuF
planDesc	05/2023
Attributes	
code	IM_coding
source	inframodel
infraCoding	0
infraCoding...	no description

Date log	Päivämäärä	Rakennuskohde	Lisätiedot	Mittauksen suoritti	Kantavuus [MPa]	Tiiveys
8/19/2022 4:10:1	12.7.2022	HL 30	SV250 Mirandankat	Eero Leino	151.666667	1.1666666
8/19/2022 4:12:1			SV250 Mirandankat	Eero Leino	128.666667	1.099715
8/19/2022 4:13:3			SV250 Mirandankat	Eero Leino	111.333333	1.1719298
8/19/2022 4:18:5			SV250 Mirandankat	Eero Leino	136	1.2035398
8/19/2022 4:20:2			SV250 Mirandankat	Eero Leino	126	1.2115384
8/19/2022 4:23:5			SV250 Mirandankat	Eero Leino	136	1.2035398
8/19/2022 4:25:2			SV250 Mirandankat	Eero Leino	125.666667	1.0927536
8/19/2022 4:26:4			SV250 Mirandankat	Eero Leino	144.333333	1.2886904
8/19/2022 4:28:1			SV250 Mirandankat	Eero Leino	142.666667	1.0567901
8/19/2022 4:29:2	20.7.2022	HL 44	SV250 Mirandankat	Eero Leino	155	1.220472

Figure 3. Digital Twin components as defined in the scope of this thesis

The closest corresponding definition found in scientific literature is the Digital Twin Instance – “A fully annotated 3D model with general dimensioning and tolerances, that describes the geometry of the physical instance and its components, a bill of materials, a bill of process, a service record and operational states captured from actual sensor data”. [10.]

2.1.4 Intersection of BIM, As-Built Models and Digital Twin

Building information technology, together with the as-built concept, forms the backbone of the Digital Twin. Initially developed during the design phase and subsequently updated with the required construction, quality assurance and maintenance data, an information model evolves into a DT once the operational phase begins. Figure 4 illustrates an example of what an information model may look like at the different phases of the life cycle of an infrastructure asset.

A high-quality 3D representation of the entire asset enables early detection of possible conflicts and clashes, which is critical for efficient urban planning, construction and coordination [11]. Furthermore, a well-maintained 3D model of built assets will provide essential background information for demolition, redesign and redevelopment activities, thereby reducing project lead times and need for corrective actions.

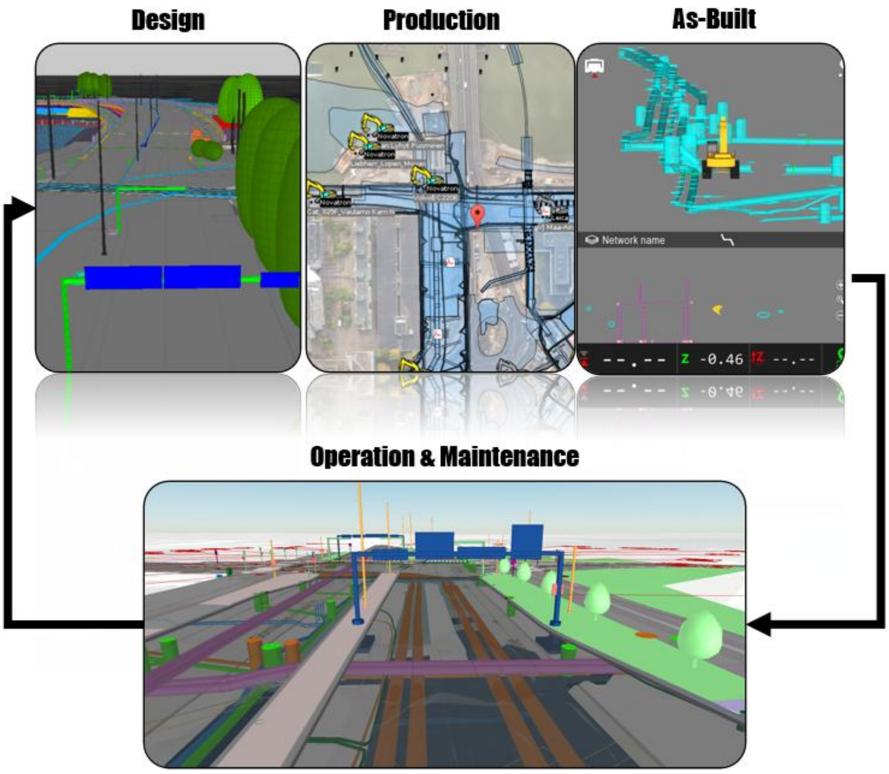


Figure 4. Model-based data lifecycle as defined in the scope of this thesis

2.2 Relevant Regulatory Framework

This chapter examines some regulatory practices and standards relevant to the civil construction sector with particular focus on as-built modelling. It is noticeable that most existing regulations and standards approach BIM primarily from the building construction and architectural perspective, leaving infrastructure domain aside. One way to explain this imbalance is that the building elements are easier to standardize - walls, windows and doors are likely to show similarities in their spatial and material properties – whereas infrastructural assets, such as roads or railways tend to be unique and showing higher levels of disparity within the same object classes. In Finland, civil construction as-built modelling has not yet been heavily regulated, and existing guidelines leave plenty of room for interpretation. However, certain foundational rules have been established by both local and national level governing bodies, which provides a solid basis for further development.

2.2.1 buildingSMART

“buildingSMART is the worldwide industry body driving the digital transformation of the built asset industry. buildingSMART is committed to delivering improvement by the creation and adoption of open, international standards and solutions for infrastructure and buildings”. [13.]

The Finnish branch of buildingSMART(bSF) is a locally established organisation that is responsible for developing and publishing country-level regulations for information modelling and digitalisation of the built environment. It is actively involved into relevant development initiatives and plays a key role in spreading knowledge about accepted BIM practices. Members of bSF are typically experienced BIM professionals who combine their bSF involvement with their regular employment in the industry.

Regarding as-built modelling in civil construction, there are several significant contributions from bSF, which are Common Infra BIM Requirements(YIV), survey coding enumeration for infra elements and course structures, Inframodel nation-wide data exchange standard and upcoming built environment information modelling standardisation package. Together, these components form strong foundation for model-based data management for civil construction in Finland. [7.]

2.2.1.1 YIV and RYTV

This document establishes general guidelines and requirements for information modelling within the infrastructure domain. It is based on current best practices and is continuously updated to reflect the industry needs. YIV covers entire asset life cycle, from preliminary data acquisition to the handover and commissioning. The document, however, does not include a chapter dedicated to maintenance & operation phase, yet the chapter is described as “coming” at the moment of writing this thesis, and provides no specific guidance on Digital Twin implementation. Further inquiry states that YIV is not likely to be updated, since it will be gradually replaced by Built Environment Information Modelling Standardisation (RYTV) document compound in the near future. [14.]

The official definition of as-built model provided by YIV has already been mentioned in chapter 2.1.2. According to YIV the model should contain XYZ-as-built records obtained by either land-survey or machine automation control equipment, error vectors, and possibly additional mapping data, obtained using methods other than global navigation satellite system (GNSS) measurements. The accuracy and density levels for the as-built measurements are structure-specific and can be found in the YIV chapter 4.3

In the YIV, as-builts are seen as a part of a larger deliverable compound, that includes process data, quality assurance documentation, initial data, as-builts and accompanying documentation. One document that is thoroughly described in the YIV is the as-built model report. It must include the following datasets: project information, model author, used software, possible deviations description, data format, coordinate system, both vertical and horizontal, file name and description of the content. Open BIM data formats listed further in this thesis are able to accommodate all of the above-mentioned parameters within the model as metadata, thus eliminating the need for a separate model report.

As-built models can be derived either from a design or execution model, by performing relevant updates or entirely from as-built survey data, by modelling an entirely new geometry dataset. Even though it is advisable to prioritize the first option, as it leverages already existing models and avoids remodelling from scratch, in long-term civil construction projects, the design or execution models may not be suitable for remodelling, because of outdated schema or extensive field modifications, which makes second option more favourable.

YIV assigns the responsibility for ensuring as-built model compliance with the requirements to the BIM coordinator. However, depending on the complexity of the project, these responsibilities may be distributed among multiple BIM professionals. As-built models must be handed over to the project owners alongside other deliverable documentation as illustrated in figure 5.

All as-built elements must carry course- and entity specific codes. Code enumeration lists are published and maintained by bSF and will be discussed in detail in the thesis chapters 2.2.1.3 and 2.2.1.4. Modern land survey hardware allows the codes to be assigned to survey points and lines, thereby making the data post-processing smoother and less prone to flaws.

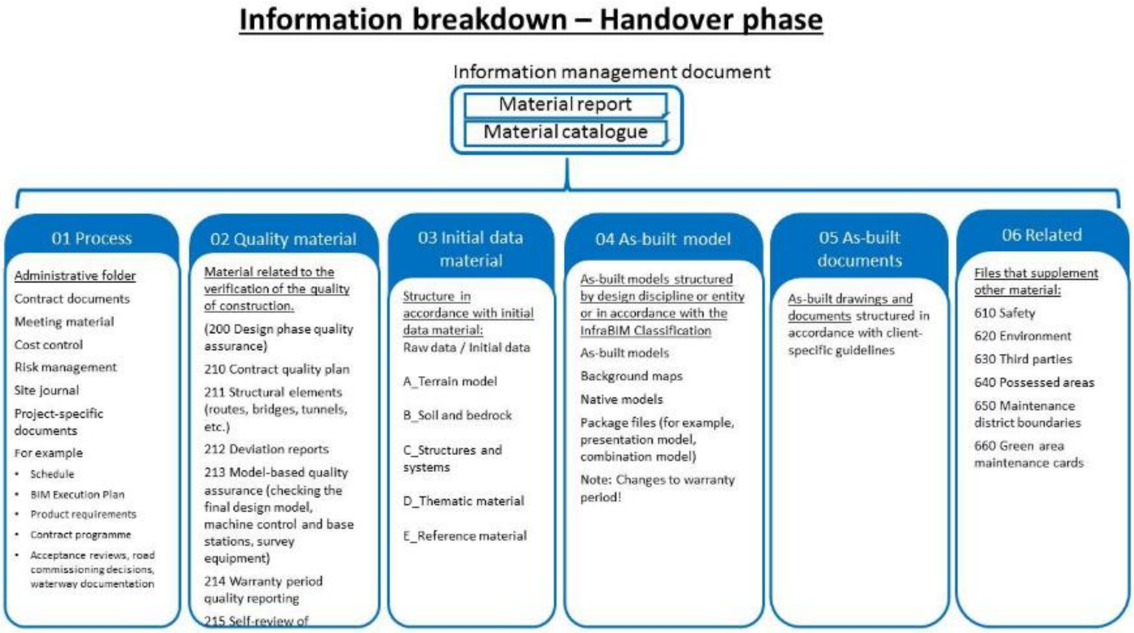


Figure 5. Deliverables according to YIV [7]

At the time of writing this thesis, there is limited information available on RYTV, as it is still under development and not expected to be adopted before 2026. Its purpose, however, is to unify and harmonise information modelling requirements across Finland and to ensure that the standardised modelling principles are applied throughout the whole lifecycle of the built environment. [14.]

2.2.1.2 InfraBIM

An as-built information package is normally comprised of multiple structural layers. Each layer should be presented as a separate model, and to keep the collection plain and clear it is advised to use InfraBIM enumeration, as visualised in figure 6. An InfraBIM code - also referred to as a surface code, is usually a 6-digit number, where the first 2 digits describe the generic domain, 2 middle digits are structure-specific and 2 last ones are reserved for more details but rarely used. The complete list in both Finnish and English can be found on the buildingSMART Finland online resource. [7.]

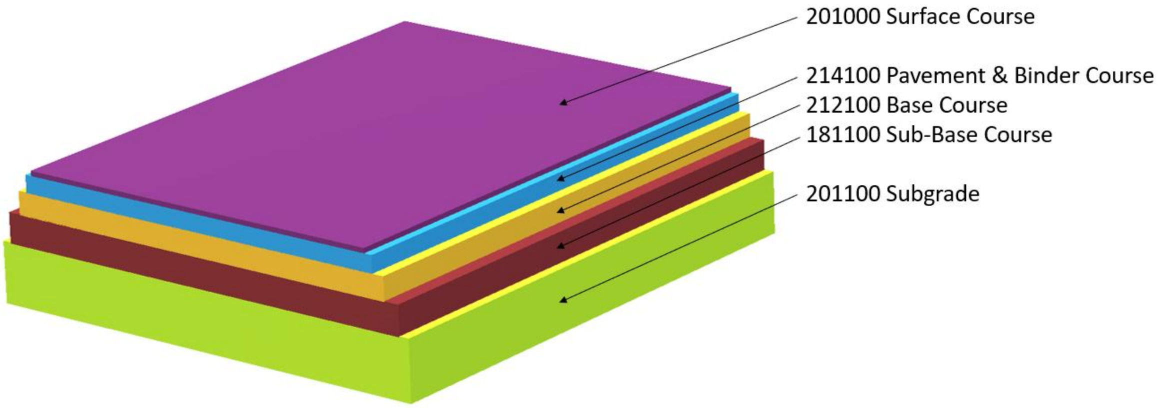


Figure 6. Surface Coding example

2.2.1.3 InfraRAK

Another important enumeration system is the object specific InfraRAK coding, usually referred to as infra code. The length and information content of the code may vary, depending on the structure, and it can describe the type, the size or the material of the entity. An example of InfraRAK codes assigned to the entities is presented in figure 6.

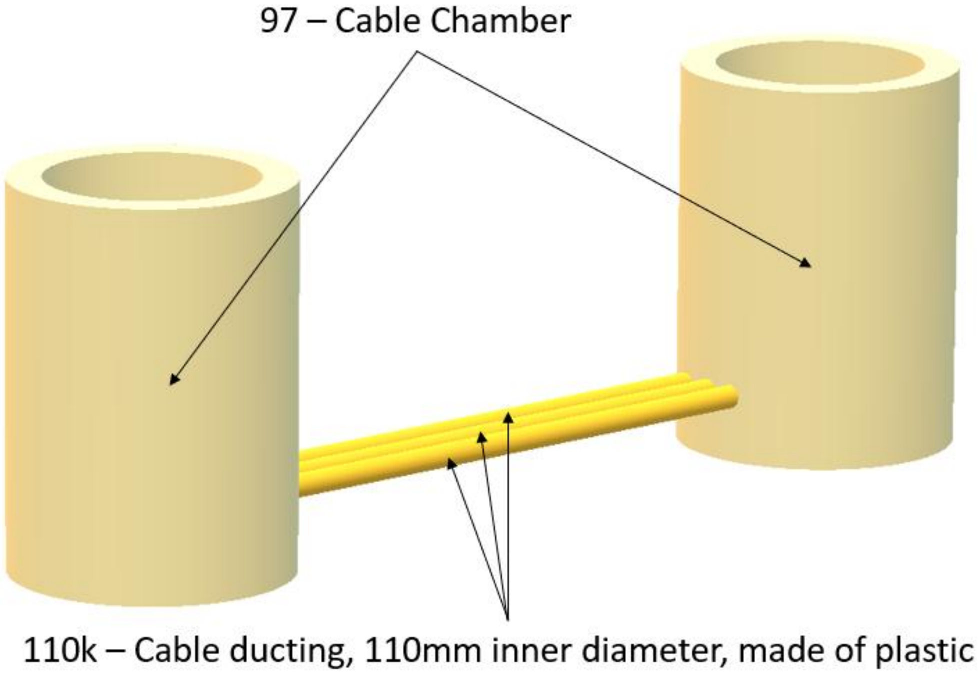


Figure 7. Infra Coding example

The complete list is available in Finnish on the buildingSMART Finland online resource. In some cases, local project-specific extensions or clarifications may be applied and, in this case, it is advisable to include the extended code list as a part of the deliverable compound. [7.]

2.2.2 Standardisations

Both buildingSMART provisions and International Organisation for Standardisation (ISO) standards are voluntary, which means that companies or projects are not legally obligated to follow them. However, choosing to operate in accordance with these regulations improves credibility of a company or a project. ISO operates on an international level, and its standards are recognised globally, whereas buildingSMART Finland provides state level guidelines, tailored for local context.

This chapter discusses two major international standards that guide the BIM: ISO19650, which specifies organisation and digitalisation of information and ISO16739, which defines industry foundation classes (IFC) data exchange format. Additionally, it covers Inframodel, a locally developed data exchange standard issued by buildingSMART Finland. Together, these three standards comprise the core guideline collection for producing, storing, exchanging and updating information model-based data in Finland.

2.2.2.1 ISO19650 Organisation and Digitalisation of Information

Th standard is defined as “organisation and digitalization of information about buildings and civil engineering works, including building information modelling”. It comprises five parts that cover the basic concepts, the delivery phase, the operational phase, information exchange and security. [15].

The standard does not directly reference as-built models, neither does it address the technical aspect of modelling. Instead, it talks about project information models (PIM) and asset information models (AIM), which, as understood, may and should include as-built models. To be precise, PIMs contain, among other data, details on geometry and location. The relevant PIM information is transferred into AIM at the end of the delivery phase and the beginning of the operational phase. [16].

In civil construction terms, transition from PIM to AIM happens as the asset construction is completed and being handed over to the owner for operation & maintenance phase. [17.] This marks BIM model data evolution from an as-built model into a DT. The respective workflow according to the ISO19650 is illustrated in figure 8.

Figure 8. Information management lifecycle according to ISO19650 [15.]

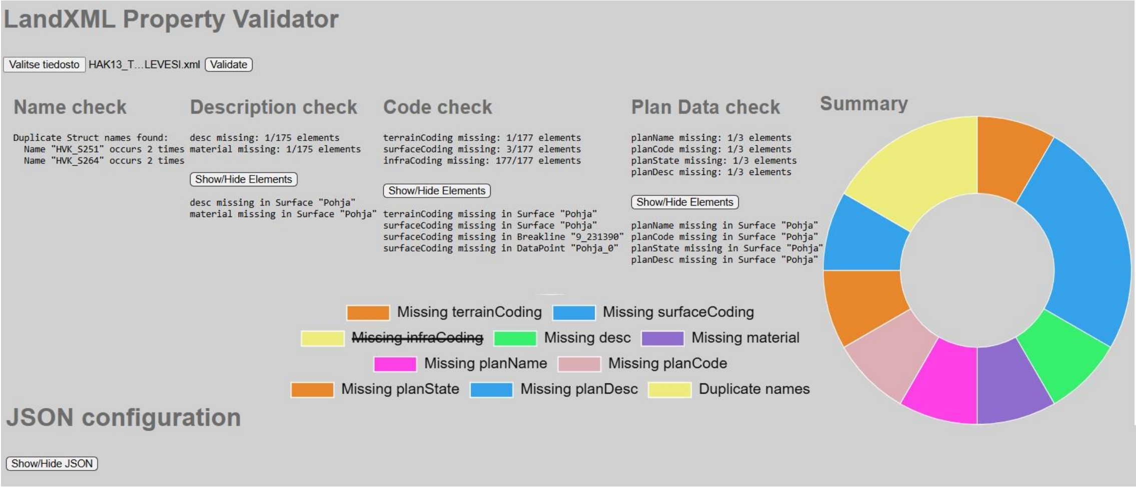
During both PIM and AIM phases, model-based data must be actively maintained in order to remain credible. Part 4 of the ISO19650 addresses this requirement through outlining a set of criteria for reviewing the exchangeable information, which includes Common Data Environment (CDE), conformance, continuity, communication, consistency, completeness and informative properties.

2.2.2.1.1 CDE, Conformance and Communication Criteria

The CDE, conformance and communication aspects involve checking the following properties:

- naming and metadata requirements
- information that supports the current workflow
- schema compliance
- encoding (including allowed characters and date format)
- units of measure. [18.]

In the context of civil as-built modelling, measurement units, naming practices, and metadata rules should be clearly defined for each project. These can be controlled using visual inspection, an Information Delivery Specification (IDS), or a locally developed validation tool, such as one used in the Crown Bridges project. The validator application checks for duplicate entity names, missing metadata properties and unacceptable values. An example validation report is shown in figure 9. In addition, information that supports the current workflow may be attached to the model through external links, leading to a locally developed as-built modelling handbooks, survey rulesets or similar project documentation.



```
Duplicate Struct names found:
  Name "HVK_S251" occurs 2 times
  Name "HVK_S264" occurs 2 times
```

```
desc missing: 1/175 elements
material missing: 1/175 elements
```

```
desc missing in Surface "Pohja"
material missing in Surface "Pohja"
```

```
terrainCoding missing: 1/177 elements
surfaceCoding missing: 3/177 elements
infraCoding missing: 177/177 elements
```

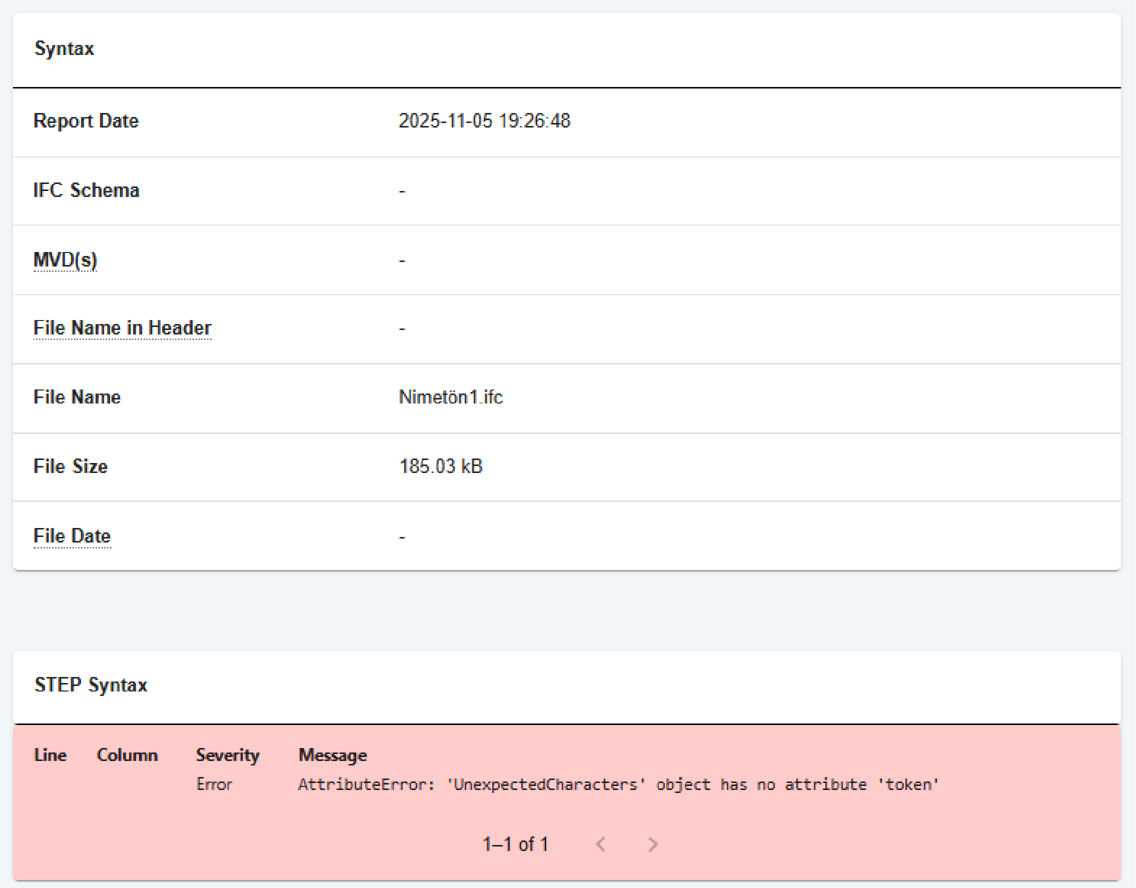
```
terrainCoding missing in Surface "Pohja"
surfaceCoding missing in Surface "Pohja"
surfaceCoding missing in Breakline "9_231390"
surfaceCoding missing in DataPoint "Pohja_0"
```

```
planName missing: 1/3 elements
planCode missing: 1/3 elements
planState missing: 1/3 elements
planDesc missing: 1/3 elements
```

```
planName missing in Surface "Pohja"
planCode missing in Surface "Pohja"
planState missing in Surface "Pohja"
planDesc missing in Surface "Pohja"
```

Figure 9. LandXML Inframodel validator application used in Crown Bridges project

Schema compliance, including allowed characters and value formats, may require the use of an officially accepted validation engine, such as one developed by buildingSMART. This engine is designed to validate IFC files in terms of syntax, schema, normative rules and industry practices [19.] An example of syntax validation report is presented in figure 10. This report warns about the usage of unaccepted characters, specifically “ö” in the filename “Nimetön.ifc”



Syntax

Report Date	2025-11-05 19:26:48
IFC Schema	-
MVD(s)	-
File Name in Header	-
File Name	Nimetön1.ifc
File Size	185.03 kB
File Date	-

STEP Syntax

Line	Column	Severity	Message
		Error	`AttributeError: 'UnexpectedCharacters' object has no attribute 'token'`

Figure 10. IFC validation output example

2.2.2.1.2 Continuity Criteria

Information continuity checks, provided by ISO19650 include the following procedures:

- comparison against other information containers
- comparison against previous versions of the information container
- comparison against previous status change
- comparison against trigger events
- previous delivery or operational phases. [18.]

In the context of civil as-built modelling, comparison against other information containers implies evaluating as-built against their surroundings, such as the modelled existing built environment or initial project data. Additionally, as-built models naturally evolve - they are refined and updated - and thus can be compared against previous versions of themselves.

Assessment and comparison against as-designed models correspond to the comparison against previous status change, and an example of previous deliveries could refer to partial handovers, when a certain portion of the asset is commissioned before the completion of the remainder. Lastly, the comparison against trigger events may involve reflecting on model updates and corrective actions, derived from model reviews and evaluation events. All these actions should be done with respect to geospatial location – at both object and site levels – schedule, spatial structure, process and entity naming, as well as associated attribute, property, and material metadata. [18.]

2.2.2.1.3 Consistency, Completeness and Informative Criteria

Consistency, completeness and informative criteria suggest the following property examinations:

- identification of duplicates, gaps, overlaps and contradictions in spatial, physical and procedural aspects
- attribute consistency and allocation
- location and property value accuracy
- Level of Information Need (LOIN) verification
- functional, technical and performance evaluation. [18.]

For the as-built models of Crown Bridges project, spatial or metainformational inconsistency examination relied predominantly on human-performed visual inspection, with assistance of automated clash analysis and validation tools. Procedural inconsistency and LOIN verification were addressed through structured process planning and recurrent review meetings. Attribute and allocation consistency were ensured by using unified element and surface coding systems – InfraBIM and InfraRAK – which were explained in chapter 2.2.1.1 and 2.2.1.2, as well as by maintaining schema compliant hierarchical structures within the selected data exchange formats.

Value accuracy along with functional evaluation of the as-built models were performed through the following activities:

- regular survey equipment calibration
- detection of duplicate survey points
- detection of points with zero elevation
- BIM platform compatibility testing
- segmentation of as-built models into appropriately sized compounds

In terms of model-based data storage, part 1 of the standard offers a set of core principles [15], which when applied, ensure the effective implementation of the CDE – a platform for collecting, managing and sharing as-built models:

- clear information container structure
- security-minded approach
- certain level of information need
- defined information quality requirements
- information traceability.

From as-built modelling perspective information container structure implies a clear folder structure, consistent naming and metadata, as well as reasonable segmentation of compounds - for example one street per compound. A security minded approach involves clear role-based separation of viewing, editing and sharing permissions, along with controlled access and robust backup protocols. LOIN establishes a minimum acceptable content for model elements. Information quality requirements, in turn, refer to documented and approved set of data parameters, such as file formats, file structure and entity and attribute classes used within the project. Finally, information traceability principle ensures that the CDE must record and provide access to the change history of each model. [15].

Regarding the information status, the standard specifies that each model within the CDE must be assigned one of four states: in progress, shared, published or archived. In the context of as-built modelling workflow, the in-progress state is applied when the model is under development and accessible only to the assigned specialist. The shared state indicates that the model has become available to other professionals for collaborative work.

The published state is assigned when the models are approved by the project owners and formally handed over for use in the operation and maintenance phase. Finally, the archive state may be applied when the asset is redeveloped, and the models no longer reflect its current condition. The model data is then retained in the historical records for reference.

This classification enables systematic tracking of the model transactions. The workflow and state description according to ISO19650 are shown in figure 11.

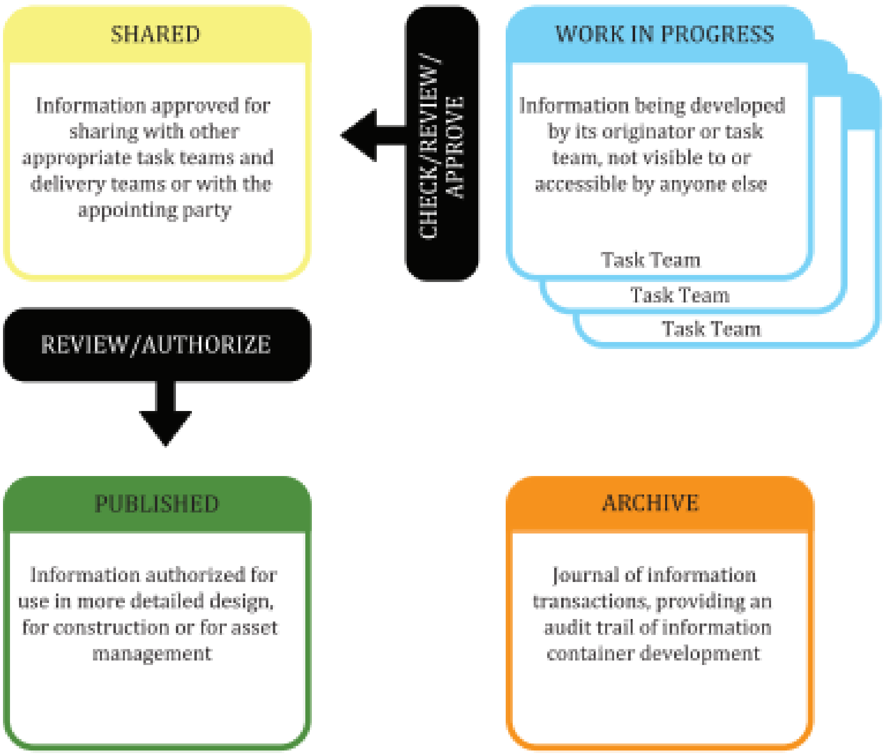


Figure 11. CDE information container states according to ISO19650:1 [15].

2.2.2.2 ISO 16739 Industry Foundation Classes

ISO 16739 is a relatively new open international standard that specifies “Industry Foundation Classes for data sharing in the construction and facility management industries”. The main outputs of the standard are the documented definitions of the conceptual data scheme and the exchange file format [20]. Its second edition expands the scope and covers infrastructural assets, such as bridges, roads, railways, and marine structures [21]. These asset classes were first introduced in IFC version 4.3, which is the latest schema version currently available. According to buildingSMART International, schema upgrade to version 4.4 is in progress, alongside working on a major schema upgrade, which will be published as IFC 5 [22].

A common misconception is that Industry Foundation Classes (IFC) is nothing but a data format, whereas in practice, it is a standardization concept and .ifc file is just one way to represent and transfer information [23]. As a non-proprietary format, IFC enables anyone to access the same piece of information written in .ifc, regardless of which BIM platform is used, which makes it suitable for visualising, storing and exchanging as-built models among engineers, project owners, and facility managers.

Influentially for as-built modelling, IFC schema defines how as-built data is written, both geometrically and semantically, which ensures data validity for a range of applications. These applications extend to sensor data integration, which transitions as-builts towards DT, which is beneficial for the operation & maintenance phase. Additionally, alongside geometry, location and metadata property sets, IFC can carry cost & schedule, extending models into the fourth and fifth BIM dimensions.

These features align with the core principles of ISO19650: information must be accessible to authorised personnel regardless of their working environment; data must be continuous and reliable, with documented ownership and change history; data must be well structured with clear inter-entity relational hierarchy; and it must be suitable for both PIM and AIM. A combination of ISO16739 and ISO19650 is often considered essential for BIM professionals [24].

2.2.2.3 Inframodel Standard by buildingSMART Finland

Inframodel (IM) is a local data exchange standard, based on international LandXML schema version 1.2, intended primarily for domestic infrastructure BIM implementation. It extends the conventional extensive markup language (XML) schema with some entity-specific properties - such as elevation levels for pipe objects - and object-specific metadata sets, for example for soil types. Data written in accordance with IM guidelines can be implemented extensively throughout the delivery phase, due to its compatibility with various BIM platforms and machine automation systems [25]. In Finland, certain asset owners require as-builts models to be delivered exclusively in the Inframodel format.

Like IFC, Inframodel is a non-proprietary format, which makes it compliant with the open BIM principles. Another similarity is that it entails both a conceptual data scheme and an exchange file format. Regarding as-built modelling, IM follows the same core principles as IFC: being open and compatible with a variety of software solutions; a part of the standard is dedicated to as-built data specification; and the ability to store metadata alongside location and geometry information. However, compared to IFC, IM has certain downsides: very limited range of object types and supported geometries, and reduced cross-platform compatibility, mainly because IM was intentionally simplified to meet infrastructure domain needs. Nevertheless, nowadays IM can still outperform IFC in some use cases, which will be examined in more detail in later chapters.

It is important to note that neither IM nor IFC schemas prescribe any measures for data protection and information security. As both of them are written in open text format, the information content may be susceptible to unauthorised modifications. Ensuring the security and reliability of model-based data stored as a human-readable text requires further examination, which is beyond the scope of this thesis. ISO19650-5 offers a set of guidelines for establishing assessment process and data security plans, while some common BIM data security practices include regular updates, relevant employee training and process automation. [26, 27.]

2.2.3 Applicable Legal Provisions

In recent years, due to rapid advancements in relevant technologies, BIM has been mentioned in regulatory provisions of different levels of obligation. Initially, municipal level actors introduced local level BIM requirements, then the Finnish national legislature included information modelling regulation into state-level legal framework.

In Finland, the regulation of digital practices in land use and construction is currently being governed by three laws, which are

- Act of the Built Environment Information System 431/2023
- Construction Act 751/2023
- Land Use and Building Act 132/1999. [28.]

While all three acts mention BIM, as-built modelling is featured - either directly or indirectly - in only two of them: construction act (finn. Rakentamislaki) and act of the built environment information system (finn. Laki rakennetun ympäristön tietojärjestelmästä).

2.2.3.1 751/2023 Construction Act

“The Construction Act regulates the design, construction and use of buildings and built compounds”. As-Built models are first mentioned in chapter 6, which concerns permit acquisition and licensing affairs. Here, an as-built model is defined as a building design or implementation information model that contains data on the actual built asset, including the parts that deviate from the original design and is presented in a machine-readable form. The model must also include basic information on the used products and their properties. [29.]

Apart from chapter 6 of the Construction Act, as-built models are also mentioned in chapter 8 (Responsibilities), which assigns as-built modelling tasks to designers and in chapter 10 (Execution), which states that the delivery process must incorporate as-built data delivery to the project owners. [29.]

It is also worth noting that the wording of the Act leaves room for interpretation and adaptation. For instance, there is no explicit definition of what “*the used products and their properties”* exactly are, and many clauses end with the following statement: “more detailed provisions may be given by the Ministry of

Environment”. Such provisions are supplementary pieces of legislation that extend and clarify the primary legal text. For example, the provision “on the content of as-designed models and official inspections” - which is currently under preparation - is going to provide templates for building permit applications, referenced in chapter 6 of the Construction Act. [29.]

2.2.3.2 431/2023 Act of the Built Environment Information System

“Act of the built environment information system applies to the establishment, maintenance and development of the Built Environment Information System and its services, as well as to the processing of data submitted into and stored in the information system”. The act does not address as-built models directly, instead, it refers to the Clause 73 Construction Act, which states that as-built models must be delivered to the built environment information system. [30.]

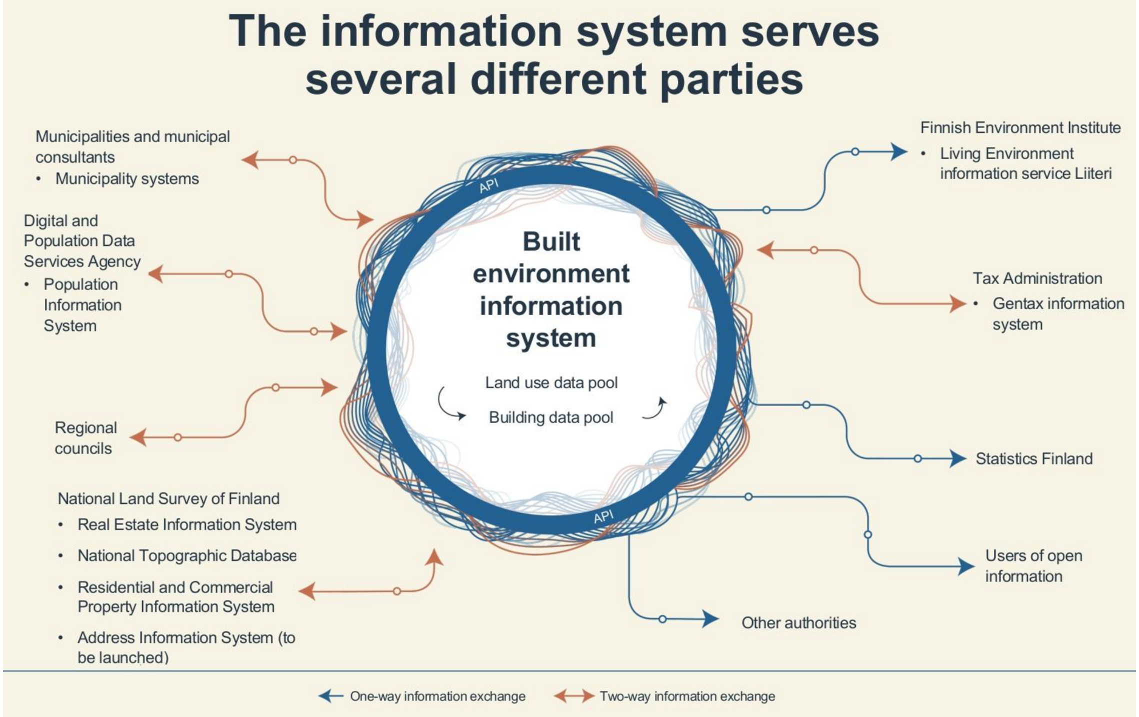
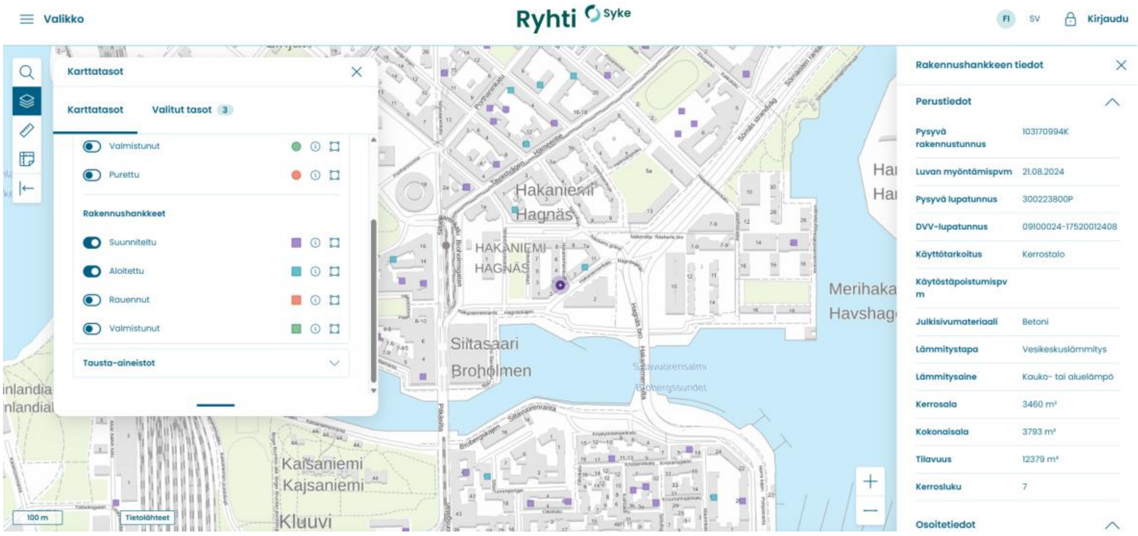


Figure 12. Links between the built environment information system and other institutions [31].

The built environment information system - *RYHTI* in Finnish - is an upcoming, nation-wide information system that aims to connect building and land use data in a unified database [31]. The objective of the system is to ensure that the information remains well structured, available and accessible. Furthermore, RYHTI will support the planning and construction of the living environment, facilitate the processing and issuance of the building permits, and maintain the property, population and the business registries. The framework of the system is illustrated in figure 12. Currently the system interface offers only a 2D map-based view with a search directory, as shown in figure 13, but lacks a 3D-visualisation engine for model viewing.



Pysyvä rakennustunnus	103170994K
Luvan myöntämispvm	21.08.2024
Pysyvä lupatunnus	300223800P
DVV-lupatunnus	09100024-17520012408
Käyttötarkoitus	Kerrostalo
Käytöstäpoistumispvm	
Julkisivumateriaali	Betoni
Lämmitystapa	Vesikeskuslämmitys
Lämmitysaine	Kauko- tai aluelämpö
Kerrosala	3460 m²
Kokonaisala	3793 m²
Tilavuus	12379 m³
Kerrosluku	7

Figure 13. RYHTI map-based interface [32].

The built environment information system is currently under development and is planned to be fully operational by 2029. However, data submission into the system has been possible since 2024 [31]. Regarding the asset delivery, the system will contain

- final inspection protocol, completed in accordance with model-based data (format not specified)
- as-built models (in IFC format)
- domain specific designs (in PDF and/or IFC)
- modifications made during the construction phase, referred to as as-built drawings (PDF, IFC)
- modification authorisation protocol, based on model-based data (in PDF)

- other possible contents to be announced later, as the system develops.

Although, the RYHTI content description does not address infrastructure as an individual domain, instead referring to construction and land use, the term Built Environment included in the name leaves an impression that the information system aims to collect data on buildings and surroundings, both above and below ground. This interpretation is also supported by the Finnish Parliament's statement that the risks associated with data on critical infrastructure should be assessed and relevant data-safety and security practices should be implemented. The responsibility for establishing and developing the system lies on the Finnish Environment Institute (SYKE). [30.]

2.2.3.3 The Intersection between the Regulations and the Summary

Apart from the as-built model required for the construction permit, chapter 6 of the Construction Act provides a list of crucial information needed for construction authorisation. The list includes, among others, drawings and a 3D model of the asset, bill of foundation works, bill of energy consumption, bill of environmental impacts and bill of materials. The bill of materials was already mentioned in the common Infra BIM requirements (YIV) and the digital twin definition. 3D model is described as a combination of location, shape and the geometrical data of the asset that resembles the definition of PIM content in ISO 19650. [15.]

Regarding the bill of environmental impact and energy efficiency, the Construction Act assigns the responsibility for maintenance of a nation-wide emission database to the Finnish Environment Institute [29]. The database is intended to hold data on carbon storage, carbon sink, carbonation and vegetation, covering material production, logistics, replacement, recycling, construction and operation phases. The information is free to access and use, as long as the source is referenced [33]. The database is divided into building and infrastructure sections and is available for integration [34].

The laws are interconnected through internal references and thus best be applied together. The Construction Act came into force 01.01.2025 which means the data submission into the RYHTI system, and the model-based permit and authorization process are now mandatory [31]. However, the transition period for the data submission into the system extends until 01.01.2028.

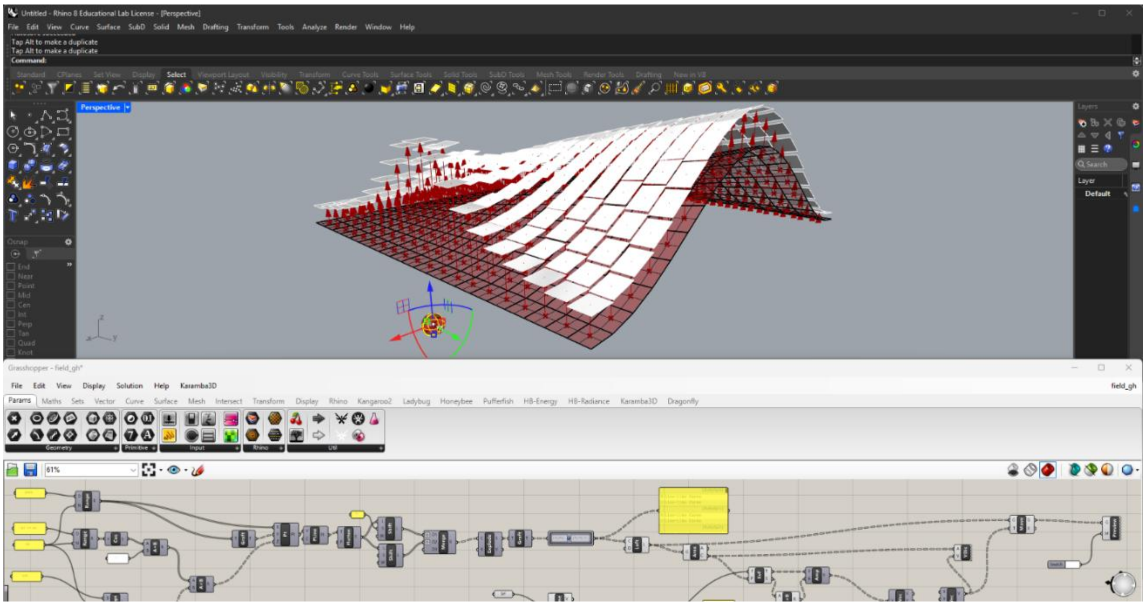
In summary, BIM practices are a subject to regulation at multiple levels and as-built models are referenced on each of them. 3D geometry, location and material properties are repeatedly emphasized and may be considered a minimum required level of information. Nevertheless, further clarification on 3D geometry and the material description may be necessary - potentially through provisions issued by the Ministry of Environment - as current definitions leave room for interpretation, which could result in significant variation in levels of detail between models.

2.3 Technologies enabling Infrastructure As-Built Modelling

2.3.1 Design, Sketching and Sculpting

Tools for BIM modelling generally fall into the CAD software category, which has been in use since the 1980s. Typical BIM modelling platforms operate with a scalable set of rules and parameters, which may include predefined geometry types or families, user-defined or standardised object libraries, and relational hierarchies between elements. [5.]

The parametric approach keeps the process under the engineer’s control, while reducing room for mistakes, as each parameter conforms to specific set of rules and limitations. It also increases the process flexibility and adjustability, allowing the exploration of all possible design variations [3]. Additionally, classifying objects into types and families enhances the modelling workflow by enabling features like entity filtering or bulk editing. An example of parametric modelling application interface is shown in figure 14.



```
Tap Alt to make a duplicate
Tap Alt to make a duplicate
Command:
```

Figure 14. Parametric modelling workflow example in Rhino 3D, featuring Grasshopper visual programming language

Modern applications typically offer user a 3D interface with an option to include 2D drawings as a reference. They also allow inclusion of 2D elements - such as texts, symbols and annotations – directly into the 3D environment [5]. In response to the phenomenon of extended BIM dimensions, some applications offer immediate 4D integration, when the schedule chart is visualised alongside the 3D view [35].

Historically, modelling applications required significant computing power and merely featured local hard drive installation. In contrast, modern software developers tend to lean towards browser-based solutions, offering either fully browser-based or a hybrid browser-and-desktop product, which allows greater flexibility.

Today’s generation of BIM tools is comprised of both large commercial product families - such as Autodesk, Bentley or Tekla - and community-developed BIM platforms, like Bonsai add-on for Blender (formerly known as BlenderBIM). However, when filtering out software solutions that are not tailored for civil construction, the list narrows down to only a few suitable options. These include Trimble Novapoint, Autodesk Civil 3D, Bentley Open Roads Designer and Tekla

Structures. [36.] All these applications are suitable for as-built modelling – whether the model is based on land survey or updated from as-designed data.

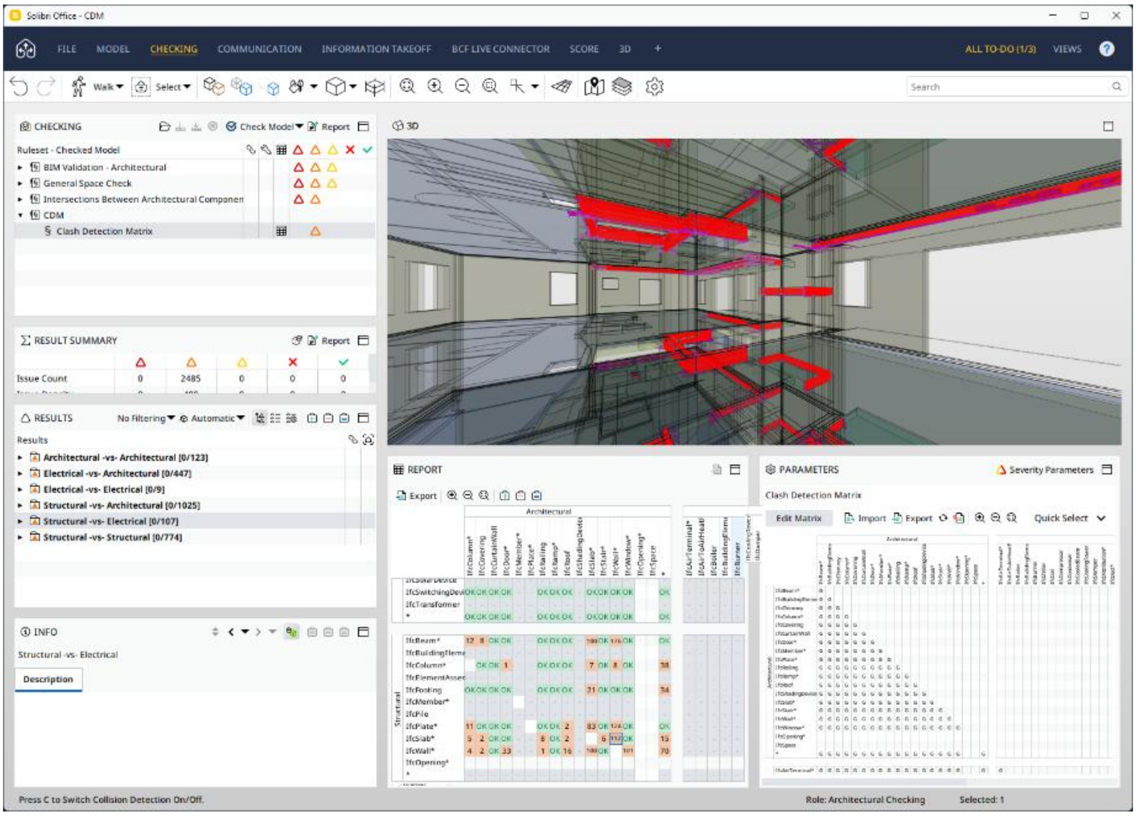
2.3.2 Rendering, Sharing and Analysing

Another common type of a BIM application is a model viewer. BIM model viewer is typically a software capable of reading BIM files, rendering geometries, and displaying entity-specific properties, but does not allow any model manipulations [4]. Most viewers by default place rendered geometries in a three-dimensional space, even when rendering 2D elements. Some viewers extend their functionality to showing background maps, generating cross-section view, checking for collisions, leaving reviews and comments and visualising the 4D schedule.

When a viewer offers extensive BIM collaboration toolset, it can practically function as a CDE, defined as "agreed source of Information for any given project or Asset, for collecting, managing and disseminating each Information Container through a managed process″ [15]. Collaboration features may include writing notes, assigning tasks, sharing views or sending messages through the viewer platform.

Some globally known BIM model viewing platforms are Autodesk Viewer, Trimble Connect, Bentley Viewer, Solibri and Dalux [36]. Since an as-built model is a BIM model, it is expected to be compatible with a viewer application, as long as it is in a prominent, non-proprietary format. Typical operations done using such software featuring as-builts include:

- comparing as-builts against as-designed models, possibly performing clash analysis, an example of which is shown in figure 15.
- visualising current state to support the communication and decision making
- calculation of used materials and object quantities.



	△	△	△	✕	✓
Issue Count	0	2485	0	0	0

- Architectural -vs- Architectural [0/123]
- Electrical -vs- Architectural [0/447]
- Electrical -vs- Electrical [0/9]
- Structural -vs- Architectural [0/1025]
- Structural -vs- Electrical [0/107]
- Structural -vs- Structural [0/774]

Structural -vs- Electrical

Figure 15. Structural and clash analysis workflow in Solibri [37.]

2.3.3 Land Surveying

As discussed earlier, an as-built model represents the built asset exactly as it is, in terms of appearance, condition, structure or function [4]. Depending on the project specifics, the location and the shape of an object can be confirmed through either land survey or reality capture. Land surveying is more natural for infrastructure projects, as it favours open areas and requires a stable connection to the receiver, whereas reality capture technology is better suited for enclosed spaces, such as tunnels or building interiors.

Land surveying is a process of obtaining XYZ coordinates of specific points in space. Tools commonly used by land surveyors in Finland include portable GNSS device, a terrain tablet and a real time kinematic (RTK) rover are demonstrated in figure 16. [38.]

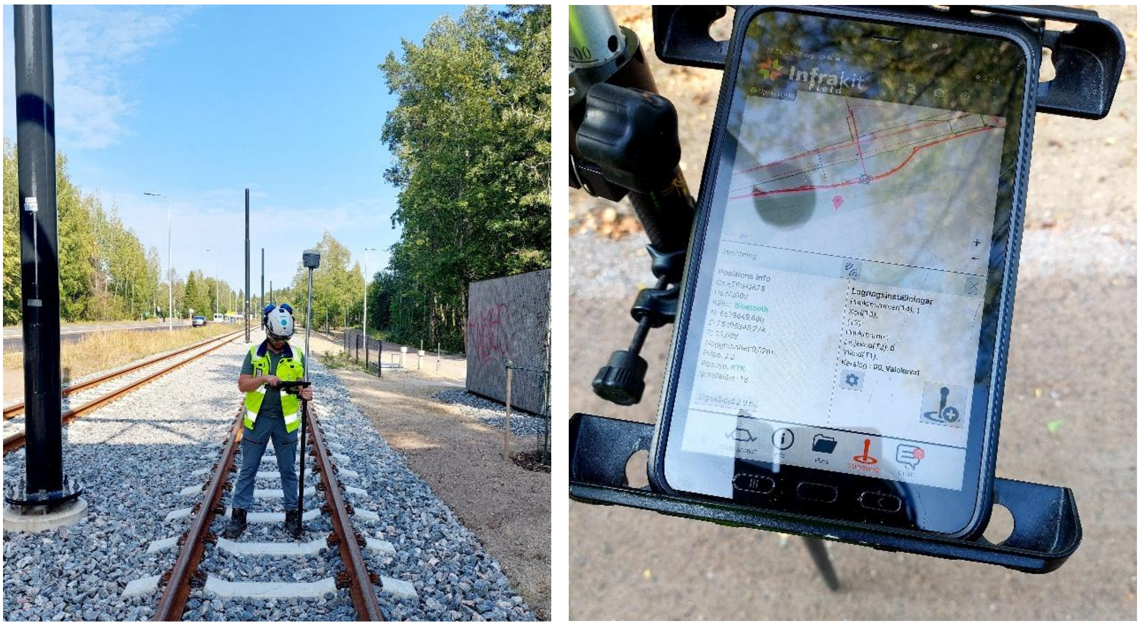


Figure 16. Land Surveying equipment illustration

Two most common surveying methods are total station surveying and the GNSS, often referred to as global positioning system (GPS) instead. A total station is a high precision device, used to measure angles and distances, and to determine the target location based on those measures. It is ideal for structures that require a high level of accuracy. GPS surveying, on the other hand, relies on satellite-based positioning and normally features handheld devices. GPS has a lower accuracy level, therefore unsuitable for high precision measurements but ideal for large area mapping. [39.]

The desired output is a set of survey points, presented either as a group of scattered dots or a continuous line. Each survey point must have three-dimensional coordinates along with a concise description. While the exact output format may vary depending on the device and region, it eventually comprises a table. An example of such table is demonstrated in figure 17.

Point data is then converted from native format to a BIM-compatible format, such as CSV. Survey points are often referred to as as-built points, because they form a backbone of an as-built model, but do not yet constitute a complete model.

DATE	POINT NUMBER	CODE	SURFACE CODE	LINE CODE	X	Y	Z	DZ	SOURCE
22/07/2022, 13:00	KAN46	100 (L) (Maanpinnan hajapiste/-viiva)	14	0	6672729.609	25501122.051	12.994	-0.010	TOTAL_STATION
22/07/2022, 13:00	KAN45	100 (L) (Maanpinnan hajapiste/-viiva)	14	0	6672728.892	25501123.514	13.018	0.015	TOTAL_STATION
22/07/2022, 13:00	KAN44	100 (L) (Maanpinnan hajapiste/-viiva)	14	0	6672728.286	25501125.178	12.987	-0.004	TOTAL_STATION
22/07/2022, 13:00	KAN43	100 (L) (Maanpinnan hajapiste/-viiva)	14	0	6672727.403	25501127.134	12.989	0.002	TOTAL_STATION
22/07/2022, 13:00	KAN42	100 (L) (Maanpinnan hajapiste/-viiva)	14	0	6672736.584	25501131.661	12.361	-0.012	TOTAL_STATION

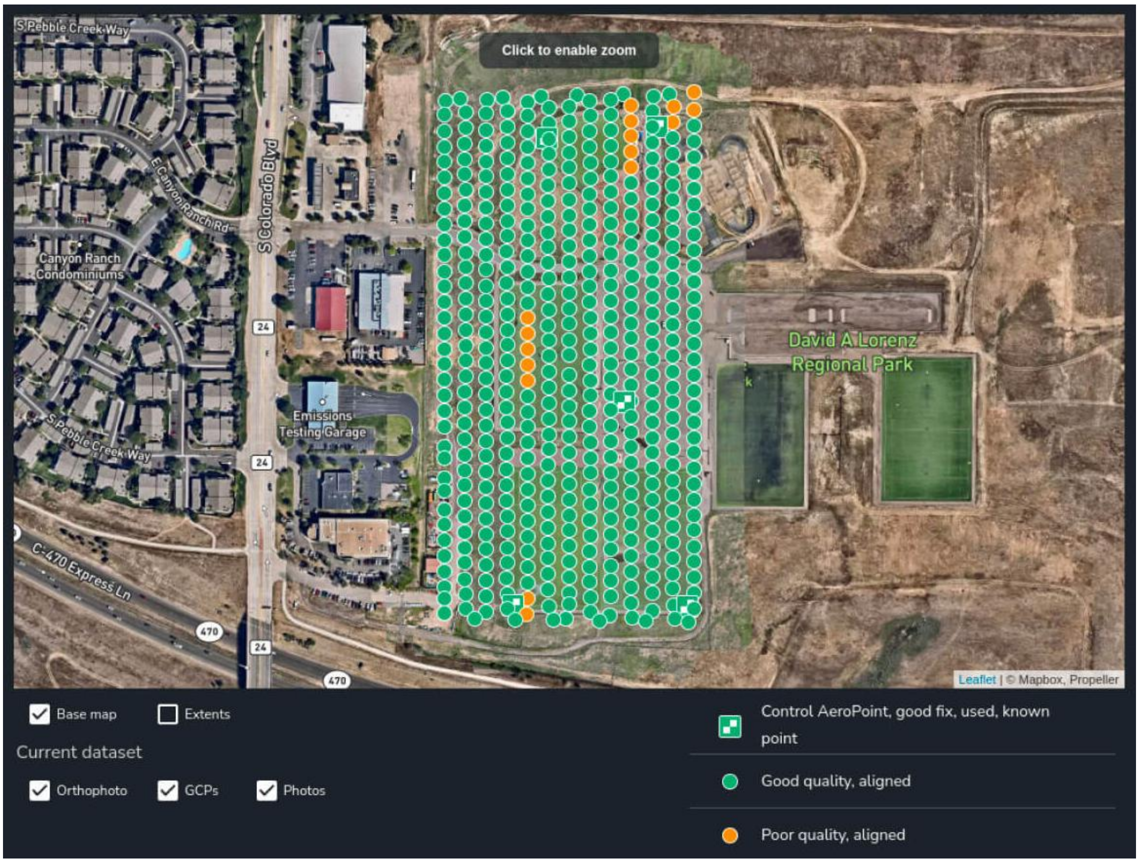
Figure 17. As-Built points data table example

2.3.4 Reality Capture

While land surveying normally requires manual work - a surveyor must carry the device to each measurement location - the industry now offers more automatic solutions to collect geospatial data. Two commonly used reality capture methods are laser scanning and photogrammetry.

Laser scanner is either a stationary or handheld device that uses LiDAR technologies to create Point Clouds. A point cloud is a set of 3D points that represents an object, space or the whole asset [4].

Photogrammetry is a process of generating geospatial data through image analysis. To cover larger areas, drones are actively used to capture images. These images are captured according to defined flight plan and later processed using suitable software. The output of the photogrammetric analysis is usually either a mesh or a point cloud [4]. An example of photogrammetry processing software is demonstrated in figure 18. and an example of a point cloud produced from a set of aerial images is shown in figure 19.



☑ Base map ☐ Extents

Current dataset

☑ Orthophoto ☑ GCPs ☑ Photos

▣ Control AeroPoint, good fix, used, known point

● Good quality, aligned

● Poor quality, aligned

Figure 18. Drone Survey workflow example in Propeller [40.]

Point clouds can be converted into meshes and later into BIM models [41]. Two typical challenges with point clouds are high point density and the difficulty to distinguish object boundaries. In their simplest form point clouds are just sets of vertices, while some scanners are also capable of capturing their colour, waveform and other properties. Dense point clouds require extensive computing power for data rendering and manipulation. While modern applications offer efficient computational algorithms for point cloud thinning and object extraction, producing a proper BIM model still requires substantial manual work and this process commonly known as Scan2BIM.

Figure 19. Point cloud obtained from a drone survey

From civil construction and as-built modelling perspective, both photogrammetry and laser scanning can be extremely useful, though each comes with a set of requirements and limitations. A drone survey normally requires a drone heavier than 250g, which in the EU are subject to drone registration, drone operator licence and an activity report [42]. Additional restrictions may derive from the surroundings, residential areas and the wildlife habitats.

Drone data processing requires specialised software and skilled operators. When done properly, a drone survey can provide accurate reference points for as-builts, as well as high-quality background maps to support modelling process. Figure 20. illustrates as-built mapping from a mesh obtained from drone survey. Additionally, drones require open airspace and are generally used above ground, which makes them best suited for superstructure surveying. However, research suggests that the underground drone usage is possible with a specially adapted equipment. [43.]

Laser scanning similarly requires dedicated software and trained personnel, along with expensive equipment and high-performance computing capacity. Also, point clouds tend to contain redundancies and noise, making further processing challenging and laborious. Despite rapid advances in scanning technology, fully automated Scan-to-BIM solutions still produce results far from being satisfactory. [44.]

Figure 20. Drone survey-based mapping workflow in Propeller

2.3.5 Digital Twin Enabling Applications

Using digital twin during the delivery phase positively impacts three major project success factors: quality, cost and schedule [45]. As discussed earlier, the perception of the digital twin phenomenon and the definition of the term itself may vary depending on industry, application purpose, and level of user's expertise. A digital twin may be seen as a model, a tool, or a process. Therefore, to understand its relevance to as-built modelling, it is necessary to examine several different application types that may align with these interpretations and assess how do they affect the above-mentioned delivery success factors.

2.3.5.1 Digital Twin as a Model

The definition and basic concepts of CDE were previously described in chapter 2.3.2 of this thesis. Technically, CDE is not a digital twin application by itself, but it can be transformed into one through a set of external manipulations. If the model data within CDE is classified as as-built or as-is, the linked documentation provides access to the latest versions, and if the model updates are synchronised with the asset construction, then the CDE content actually mirrors the current state of the asset. In this case, the platform can function as a Digital Twin enabling application.

An example of a widely used common data environment for infrastructure that meets above mentioned criteria, is Trimble Connect. It is defined as “a cloud-based common data environment and collaboration solution, designed for real-world construction”, and its standard user interface is illustrated in figure 21. [46.]

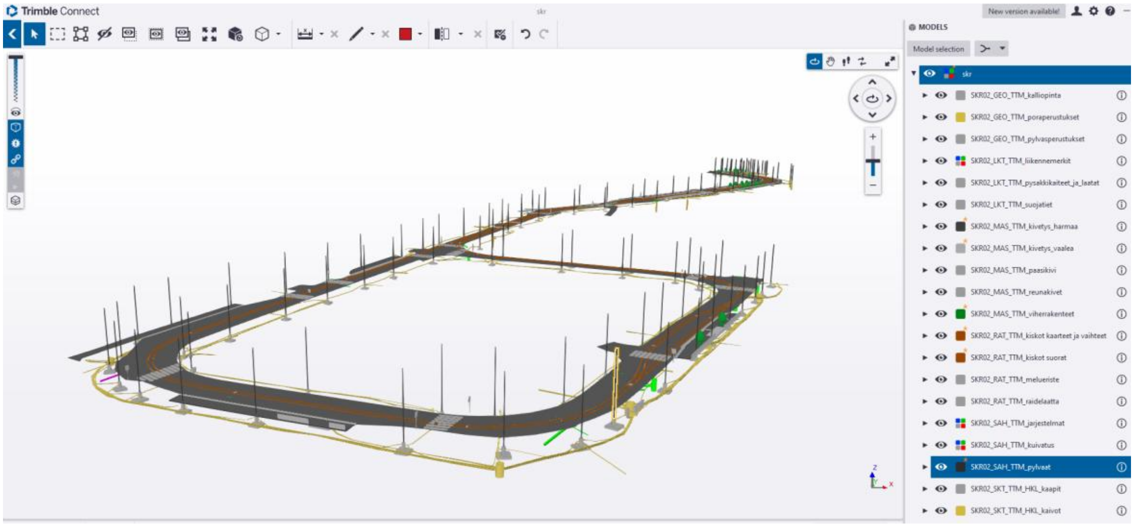


Figure 21. Trimble Connect user interface illustration

This type of application extensively supports quality assurance and schedule control. Cost and CO2 information may be also linked to the model for budget and emission progress monitoring. In this context, as-built data may also be referred to as as-is data.

2.3.5.2 Digital Twin as Technology

Applications that continuously receive real-time sensor data or vehicle location, meet the Digital Twin as a Tool or Digital Twin as a Process description. One example is a weather sensor that constantly transmits temperature, wind speed, and humidity readings. When installed on a bridge, such sensor becomes a practical tool to obtain valuable data for structural health assessment and temperature-response mapping, as bridge deterioration is influenced, among other factors, by climate fluctuations and extreme weather events. [47.]

Another relevant example is a platform for real-time on-site construction vehicle management, which is especially valuable for large construction sites due to their vast area and numerous subcontractors [35]. Such applications contribute to delivery schedule adherence, occupational safety, and resource planning. It is worth noting that not all solutions of this type support 3D visualisation, as it is not their primary function. However, some platforms combine real-time site management capabilities with a 3D model viewer, resulting in a powerful toolkit for civil construction projects. One example is Infrakit Office, which is described as “a cloud-based BIM Viewer with integrated land survey and vehicle management” [48], and its operational interface is illustrated in figure 22.

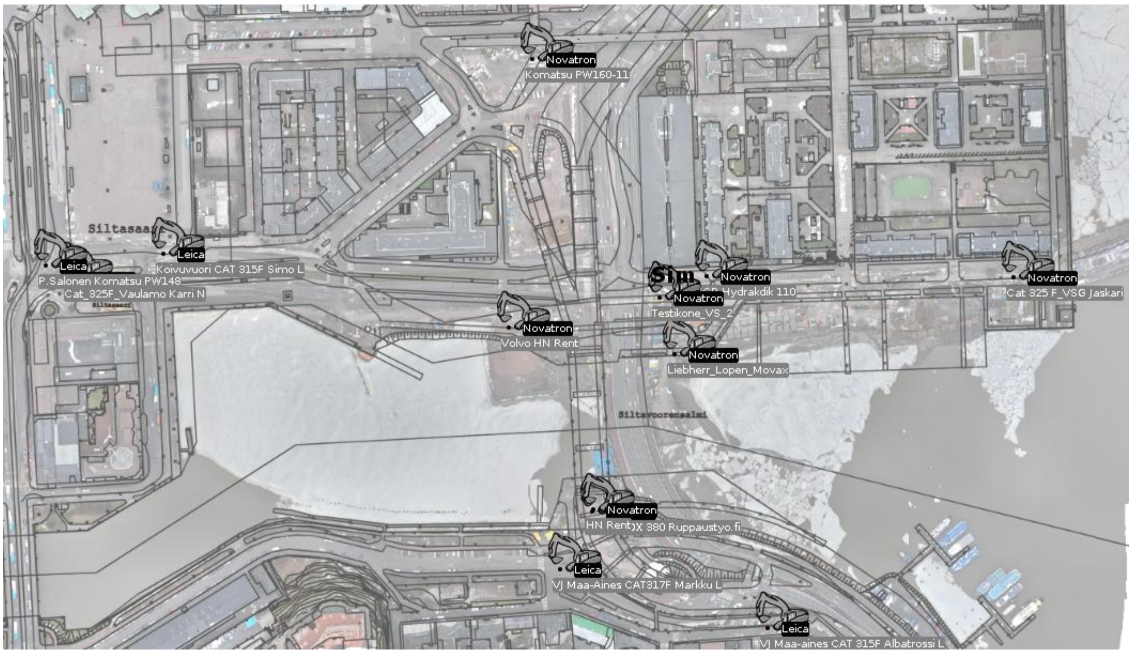


Figure 22. Vehicle management workflow illustration

In most cases project information management relies on a range of specialised software applications, as a single CDE rarely meets all requirements or satisfies all involved parties [49]. In such situations, several interdependent data environments that serve different purposes are employed in combination. For example, a CDE for BIM management may be integrated with a document management hub, aligned with the site and vehicle monitoring platform, and connected to sensor data transmitters. In case of applying such a multi-platform approach, it is important to consider that an increased number of interconnected data environments also increases the risk of data being lost in transition, as more data migration routes are created and more information is transferred through. An example of indirect integration of BIM viewing platform with weather sensor data is illustrated in figure 23.

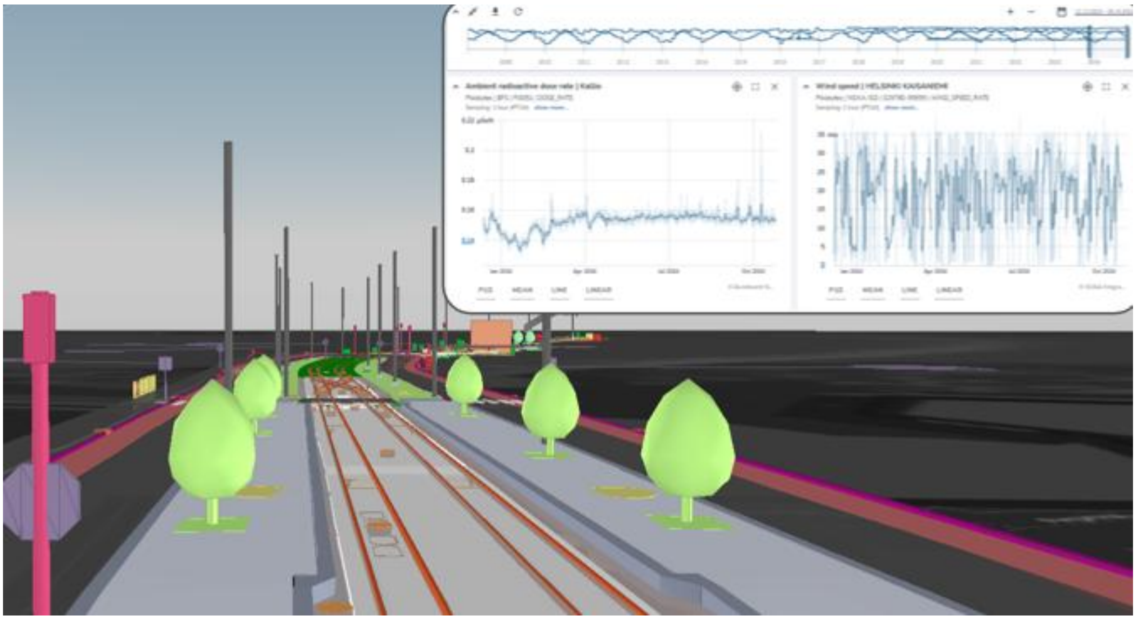


Figure 23. An example of open weather sensor data [50] being embedded into an as-Built BIM model

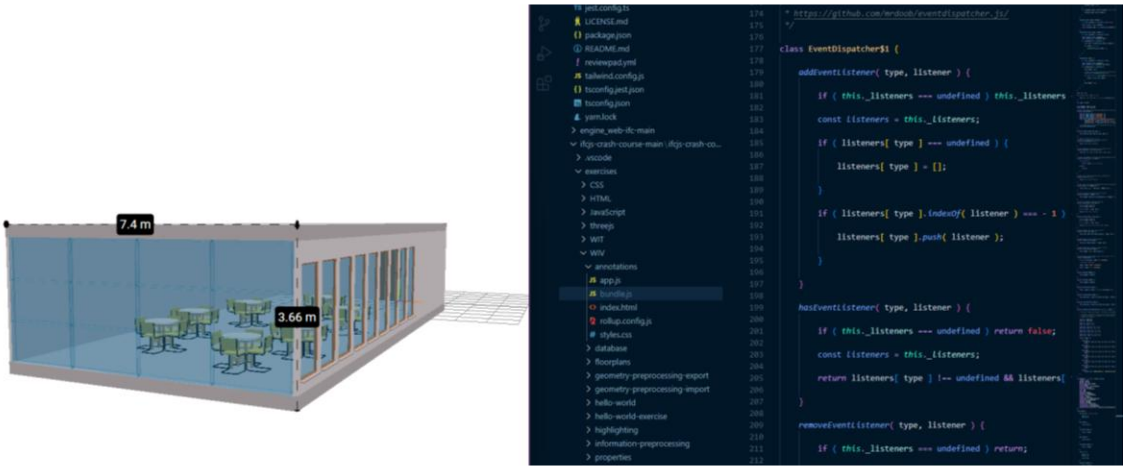
2.3.6 Developer-Friendly Tools

A distinct category of tools designed for engineers with a computational mindset or those who prefer creating their own customised applications. This chapter introduces two prominent instruments of this kind: IfcOpenShell and ThatOpenEngine. They both are focused on working with IFC data format, which, along with other BIM modelling formats, were examined in detail in chapter 2.3.7.

2.3.6.1 That Open Engine

ThatOpenEngine, also formerly known as ifc.js is a JavaScript-based toolkit that enables building custom 3D BIM applications from scratch [51]. The selection of available features includes but is not limited to, displaying and navigating IFC models, performing measurements, generating model reports, and integrating with external services such as the Mapbox background map application programming interface (API) [52]. The toolkit is being continuously developed with new features being added regularly.

A key advantage of ThatOpenEngine is that it does not require users to be professional developers, although, some understanding of HyperText Markup Language (HTML), cascading style sheets (CSS) and JavaScript (JS) or TypeScript (TS) is beneficial. The developing company also provides training courses for both beginners and advanced users. The courses cover all related techniques and workflows, and an example of what can be done with ThatOpenEngine is illustrated in figure 24.



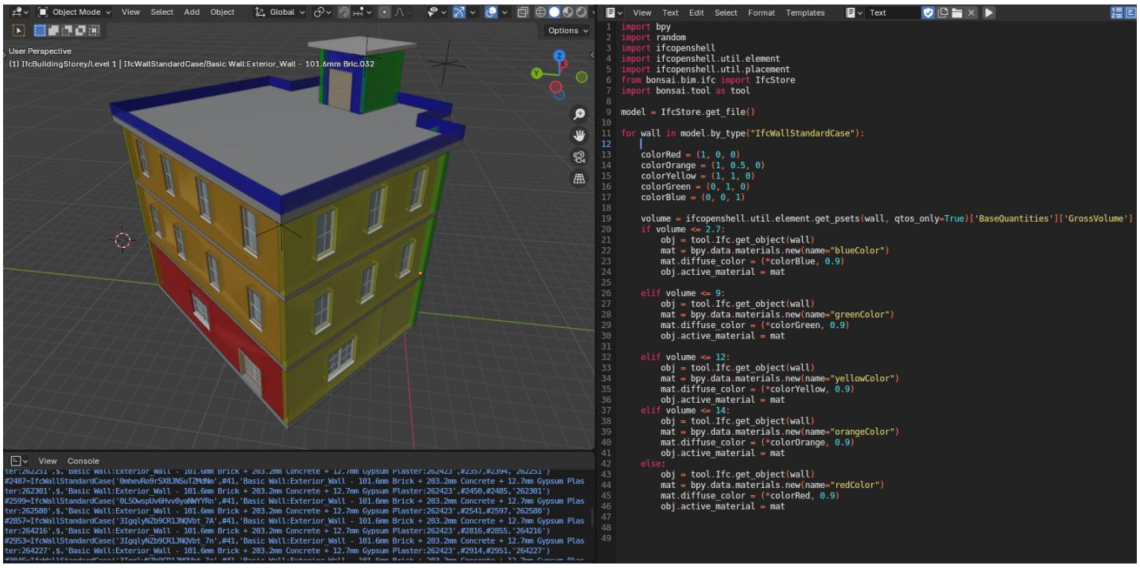
```
 * https://github.com/mrdoob/eventdispatcher.js/
 */

class EventDispatcher$1 {

	addEventListener( type, listener ) {

		if ( this._listeners === undefined ) this._listeners

		const listeners = this._listeners;

		if ( listeners[ type ] === undefined ) {

			listeners[ type ] = [];

		}

		if ( listeners[ type ].indexOf( listener ) === - 1 )

			listeners[ type ].push( listener );

		}

	}

	hasEventListener( type, listener ) {

		if ( this._listeners === undefined ) return false;

		const listeners = this._listeners;

		return listeners[ type ] !== undefined && listeners[

	}

	removeEventListener( type, listener ) {

		if ( this._listeners === undefined ) return;
```

Figure 24. That Open Company (formerly ifc.js) implementation example

2.3.6.2 IfcOpenShell

IfcOpenShell is an open-source programming library, that primarily features C++ and Python scripting and is considered the oldest and most mature IFC library available. Its capabilities include viewing, editing and authoring .ifc files, including complex geometries and relational inter-entity hierarchies, generating drawings, performing structural analysis, creating schedules, calculating quantities and more. Just like ThatOpenEngine, the library is being regularly updated featuring numerous contributors from around the world [53.]

Apart from the core library itself, IfcOpenShell ecosystem entails a number of utilities, with the most notable being Bonsai, also formerly known as BlenderBIM. Bonsai is a graphical add-on for a freeware tool named Blender, which connects IfcOpenShell features to a user-friendly, CAD-like user interface. An example of Blender workflow, that features Bonsai add-on and IfcOpenShell scripting, is illustrated in figure 25.



```python
import bpy
import random
import ifcopenshell
import ifcopenshell.util.element
import ifcopenshell.util.placement
from bonsai.bim.ifc import IfcStore
import bonsai.tool as tool

model = IfcStore.get_file()

for wall in model.by_type("IfcWallStandardCase"):

    colorRed = (1, 0, 0)
    colorOrange = (1, 0.5, 0)
    colorYellow = (1, 1, 0)
    colorGreen = (0, 1, 0)
    colorBlue = (0, 0, 1)

    volume = ifcopenshell.util.element.get_psets(wall, qtos_only=True)['BaseQuantities']['GrossVolume']
    if volume <= 2.7:
        obj = tool.Ifc.get_object(wall)
        mat = bpy.data.materials.new(name="blueColor")
        mat.diffuse_color = (*colorBlue, 0.9)
        obj.active_material = mat

    elif volume <= 9:
        obj = tool.Ifc.get_object(wall)
        mat = bpy.data.materials.new(name="greenColor")
        mat.diffuse_color = (*colorGreen, 0.9)
        obj.active_material = mat

    elif volume <= 12:
        obj = tool.Ifc.get_object(wall)
        mat = bpy.data.materials.new(name="yellowColor")
        mat.diffuse_color = (*colorYellow, 0.9)
        obj.active_material = mat
    elif volume <= 14:
        obj = tool.Ifc.get_object(wall)
        mat = bpy.data.materials.new(name="orangeColor")
        mat.diffuse_color = (*colorOrange, 0.9)
        obj.active_material = mat
    else:
        obj = tool.Ifc.get_object(wall)
        mat = bpy.data.materials.new(name="redColor")
        mat.diffuse_color = (*colorRed, 0.9)
        obj.active_material = mat
```

```
ter:262251',$,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plaster:262423',#2357,#2394,'262251')
#2487=IfcWallStandardCase('0mhevRo9r5X8JNSuTZMdNm',#41,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plas
ter:262301',$,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plaster:262423',#2450,#2485,'262301')
#2599=IfcWallStandardCase('0LSOwspUv6Hvv0yaNWYYRn',#41,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plas
ter:262580',$,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plaster:262423',#2541,#2597,'262580')
#2857=IfcWallStandardCase('3IgqlyNZb9CR1JNQVbt_7A',#41,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plas
ter:264216',$,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plaster:262423',#2816,#2855,'264216')
#2953=IfcWallStandardCase('3IgqlyNZb9CR1JNQVbt_7n',#41,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plas
ter:264227',$,'Basic Wall:Exterior_Wall - 101.6mm Brick + 203.2mm Concrete + 12.7mm Gypsum Plaster:262423',#2914,#2951,'264227')
```

Figure 25. Bonsai + Blender workflow example, featuring IfcOpenShell scripting

2.3.7 Data Exchange Formats

BIM file formats are designed to store and share structured information. They are generally divided into proprietary and non-proprietary, also known as open BIM formats. Proprietary formats can be read only by the software they originate from or by a limited set of applications equipped with format converters. Examples include RVT, native format for Autodesk Revit, and NWD, native format for Autodesk Navisworks [54]. This chapter focuses on two very common and infrastructure-relevant open BIM formats, IFC and LandXML.

2.3.7.1 LandXML

The LandXML format was developed by an industry consortium of partners based on XML syntax [55, 56]. XML’s structure is straightforward and easy to comprehend, as it follows class-subclass relational hierarchy, in which each class can contain multiple subclasses. This structure is often called an XML-tree, because the structure can be visualised as a root branching into leaves. Similar markup language structure is used in HTML; therefore, it may look familiar even to users who have never worked with LandXML before. An example of XML syntax is shown in listing 1.

```
<city>
  <transportNetwork>
    <speedTram>
      <route>…KSA…</route>
    <speedTram>
  </transportNetwork>
</city>
```

Listing 1. XML syntax example.

The most commonly used LandXML schema version is 1.2, adopted in 2008 [55]. This version is widely recognized as the standard LandXML and is compatible with numerous BIM applications. Format upgrade with property extensions and new entities was proposed in 2014, as LandXML 2.0, but was never widely adopted, therefore BIM platforms are generally not expected to be read and write this version.

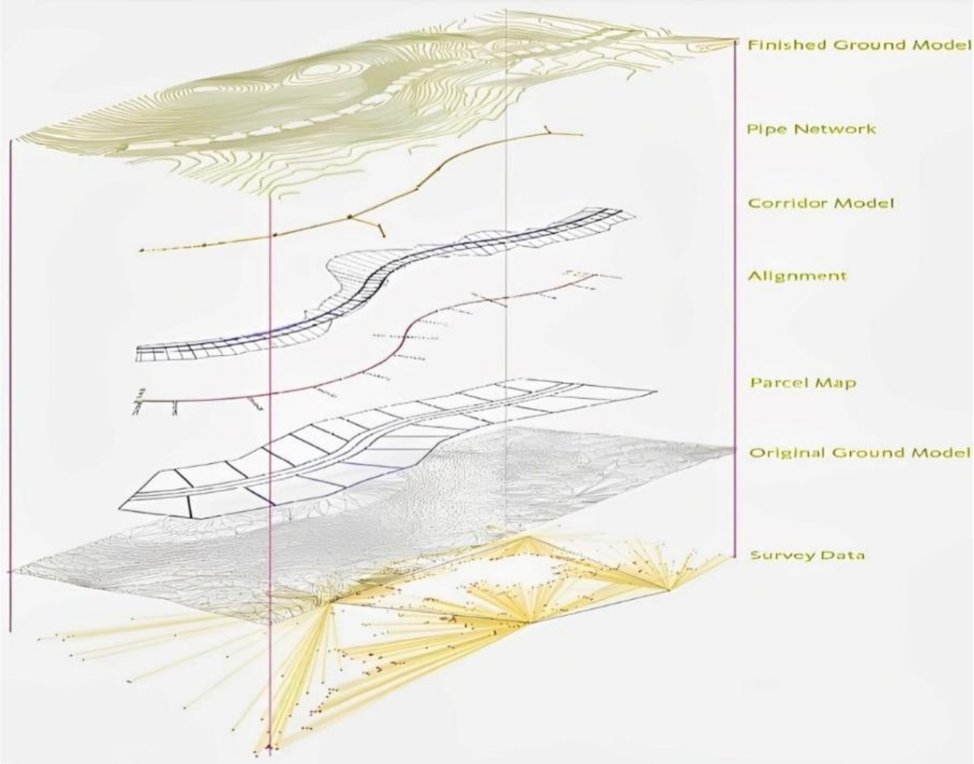


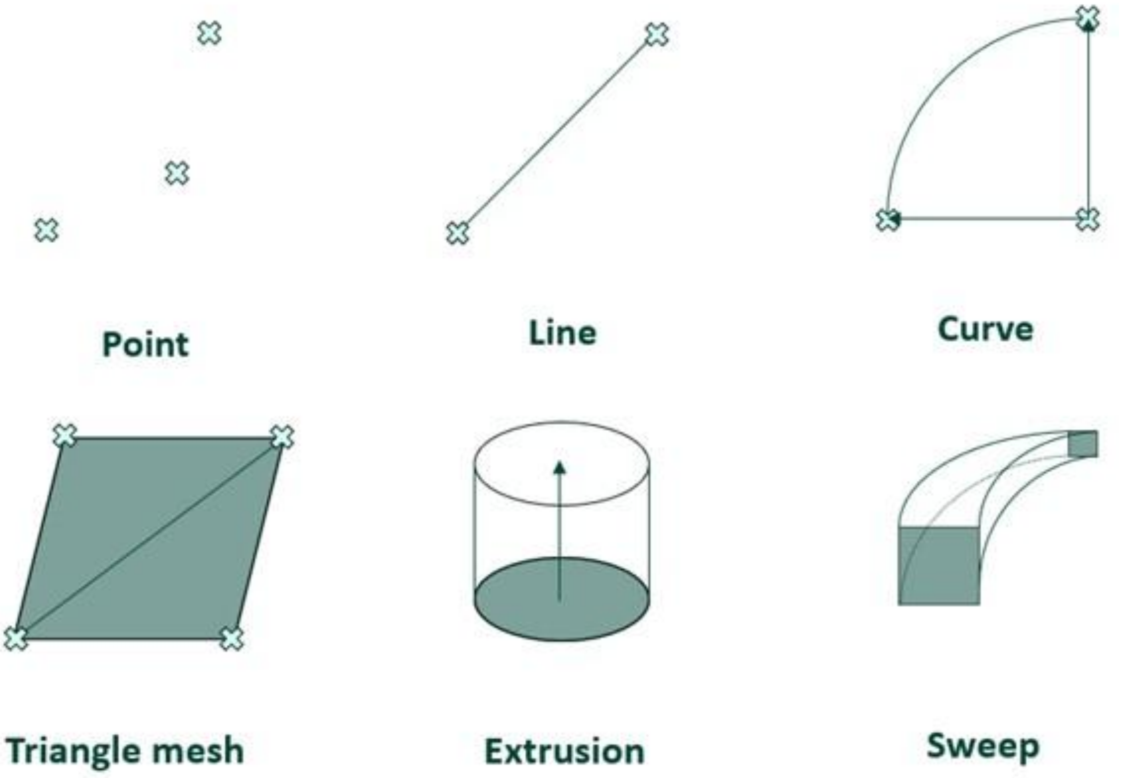
Figure 26. Land XML schema components [57].

BIM model written in LandXML can contain different data layers, as figure 26. demonstrates, and from the civil construction perspective the most relevant layers are:

- alignments
- ground models, also referred to as terrain models
- pipe networks
- survey data.

In practice, survey data is not always presented as a separate layer but rather embedded within a ground model or a pipe network layer.

Regarding the geometrical content, LandXML can store both volumeless and voluminous geometries, both of which are demonstrated in figure 27. However, the voluminous geometry encoding is limited to triangle meshes and extrusions. More complex solutions, such as Sweep Geometry, Boundary Representation (BREP), or Constructive Solid Geometry (CSG) representations, are beyond LandXML capabilities. [57.]



Figures 27. Volumeless (point, line, curve) and voluminous (mesh, extrusion, sweep) geometries

2.3.7.1.1 Alignments

An alignment generally represents a centreline of a road and is a fundamental element of road design, as it directly influences driving smoothness and safety [58]. In other types of infrastructure construction projects, alignments may instead define the centreline of a railway or a waterway, as well as be placed across the project area for referencing purposes.

A key concept in alignment referencing is the station, which refers to a specific point on the alignment and indicates its distance from the starting point. When an object is assigned an alignment reference, it means that a perpendicular line, can be drawn from the object’s origin to the alignment, intersecting it precisely at the given station. [58.]

From the geometry perspective, alignments are considered volumeless elements and composed of straight sections, also referred to as tangents, circular curves and spiral transition curves. Dimensionally, alignments can be either horizontal or vertical, and in combination they form the actual route. An example of such combination is illustrated in figure 28. The main difference between a line and a curve is that a line always represents the shortest straight distance between two points, whereas a curve is bent according to a given set of parameters. Transition curves, or spirals, used to smoothen the progression between tangents and circular curves in both horizontal and vertical alignments. [58.]

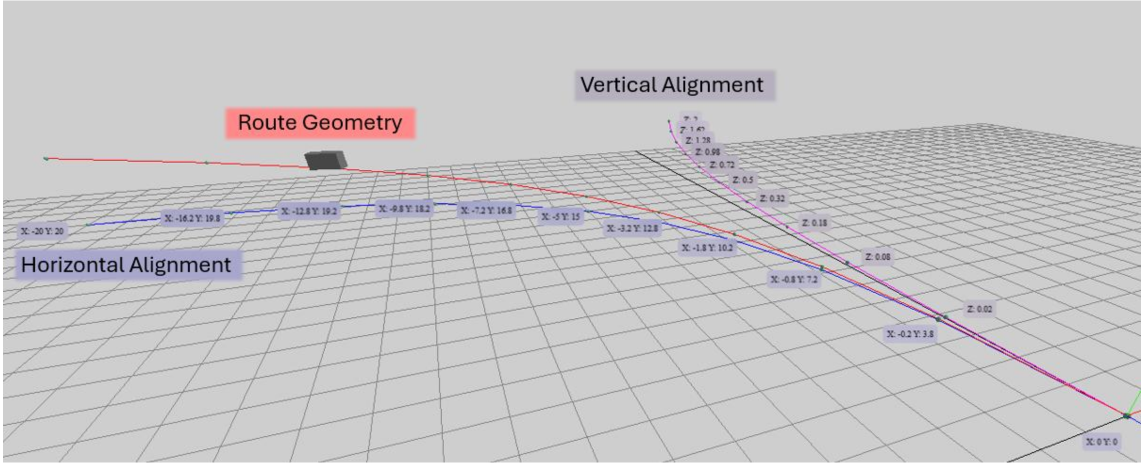


Figure 28. An alignment geometry visualisation example

Alignments generally belong to the design phase of a project and rarely updated with as-built points. Nevertheless, they can be attached to as-built data for referencing and quality control purposes. An example of an alignment entity encoded in LandXML format is illustrated in listing 2.

```
<Alignment name="route" length="31.4" staStart="0.0">
   <CoordGeom>
      <Curve staStart="0.0" rot="ccw" length="31.4" radius="20.0" chord="28.3"
dirStart="0.0" dirEnd="90.0">
         <Start>0.0 0.0</Start>
         <Center>-20.0 0.0</Center>
         <End>-20.0 20.0</End>
      </Curve>
   </CoordGeom>
</Alignment>
```

Listing 2. An alignment example written in LandXML

2.3.7.1.2 Terrain Models

Terrain models normally represent a surface course and are typically modelled as Triangulated Irregular Network (TIN) meshes, which consist of indexed vertices, and triangular faces, as figure 29 illustrates. The LandXML schema does not prescribe a specific triangulation algorithm, but Delaunay triangulation is commonly used to generate triangulated irregular networks. [59.] In practice, it is common to enrich terrain models with hingelines, which highlight area boundaries or surface folds. Hingelines can also serve as references for model-based quality control, enabling comparisons between different layers or versions. Regarding as-built terrain models, both vertices and hingelines can be obtained through land surveying.

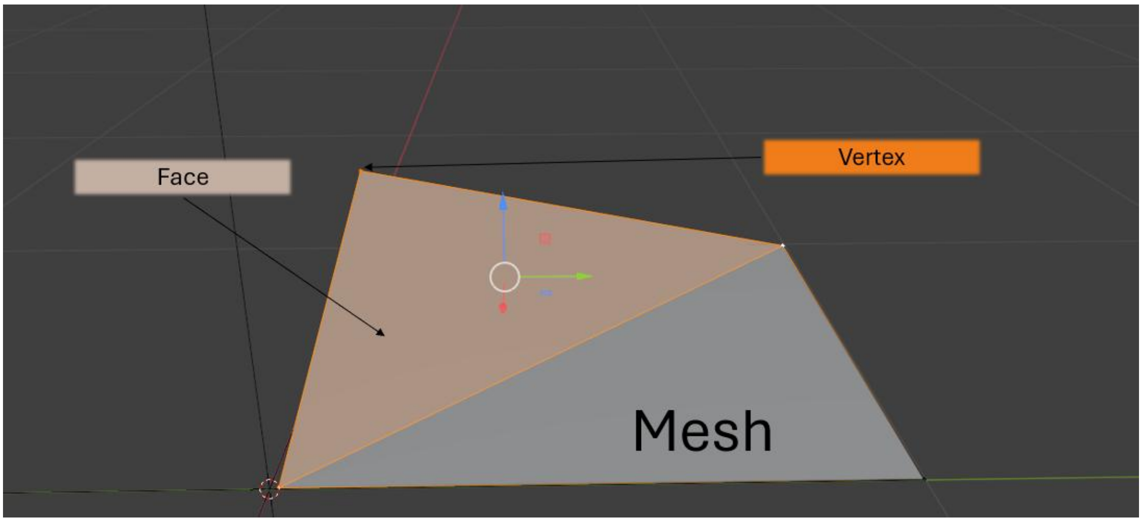


Figure 29. A triangulated mesh visualisation example

In the LandXML schema, triangulated meshes are stored within surface-classed entities, and there is no limitation on the number of surface elements within a single model. Similarly, shape complexity of surface geometry is not restricted, meaning that LandXML can potentially contain highly complex triangular surfaces, including both open and closed shells. However, in practice, certain software solutions may encounter difficulties in visualising atypical surface patterns, such as vertical, upside-down or overlapping triangles. An example of a TIN mesh encoded in LandXML format is illustrated in listing 3.

```
<Surface name="Plane" state="existing">
   <Definition surfType="TIN">
      <Pnts>
         <P id="1">0.0 0.0 0.0</P>
         <P id="2">0.0 1.0 0.0</P>
         <P id="3">1.0 1.0 0.0</P>
         <P id="4">1.0 0.0 0.0</P>
      </Pnts>
      <Faces>
         <F>1 2 3</F>
         <F>3 4 1</F>
      </Faces>
   </Definition>
</Surface>
```

Listing 3. A triangulated mesh example written in LandXML

2.3.7.1.3 Utility Lines

In LandXML, utility lines are generally represented as collections of elements called pipe networks [60]. Network types supported in schema version 1.2 include water, storm, sanitary and other, while the Inframodel standard expands this list with additional categories such as heating, cooling, gas, waste, electric and telecommunication [25]. A pipe network can contain two entity classes: struct, which represents a manhole, and pipe, as illustrated in figure 30. Both entities are modelled as extruded geometries with a circular hollow section. The main difference between them is that the struct elements are always vertical, whereas pipes may be tilted at any arbitrary angle.

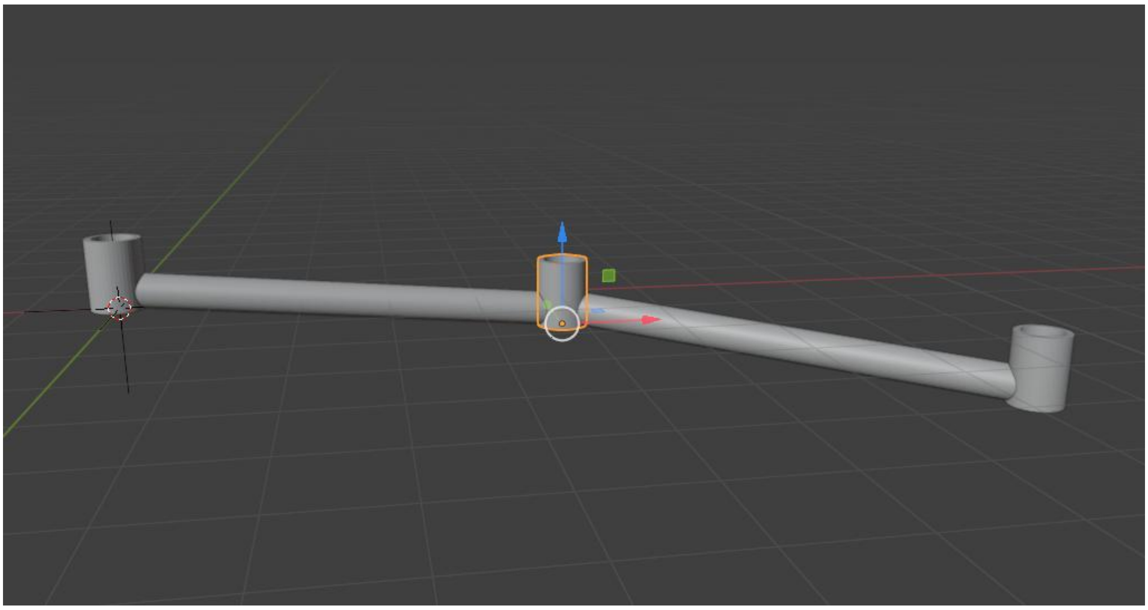


Figure 30. A pipe network visualisation example

As-built pipe and struct 3D objects can either be generated directly from land survey data, or through positional adjustments in the as-designed model, although the second option is generally more labour-intensive. A notable feature, introduced in the Inframodel standard, is the pipe elevation level parameter, which specifies the point in the pipe profile taken as the surveying reference. The allowed reference options are shown in figure 31.

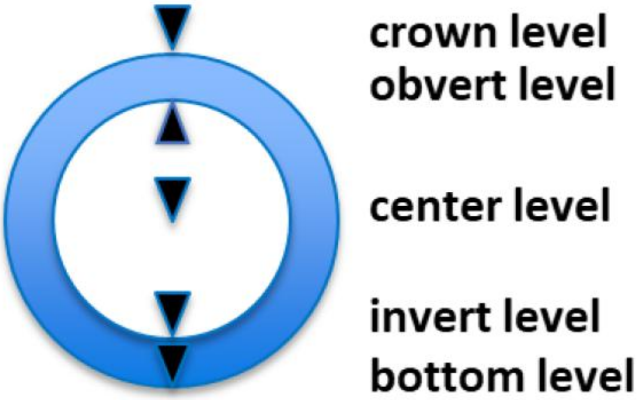


Figure 31. Elevation levels recognised in Inframodel Standard [25]

In Finland, a commonly accepted practice is to survey pressure-flow pipes from the crown and gravity-flow pipes from the invert. An example of a pipe network encoded in LandXML format is illustrated in listing 4.

```
<PipeNetwork name="Drainage" pipeNetType="storm" desc="rainwater">
   <Structs>
      <Struct name="Well_1" elevRim="2.0" elevSump="0.0">
         <Center>0.0 0.0</Center>
         <CircStruct diameter="1.5" material="concrete" thickness="0.1"/>
      </Struct>
   </Structs>
   <Pipes>
      <Pipe name="Pipe_1" length="12.0" slope="0.05">
         <CircPipe diameter="0.5" material="concrete" thickness="0.05"/>
      </Pipe>
   </Pipes>
</PipeNetwork>
```

Listing 4. A pipe network example written in LandXML

2.3.7.1.4 Metadata Structure

Additional information can be assigned to any entity within a LandXML model, using the feature container [60]. Original schema allows the container to hold an unlimited number of properties and values, although local specifications or standards, may introduce additional rules and limitations.

For instance, the Inframodel defines entity-specific properties while still allowing the assignment of zero or more user-defined property fields. An example of a feature container encoded in LandXML format is illustrated in listing 5.

```
<Feature code="IM_cable" source="inframodel">
     <Property label="label" value="internet"/>
     <Property label="owner" value="City Council"/>
     <Property label="networkType" value="data"/>
     <Property label="cableType" value="optic_fiber_120x"/>
     <Property label="radiusAround" value="0.05"/>
</Feature>
```

Listing 5. A property set xml example

2.3.7.2 IFC

Apart from the data exchange standard discussed in chapter 2.2.2, the term IFC also refers to the data exchange format itself, typically stored in files with the .ifc extension and written using STEP structured text. While alternative encodings such as ifcXML or ifcJSON also exist, IFC-STEP remains the most widely used and is explicitly recommended by buildingSMART International. [23, 61.]

Unlike XML, the STEP syntax may be less familiar to the general public. It originated from an older Product Data Exchange Specification and evolved into an independent product, formalised in 1994 as ISO 10303. After being adopted by CAD applications, it soon became standard exchange format for CAD [62]. A distinctive feature of STEP is the use of an express ID at the beginning of each line, which serves as a unique identifier within the file. An example of IFC-STEP syntax is illustrated in listing 6.

```
#10=ORGANIZATION('KSA', #14 );
#11=PRODUCT('KS', #12, #13);
#12=PRODUCT_DEFINITION('SpeedTramLine');
#13=PRODUCT_CONTEXT('HKI');
#14=ORGANIZATION_ROLE('contractor');
```

Listing 6. IFC-STEP syntax example.

The most recent official IFC schema is 4.3.2.0, published in 2024 [63], and as discussed earlier, version 4.3 was the first to comprehensively include the infrastructure domain. While earlier IFC versions were known for their limited support of civil construction elements, IFC4 or IFC2X3 files intended for civil construction implementation are still encountered in practice [64.].

2.3.7.1.5 Entities

Entity is a fundamental unit of the IFC hierarchy, that can represent an element, a property or a relationship. The IFC schema is comprised of several hundred standardised entities, covering both physical and abstract assets. [65.]

Physical entities correspond to modellable objects, such as walls, or pipes, whereas abstract entities represent organisational or hierarchical concepts, such as a project or an element assembly. An example of a Pipe Segment entity encoded in IFC-STEP format is illustrated in listing 7.

```
/*origin - XYZ*/
#50=IFCCARTESIANPOINT((25497535.000, 6674040.000, 1.000));
/*profile axis X*/
#51=IFCDIRECTION((5., 0., 0.));
/*extrusion axis Z*/
#52=IFCDIRECTION((0., 0., 5.));
/*placement – point, X-axis, Z-axis*/
#53=IFCAXIS2PLACEMENT3D(#50, #51, #52);
/*placement related to (#10)*/
#54=IFCLOCALPLACEMENT(#10, #53);
/*profile – hollow circle*/
#55=IFCCIRCLEHOLLOWPROFILEDEF(.AREA., 'CHS', $, 0.32, 0.07);
/*extrusion – profile, axis, direction, distance*/
#56=IFCEXTRUDEDAREASOLID(#55, #9, #5, 5.0);
/*representation – context…*/
#57=IFCSHAPEREPRESENTATION(#16, 'Body', 'SweptSolid', (#56));
/*shape – representation*/
#58=IFCPRODUCTDEFINITIONSHAPE($, $, (#57));
/*entity – GUID, history, name, description…*/
#59=IFCPIPESEGMENT('1FUNwS0XGbyaWaqAcSmotV', #27, 'HV1.1', 'STORM_DRAIN', $,
#54, #58, $, .RIGIDSEGMENT.);
```

Listing 7. A pipe element ifc example

2.3.7.1.6 Entity Relations and Metadata Structure

Relational hierarchy plays a key role in IFC schema. Relationships between the entities in IFC can take multiple forms – such as contains, aggregates, voids, associates, assigns or nests – each designed for a specific use case [66]. An example of relational entities encoded in IFC-STEP format is illustrated in listing 8.

```
/* distribution system*/
#40=IFCDISTRIBUTIONSYSTEM('7XUNwwefYXDFyaWaqEGSm7',$,'STORMWATER',$,$,$,$);
/*element assembly – pipe group*/
#42=IFCELEMENTASSEMBLY('0Q1tCJWdj4kOkZUg7rkf2h',$,'PIPELINE',$,$,$,$,$,$,$);
/*assign element assembly to distribution system*/
#43=IFCRELASSIGNSTOGROUP('3mLU8Y3216GwFmiFrWMIV0',$,$,$,(#42),$,#40);
/*pipe element*/
#59=IFCPIPESEGMENT('0XUNwwefgdXGbyaWaqAcSm',$,'PIPE.001',$,$,#54,#58,$,$);
/*assign pipe element to assembly*/
#100=IFCRELAGGREGATES('1WdB196Kb72f_pKgj5rklU',$,'ELEMENTASSEMBLY',$,#42,(#59)
);
```

Listing 8. A relational hierarchy ifc example

The IFC schema allows assigning meta-properties to an entity, either by defining a type or by attaching a property set. Standardized, entity-specific property sets are to begin with the prefix Pset, while user-defined properties may use arbitrary names [67]. An example of a property set encoded in IFC-STEP format is illustrated in listing 9.

```
/*property name and value*/
#185=IFCPROPERTYSINGLEVALUE('ClimateChangePerUnit',’CO2/kg’,IFCMASSMEASURE(2.7
),#2);
/*property set contains value #185*/
#186=IFCPROPERTYSET('2eZ8S9SeedH4xgguFlcLIk',$,'Pset_EnvironmentalImpactIndica
tors',$,(#185));
/*assign property set to element #59*/
#187=IFCRELDEFINESBYPROPERTIES('tlaaQApHVmtQsskPj9i4o', $, $, $, (#59), #186);
```

Listing 9. A property set ifc example

2.3.7.1.7 Geometry

Geometric representation is a significant aspect of the IFC schema and requires special attention. The schema describes over 30 shape representation types, including very complex geometries [67]. However, it is generally recommended to use simpler geometries, such as meshes, when possible.

Geometries emphasising centreline – such as sweep along curve – can sometimes be misinterpreted by BIM software, resulting in inaccurate structural analysis [68]. An example a profile swept along the curve is illustrated in figure 32.

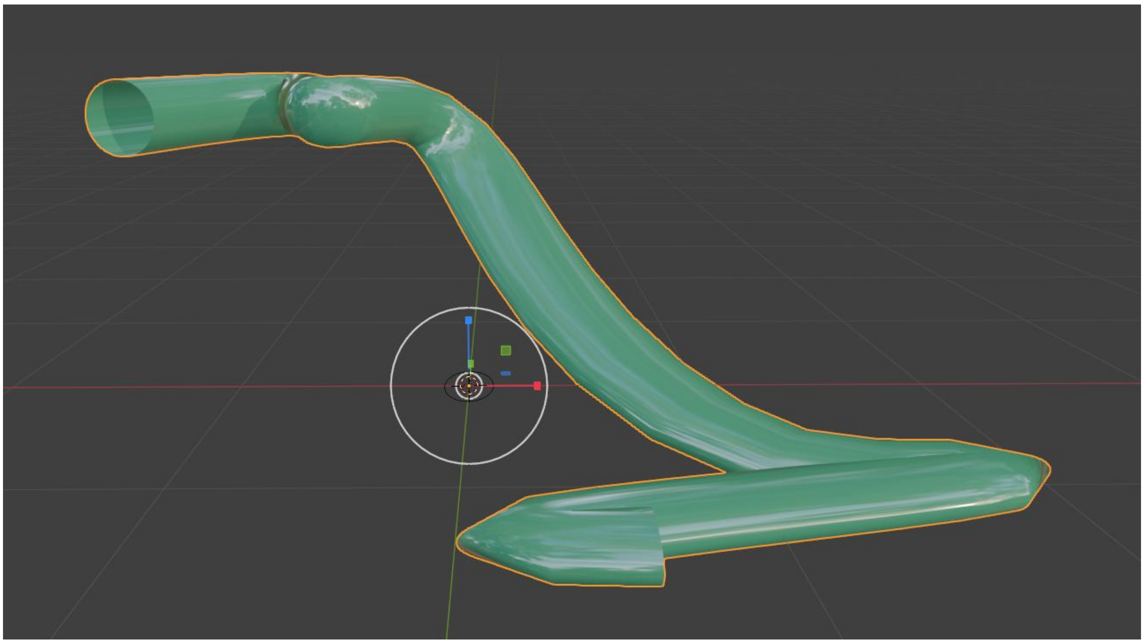


Figure 32. A profile swept along the curve visualisation example

2.3.7.3 CSV, JSON, GT

Comma-Separated Values (CSV), JavaScript Object Notation (JSON), and GeoTime (GT) are not exactly part of the BIM domain but are successfully used in the industry. Table-like, plain text structure makes them lightweight, scalable and easy to implement in land surveying and vehicle control applications.

The GT format, also known as Geonic, carried in files with .gt extension, deserves An example of a profile swept along the curve is illustrated special attention. It’s a locally used format, initially developed in 1987 for Finnish Traffic Infrastructure Agency [69]. Closest international counterpart is .prn, a space-delimited formatted text, which technically makes .gt compatible with Excel. The format is limited to simple line or point-level geometries, which is usually sufficient for storing and transferring land survey data. GT is extensively applied across Finland and is often specified in the municipal-level data delivery requirements. An example of survey data encoded in GT format is illustrated in listing 10.

```
Surface Line/Point Entity  UUID    Northing       Easting        Elevation
 331000        0      97   0001  6677132.756   2554210.952          7.459
 331000        0      97   0002  6677132.847   2554209.549          7.537
 331000        0      97   0003  6677121.180   2554207.073         10.273
 331000        0      97   0004  6677255.539   2554434.570         16.103
 331000      101    110k   0011  6677095.630   2554136.431          8.716
 331000      101    110k   0012  6677097.182   2554147.189          8.227
 331000      101    110k   0013  6677098.268   2554164.049          7.683
 331000      101    110k   0014  6677100.898   2554181.102          7.310
 331000      102    110k   0021  6677099.749   2554142.262          8.549
 331000      102    110k   0022  6677100.635   2554155.809          8.035
 331000      102    110k   0023  6677102.206   2554172.439          7.585
 331000      102    110k   0024  6677105.424   2554188.346          7.347
```

Listing 10. GT syntax example

2.3.7.4 Relationship between LandXML and IFC

According to buildingSMART Finland, IFC is intended to eventually replace LandXML in the civil construction domain [70]. However, since LandXML was specifically developed to meet domain-specific need, it may take considerable time before IFC can effectively cover all LandXML's capabilities. The two schemas are not parallel: while IFC is comprised of more than 700 building and facility-related entities, the LandXML defines just over 200 civil engineering elements [71]. Furthermore, their internal relational hierarchies differ, meaning that proper conversion between the formats requires a thorough understanding of both schemas.

As for the GT and CSV, they are not designed to replace IFC or LandXML. Instead, they may serve as supplementary formats, particularly useful when data transferring options are limited or when the required level of information need is narrowed down to volumeless representations. Regarding JSON, it is worth noting that the upcoming major IFC schema upgrade, known as IFC 5, has been announced to adopt JSON as primary syntax, marking a step away from dependency on the STEP encoding [72].

3 Review of related work

3.1 Case Selection

A structure gauge analysis report, referred to in Finnish as ATU-tarkasteluraportti, was selected for related work analysis in this thesis. Its relevance is supported by several factors. The analysis was conducted for Jokeri Light Rail project, which highly compares to the Crown Bridges Light Rail project: both involve light rail infrastructure construction in Finland, implemented only few years apart, under the same regulatory framework, by the same consortium of contractors, and for the same project owner.

Furthermore, Jokeri ATU analysis was based on point cloud data acquired through laser scanning, forming a part of the project’s as-built dataset. A similar structure gauge analysis was later performed for the Crown Bridges project. However, this time as-built models were used as the reference instead of scan data. In addition, the as-built modelling workflow for the Crown Bridges project was strongly influenced by the conclusions drawn from the Jokeri project, which further underlines the value of the case study for the present research.

3.2 Project Description

Jokeri Light Rail is a speed-tram line that connects the areas of Keilaniemi in Espoo and Itäkeskus in Helsinki, resulting in 25 kilometres of newly built environment [73]. The route alignment is illustrated in figure 33. The as-built data collection for the project involved both land surveying and reality capture technologies. Land surveying was applied to verify the positioning of structures and layers, as well as to calculate possible deviations from the designed coordinates. Volumeless as-built models were produced by refining the survey data and enriching it with InfraRAK and InfraBIM codes. The final as-built dataset was handed over to the project owners as part of the standard delivery process.

Since no comprehensive 3D as-built modelling was done for Jokeri Light Rail, the project management sought alternative ways of obtaining 3D representations of the built infrastructure. This need was primarily driven by the fact, that the clearance gauge analysis for the rolling stock must be based on 3D data.

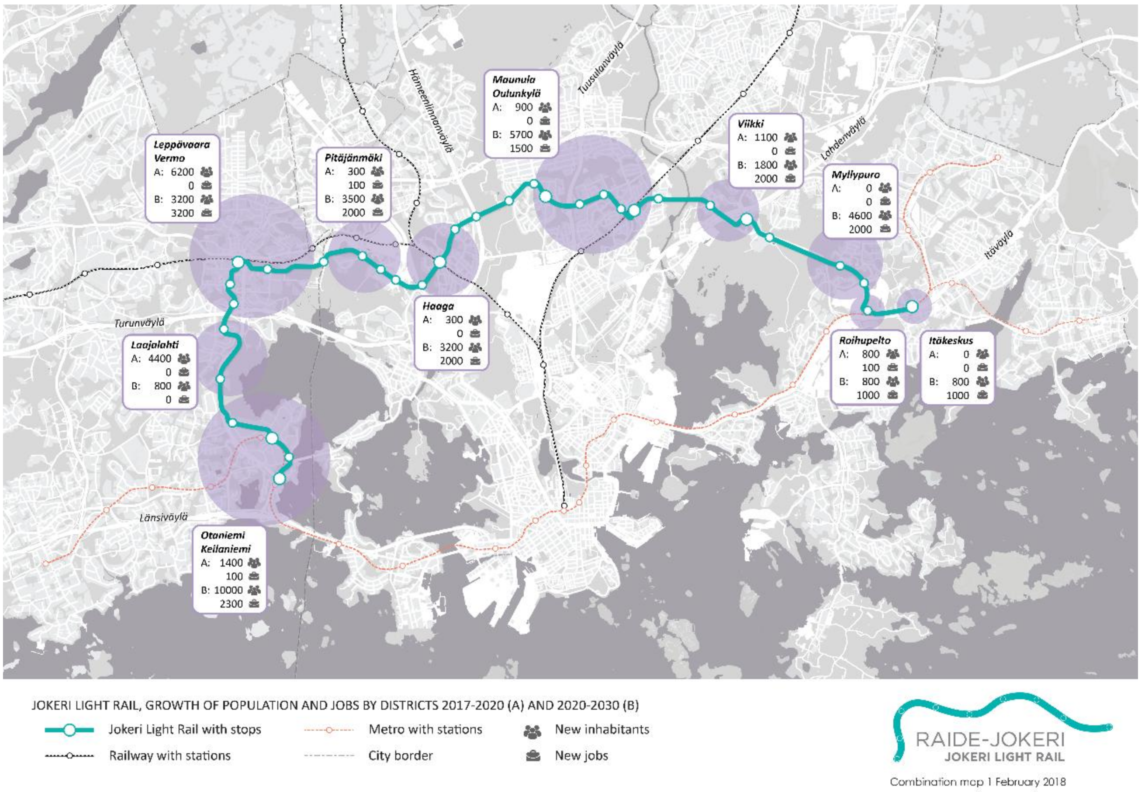


Figure 33. Jokeri Light Rail route on a map [73]

3.3 Applied Methodology

3.3.1 Structure Gauge Analysis

Structure Gauge Analysis (ATU) is conducted to ensure rolling stock can operate without encountering any physical obstacles along the route. Two key concepts that support this analysis are the structure gauge, which defines the minimal allowed distance of surrounding objects from the track, and the kinematic envelope, which specifies the space within which the streetcar is allowed to move laterally and vertically. [74.] A general collection of clearance levels applied in railways is demonstrated in figure 34.

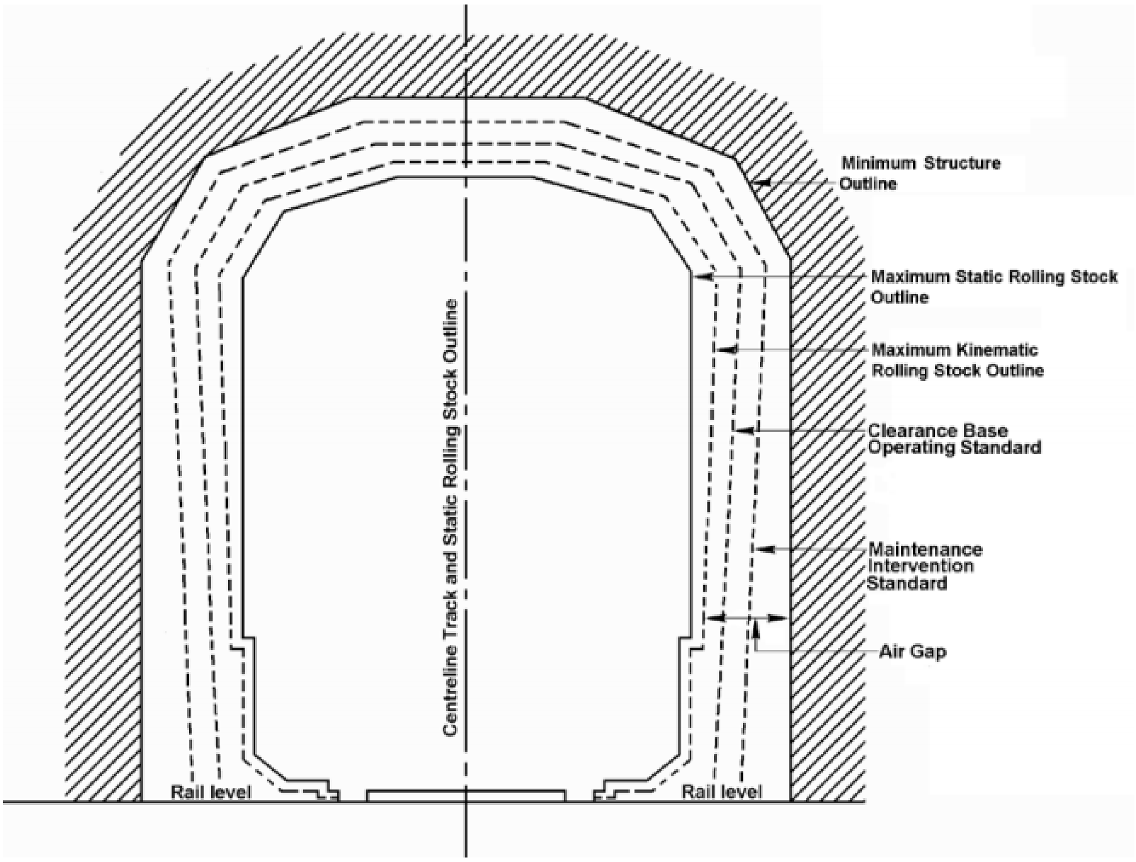


Figure 34. Rolling stock clearance levels [74].

Patterns used for the ATU analysis in the Jokeri Light Rail project are presented in figure 35. The inner pattern, highlighted in pink, represents the kinematic envelope and corresponds to the Maximum Kinematic Rolling Stock Outline in figure 34. The outer pattern, shown in yellow, represents the structure gauge and corresponds to the Clearance Base Operating Standard of the same clearance level collection.

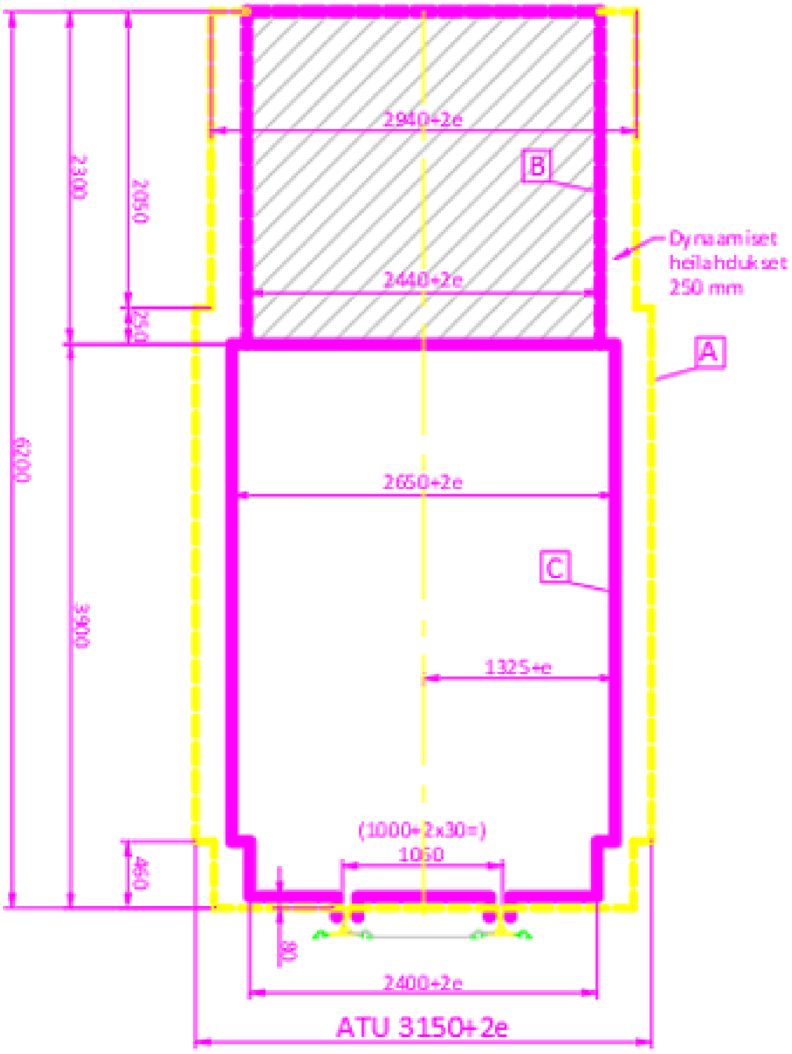


Figure 35. Structure gauge pattern used for Jokeri Light Rail line [75].

The dimensions of both structure gauge and kinematic envelope are determined by the route profile, and the clearance envelope models are created based on the track alignment. In particular, curved segments of the track require increased lateral clearance, whereas vertical clearance generally remains the same. [76.] Example of an envelope model is illustrated in figure 36.

Clearance envelope models were created for the entire Jokeri Light Rail track network, including the depot areas. The models included two levels of clearance: maximum static rolling stock outline, used particularly at low-speed locations such as tram stops, and clearance base operating standard, which covered the general sections of the route. [75.]

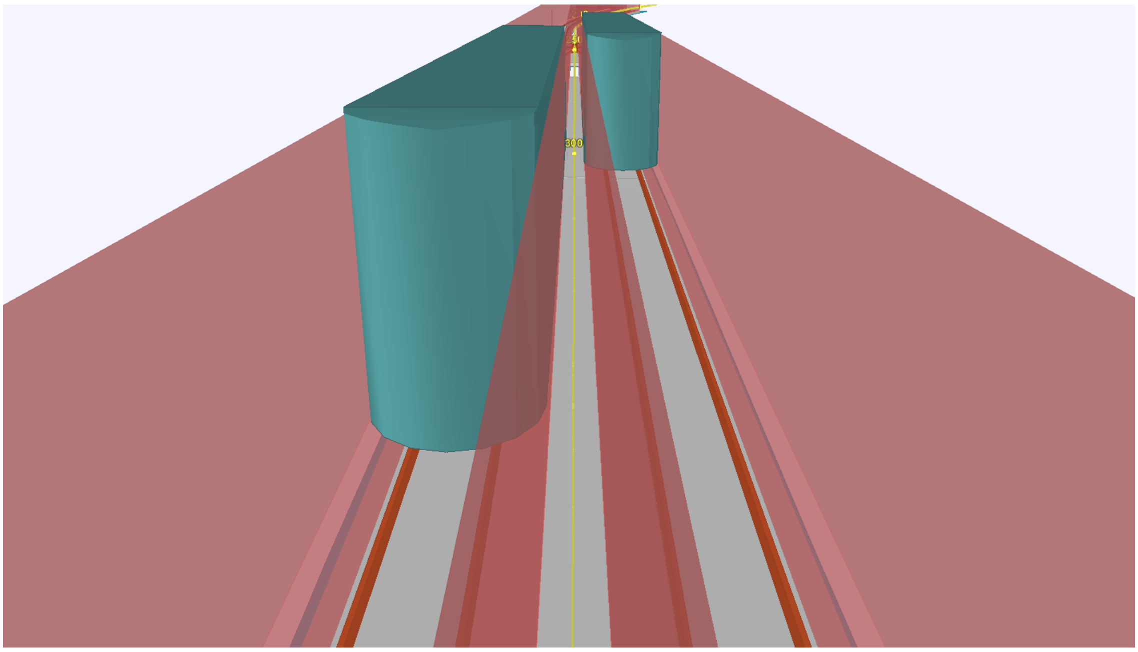


Figure 36. Clearance envelope model visualised along the tram hull (green), tracks (red), and the alignment (yellow)

3.3.2 Laser Scanning

Reality capture through laser scanning and point cloud processing was described earlier in chapter 2.3.4. The Jokeri Light Rail project scanning involved particular challenges, as its route combined long stretches in non-residential zones with low-speed sections running through densely built urban areas, and therefore required a specifically tailored approach.

A Trimble MX9 mobile mapping system, known for its efficiency in railway mapping, was selected for this task. To ensure smooth and stable transportation of the scanner along the tracks, it was mounted on a rail quad bike – an inventive, yet creative solution. The complete setup is illustrated in figure 37. The MX9 mapping system’s scanning rate of 500 scans per second allowed the quad bike to navigate through the route at relatively higher speeds, while the GNSS lateral accuracy of 0.020 m fulfilled the minimum requirements for construction-grade surveying equipment. [77.]

Figure 37. Trimble MX9 Mobile mapping system mounted on a rail quad bike

3.4 Structural Gauge Analysis Results

Post-processing of Jokeri Light Rail scanning data was performed using Trimble Business Center and TerraScan applications, involving point cloud examination, cleaning and georeferencing [77]. The resulting product was a dense point cloud in LAS format, a part of which is demonstrated in figure 38.

Figure 38. Post-processed Jokeri Light Rail point-cloud obtained by mobile scanning process [77].

Once the point cloud for Jokeri Light Rail was created, structural clearance frames were applied on the point cloud within the Microstation software platform. The final point cloud examination identified a series of locations where either permanent or temporary elements intruded into the clearance envelopes, as shown in figure 39. Based on these findings, corrective actions were taken to ensure the track readiness for rolling stock test runs. [75.]

Figure 39. Permanent and temporary traffic signs intruding into the clearance envelope space [75].

3.5 Results Overview

Despite all the overall success of the applied workflow, further examinations revealed several issues. Point cloud-based analysis failed to detect certain minor imperfections in tram stop structures, which only became apparent during the test runs and triggered a set of corrective actions.

Another significant issue concerned the size and manageability of the point cloud datasets. Their storage and processing required substantial resources, which limited their long-term applicability. Primary value for the dataset was found in warranty-related operations, as it captured the state of built environment at the time of commissioning. Once corrective measures had been applied, the point cloud no longer reflected the actual condition of the asset and updating it would have required a new resource-intensive scanning operation.

Consequently, in the Crown Bridges Light Rail project, point cloud data was not used for structure gauge analysis. Instead, an as-built modelling-based approach was adopted. A more detailed description of this workflow along with the methods comparison is presented in chapter 4.2 of this thesis.

4 Case study research

This case study focuses on examination of as-built models. The model dataset, produced for the Crown Bridges Light Rail project in Helsinki, includes 3D geometries, metadata, and supporting documentation. The applicability of the model data is evaluated from four perspectives: initial data, operation and maintenance, regulatory compliance, and auxiliary applications.

Rather than relying on a single approach, this thesis applies a mixed-method research design. The investigation combines qualitative, quantitative, comparative and exploratory methods. This methodological diversity enables a comprehensive evaluation of both established and emerging applications of as-built information models, as well as exploration of potential future use cases.

The scope of this research entails following testing categories:

- Geometry and metadata readability across platforms
- Suitability for structure gauge analysis
- Digital Twin readiness
- Suitability for AI driven examinations.

4.1 Geometry and Metadata Readability across Platforms

Geometry and metadata readability examination combined qualitative and quantitative research methods. The qualitative component was focused on overall compatibility of the model in given format with given platform. The quantitative component, in turn, evaluates whether all geometries are rendered correctly, whether location is interpreted accurately, and whether all embedded metadata fields are displayed, thereby identifying potential data loss.

The analytical procedures described above are primarily human-driven. They include manual model import, visual assessment of geometry and elements distribution across the scene, visual inspection of metadata attributes, systematic documentation of observations, and finally, the formulation of and appraisal and summary.

4.1.1 Research Data

The as-built models used for this study were from the Crown Bridges Light Rail project, where they were originally produced as a part of the project’s quality assurance and delivery process. Since higher priority was assigned to underground structures that would become concealed by the time of handover, the test models were selected primarily from the substructure as-built cluster. The structures included in this examination are:

- Piles (driven and bored),
- Pile caps (concrete slabs)
- Cable Ducting Network (including manholes and light poles),
- Water Supply and Drainage Networks (including manholes)

For the research clarity, the model data was geographically clustered. Substructure and utility line models were selected from a densely built crossroad area, resulting in a clear yet representative composition. A schematic 2D view of the above listed assets is illustrated in figure 40.

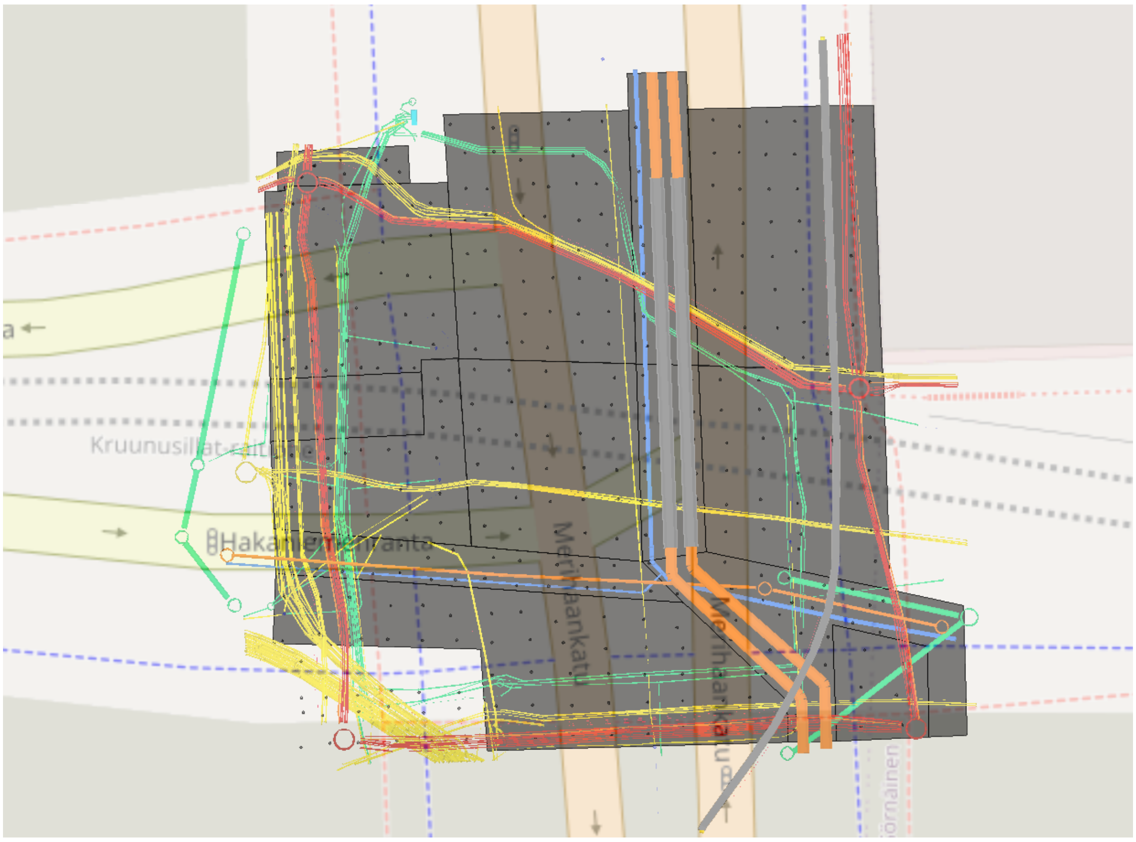


Figure 40. A schematic view of the substructure model combination

Although, LandXML written in accordance with Inframodel version 4.2 specification, was the primary accepted format for as-built deliverables in the Crown Bridges Project, for research purposes the substructure models were taken in two parallel versions: LandXML and IFC 4.3. This dual-format approach enables a broader evaluation of as-built models applicability. Both versions contained identical 3D geometries, locations, and metadata sets.

Each test model includes an extensive metadata set covering the following attributes:

- InfraRAK code and description
- InfraBIM code and description
- Element Material
- Element Label
- Construction Date
- Elevation level (for pipe elements)

- Related Documents connected through External Link
- Network type or Structure type
- Asset Identifier
- Asset Delivery Status.

As discussed above, the schemas of the two formats are not directly parallel. Therefore, since the Inframodel standard was prioritised in the project, the metadata presets defined in the IM4.2 documentation were integrated into the IFC context, rather than vice versa. The correspondence of metadata field embeddings in the testing models is illustrated in table 1.

In the IFC versions, some metadata attributes were intentionally embedded in both schema-compliant and arbitrary ways. This redundancy was introduced strictly for research purposes to compare different embedding strategies, and is not recommended in practice, as it promotes data duplication and increases the file size.

Table 1. Comparison of corresponding metadata syntax in LandXML and IFC

Meta data	LandXML embedding	IFC embedding

InfraRAK, InfraBIM code and description	`<Feature code="IM_coding" source="inframodel">`
	`<Property label="terrainCoding" value="S300B"/>`
	`<Property label="terrainCodingDesc" value="Hulevesi 300mm betoni"/>`
	`<Property label="surfaceCoding" value="312000"/>`
	`<Property label="surfaceCodingDesc" value="Hulevesiviemarit"/>`
	`</Feature>`
		`#1225=IFCPROPERTYSINGLEVALUE('infraCoding', $, IFCLABEL('S300B'), $);`
		`#1226=IFCPROPERTYSINGLEVALUE('infraCodingdesc', $, IFCLABEL('Hulevesi betoni 300mm'), $);`
		`#1228=IFCPROPERTYSINGLEVALUE('surfaceCoding', $, IFCLABEL('312111'), $);`
		`#1229=IFCPROPERTYSINGLEVALUE('surfaceCodingdesc', $, IFCLABEL('Hulevesiviemarit viettoviemari betonista'), $);`
		`#1231=IFCPROPERTYSET('1ipV2GrPyGpqx6l6r49kpy', $, 'IM_coding', 'Properties', (#1225, #1226, #1228, #1229));`

Element Material	`<CircPipe material="concrete"/>`	`#1223=IFCMATERIAL('CONCRETE_', '', 'CONCRETE');`

Element Label, Construction Date, Elevation level (for pipe	<Feature code="IM_pipe" source="inframodel"> <Property label="pipeLabel" value="HV"/> <Property label="elevType" value="invert level"/> <Property label="constuctionDate" value="2025-11-28"/> </Feature>	#1236=IFCPROPERTYSINGLEVALUE('pipeLabel', $, IFCLABEL('HV'), $); #1237=IFCPROPERTYSINGLEVALUE('elevType', $, IFCLABEL('invert level'), $); #1241=IFCPROPERTYSET('1U0CbZ8EsHR1W16MiL_Y7T', $, 'IM_pipe', 'inframodel', (#1236, #1237)); + #1023=IFCPROPERTYSINGLEVALUE('InstallationDate', $, IFCLABEL('2025-11-28'), $); #1025=IFCPROPERTYSET('14qkb2HXIFWe2dPpi0S_1J', $, 'Pset_ConstructionOccurence', $, (#1023));
Asset Identifier, Asset Status, and Related Documents (External Link)	<Feature code="IM_plan" source="inframodel"> <Property label="planName" value="HKI-VHT"/> <Property label="planCode" value="101112"/> <Property label="planState" value="DELIVERED"/> <Property label="planDesc" value="https://bit.ly/3PI29Qc"/> </Feature>	#1028=IFCPROPERTYSINGLEVALUE('PlanName', $, IFCLABEL('HKI-VHT'), $); #1029=IFCPROPERTYSINGLEVALUE('PlanState', $, IFCLABEL('DELIVERED'), $); #1030=IFCPROPERTYSINGLEVALUE('PlanDescription', $, IFCLABEL('https://bit.ly/3PI29Qc'), $); #1031=IFCPROPERTYSET('1oemQjlgJIxqVk5BvpDh6a', $, 'IM_plan', $, (#1028, #1029, #1030)); + #1024=IFCPROPERTYSINGLEVALUE('AssetIdentifier', $, IFCLABEL('101112'), $); #1025=IFCPROPERTYSET('14qkb2HXIFWe2dPpi0S_1J', $, 'Pset_ConstructionOccurence', $, (#1024));
Network type	<PipeNetwork name="STORM" pipeNetType="storm">	#15=IFCFACILITYPARTCOMMON('1ok4jCjfWFeydZZDkyXG6W', $, ' STORMWATER_DRAINAGE', 'Pipe Network description', $, #8, $, $, $., $, $);

All models used for this research category were created using Plainview software and based on land survey data. The corresponding as-designed models were used as reference for comparison. For practicality, all external links included in model metadata were shortened using bit.ly service, while the linked documentation itself was stored in the M-Files database.

The geometry in the test models was deliberately kept relatively simple and easy to interpret. The LandXML models contained extruded geometry for pipe and struct elements, as well as TIN meshes for the hull elements. The IFC models, in turn, contained extruded geometry for pipe and manhole elements, and polygonal face tessellation for hull objects. The location of all elements was specified using global coordinates, with global origin set in accordance with common infrastructure construction practices in Finland, thereby eliminating need for map conversion.

4.1.2 Research Framework

Software platform selection for the model testing was guided by the relevance to the civil construction branch, general availability, and degree of prominence within civil construction industry in Finland. The chosen solutions were divided into four functional groups:

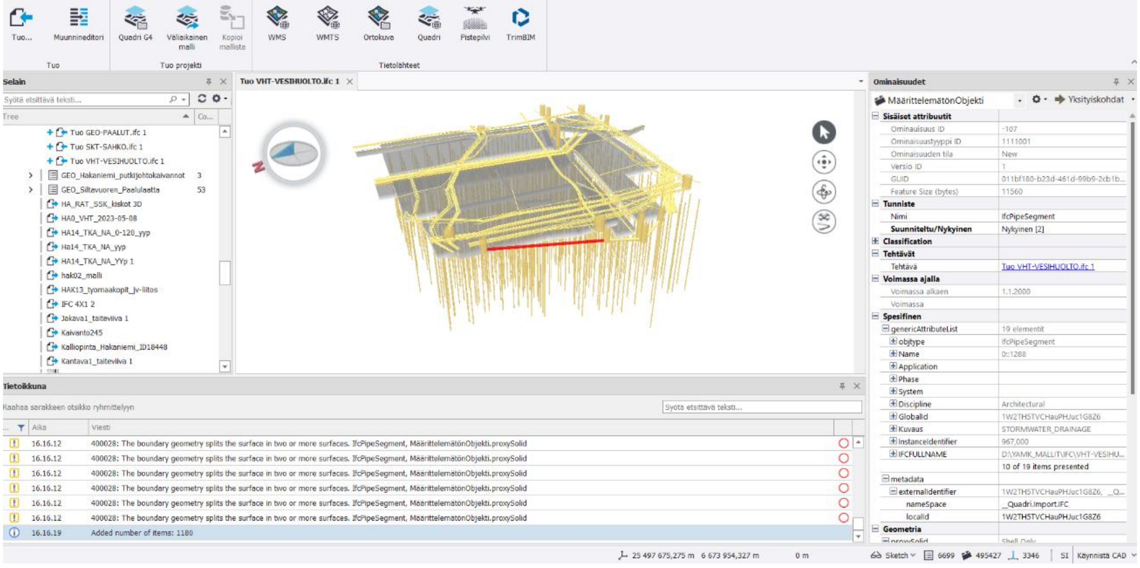
- Modelling Applications
- Model Viewers
- Land Survey and Site Management
- Asset Management and Maintenance.

It should be noted that this research chapter does not attempt to evaluate the full capability spectrum of each platform. Instead, the focus is limited to features directly relevant to the defined research methods

4.1.2.1 Modelling Applications

4.1.2.1.1 Novapoint

Novapoint is an infrastructure design platform from the Trimble product family. It was selected for testing due to its use at the design stage in both Jokeri and Crown Bridges light rail projects. With respect to this study, the platform supports the import of both LandXML and IFC.



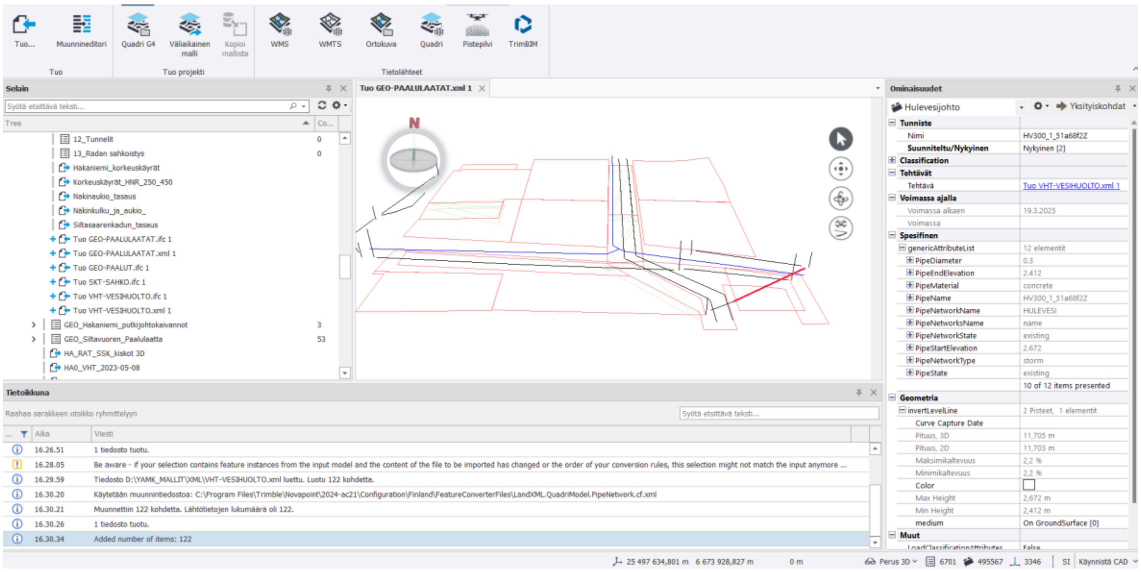
Sisäiset attribuutit	
Ominauisuus ID	-107
Ominaisuustyyppi ID	1111001
Ominaisuuden tila	New
Versio ID	1
GUID	011bf180-b23d-461d-99b9-2cb1b...
Feature Size (bytes)	11560
Tunniste	
Nimi	IfcPipeSegment
Suunniteltu/Nykyinen	Nykyinen [2]
Classification	
Tehtävät	
Tehtävä	Tuo VHT-VESIHUOLTO.ifc 1
Voimassa ajalla	
Voimassa alkaen	1.1.2000
Voimassa	
Spesifinen	
genericAttributeList	19 elementit
objtype	IfcPipeSegment
Name	0::1288
Application	
Phase	
System	
Discipline	Architectural
GlobalId	1W2THSTVCHauPHJuc1G8Z6
Kuvaus	STORMWATER_DRAINAGE
InstanceIdentifier	967,000
IFCFULLNAME	D:\YAMK_MALLIT\IFC\VHT-VESIHU...
	10 of 19 items presented
metadata	
externalIdentifier	1W2THSTVCHauPHJuc1G8Z6, _Q...
nameSpace	_Quadri.Import.IFC
localId	1W2TH5TVCHauPHJuc1G8Z6
Geometria	

```
Aika      Viesti
16.16.12  400028: The boundary geometry splits the surface in two or more surfaces. IfcPipeSegment, MäärittelemätönObjekti.proxySolid
16.16.12  400028: The boundary geometry splits the surface in two or more surfaces. IfcPipeSegment, MäärittelemätönObjekti.proxySolid
16.16.12  400028: The boundary geometry splits the surface in two or more surfaces. IfcPipeSegment, MäärittelemätönObjekti.proxySolid
16.16.12  400028: The boundary geometry splits the surface in two or more surfaces. IfcPipeSegment, MäärittelemätönObjekti.proxySolid
16.16.12  400028: The boundary geometry splits the surface in two or more surfaces. IfcPipeSegment, MäärittelemätönObjekti.proxySolid
16.16.12  400028: The boundary geometry splits the surface in two or more surfaces. IfcPipeSegment, MäärittelemätönObjekti.proxySolid
16.16.19  Added number of items: 1180
```

```
25 497 675,275 m   6 673 954,327 m   0 m
```

Figure 41. IFC test models in Trimble Novapoint 2025

IFC import in Novapoint was straightforward, with geometries rendered correctly and all necessary detail preserved, as shown figure 41. In contrast, LandXML import required additional adjustments and produced difficulties in visualising both extruded and mesh objects, resulting in wireframe representations, as demonstrated in figure 42.



Tunniste	
Nimi	HV300_1_51a68f2Z
Suunniteltu/Nykyinen	Nykyinen [2]
Classification	
Tehtävät	
Tehtävä	Tuo VHT-VESIHUOLTO.xml 1
Voimassa ajalla	
Voimassa alkaen	19.3.2025
Voimassa	
Spesifinen	
genericAttributeList	12 elementit
PipeDiameter	0.3
PipeEndElevation	2.412
PipeMaterial	concrete
PipeName	HV300_1_51a68f2Z
PipeNetworkName	HULEVESI
PipeNetworksName	name
PipeNetworkState	existing
PipeStartElevation	2.672
PipeNetworkType	storm
PipeState	existing
	10 of 12 items presented
Geometria	
invertLevelLine	2 Pisteet, 1 elementit
Curve Capture Date	
Pituus, 3D	11,705 m
Pituus, 2D	11,703 m
Maksimikaltevuus	2,2 %
Minimikaltevuus	2,2 %
Color	☐
Max Height	2,672 m
Min Height	2,412 m
medium	On GroundSurface [0]
Muut	
LoadClassificationAttributes	False

```
Aika      Viesti
16.26.51  1 tiedosto tuotu.
16.28.05  Be aware - If your selection contains feature instances from the input model and the content of the file to be imported has changed or the order of your conversion rules, this selection might not match the input anymore ...
16.29.59  Tiedosto D:\YAMK_MALLIT\XML\VHT-VESIHUOLTO.xml luettu. Luotu 122 kohdetta.
16.30.20  Käytetään muunnintiedostoa: C:\Program Files\Trimble\Novapoint\2024-ac21\Configuration\Finland\FeatureConverterFiles\LandXML.QuadriModel.PipeNetwork.cf.xml
16.30.21  Muunnettiin 122 kohdetta. Lähtötietojen lukumäärä oli 122.
16.30.26  1 tiedosto tuotu.
16.30.34  Added number of items: 122
```

```
25 497 634,801 m   6 673 928,827 m   0 m
```

Figure 42. LandXML test models in Trimble Novapoint 2025

According to the software provider, this limitation was related to ongoing platform updates, and LandXML import is expected to be improved in upcoming releases. An earlier version, Novapoint 2022, was able to display LandXML geometries correctly, as figure 43 illustrates.

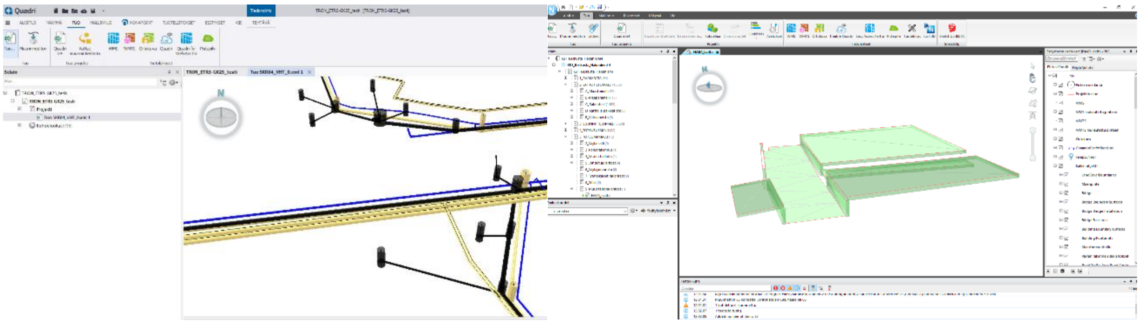
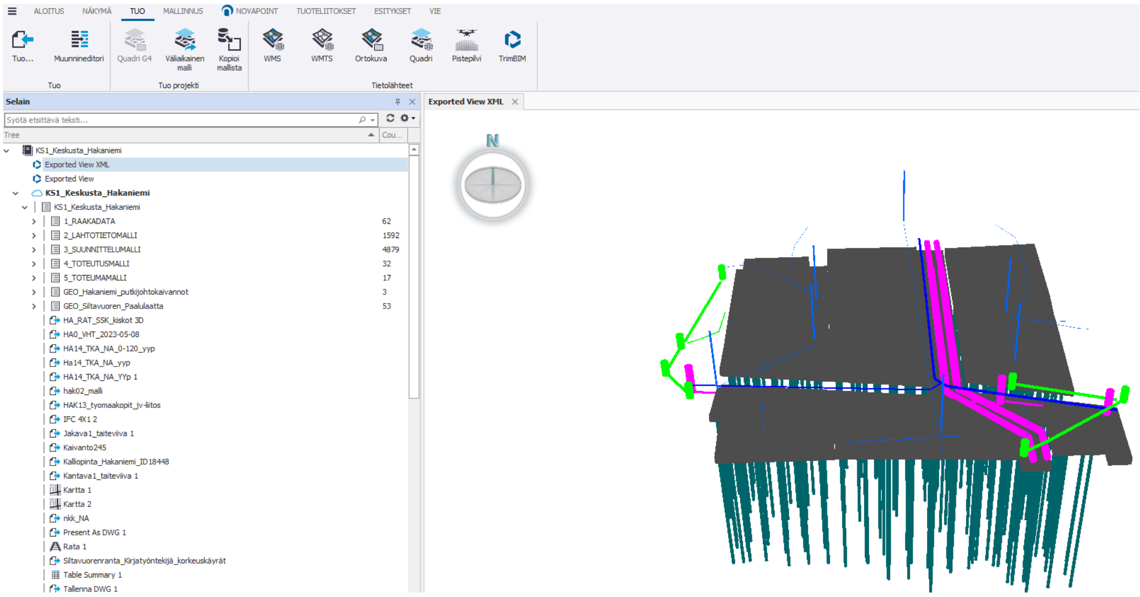


Figure 43. LandXML pipeline and pile cap models in Trimble Novapoint 2022

Metadata handling showed mixed results. Despite missing 3D volumes, LandXML import included general entity attributes, such as name, description and material, while detailed metadata sets were missing. IFC import, in turn, displayed only generic entity properties, such as name and description, leaving material information out of scope.

In cases of persistent LandXML import errors, Novapoint offers an alternative workflow, which relies on the TrimBIM format. TrimBIM was developed by Trimble for efficient data exchange within its product family [78]. It reduces file size and efficiently transfers 3D geometry but does not preserve metadata. An example of LandXML geometry converted into TrimBIM using Trimble Connect and imported into Novapoint, is shown in figure 44. Compared to the unsuccessful LandXML import demonstrated in figure 42, all modelled elements are on display, including piles, poles, correct volumes and colours.

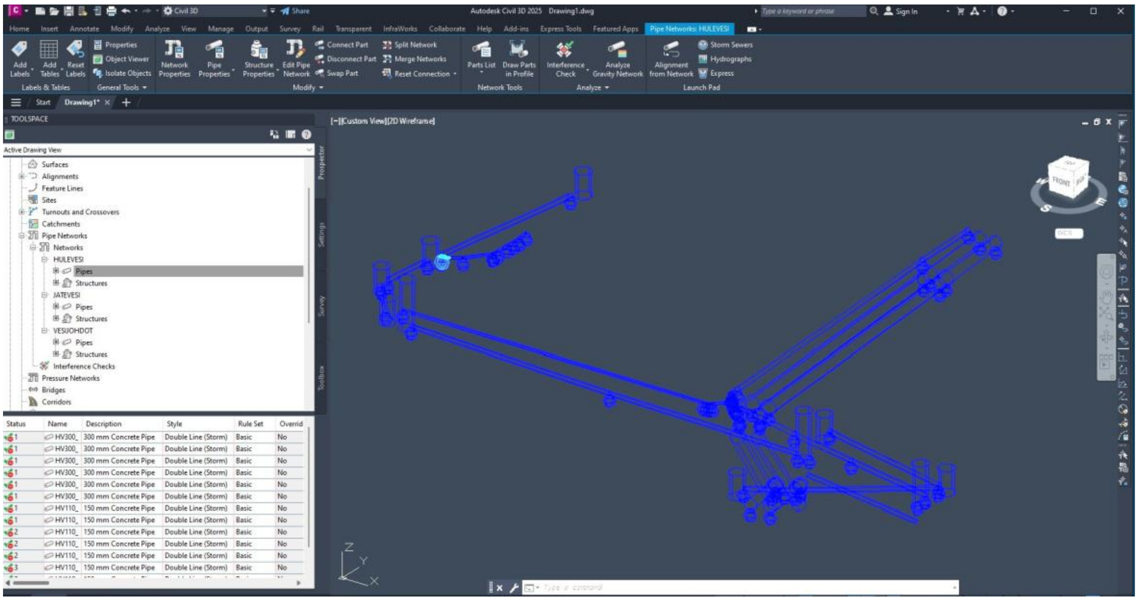


Tree	Cou…
KS1_Keskusta_Hakaniemi	
Exported View XML	
Exported View	
KS1_Keskusta_Hakaniemi	
KS1_Keskusta_Hakaniemi	
1_RAAKADATA	62
2_LAHTOTIETOMALLI	1592
3_SUUNNITTELUMALLI	4879
4_TOTEUTUSMALLI	32
5_TOTEUMAMALLI	17
GEO_Hakaniemi_putkijohtokaivannot	3
GEO_Siltavuoren_Paalulaatta	53
HA_RAT_SSK_kiskot 3D	
HA0_VHT_2023-05-08	
HA14_TKA_NA_0-120_yyp	
Ha14_TKA_NA_yyp	
HA14_TKA_NA_YYp 1	
hak02_malli	
HAK13_tyomaakopit_jv-liitos	
IFC 4X1 2	
Jakava1_taiteviiva 1	
Kaivanto245	
Kalliopinta_Hakaniemi_ID18448	
Kantava1_taiteviiva 1	
Kartta 1	
Kartta 2	
nkk_NA	
Present As DWG 1	
Rata 1	
Siltavuorenranta_Kirjatyöntekijä_korkeuskäyrät	
Table Summary 1	
Tallenna DWG 1	

Figure 44. TrimBIM test models in Trimble Novapoint

4.1.2.1.2 Civil3D

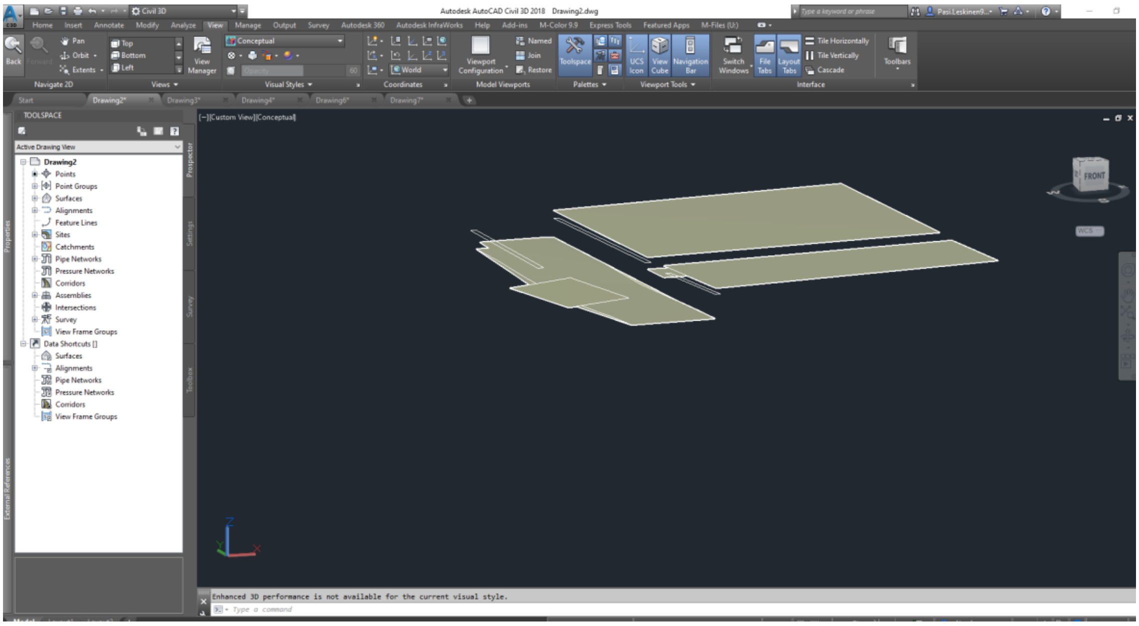
Civil3D is a design platform from the Autodesk product suite, widely applied in infrastructure design. Like Novapoint, it was used in both Jokeri and Crown Bridges light rail projects. Platform evaluation was limited to LandXML, as Civil3D does not support IFC import. LandXML import is handled through a dedicated inbuilt plugin.



Status	Name	Description	Style	Rule Set	Overrid
1	HV300_	300 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV300_	300 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV300_	300 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV300_	300 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV300_	300 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV300_	300 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV110_	150 mm Concrete Pipe	Double Line (Storm)	Basic	No
1	HV110_	150 mm Concrete Pipe	Double Line (Storm)	Basic	No
2	HV110_	150 mm Concrete Pipe	Double Line (Storm)	Basic	No
2	HV110_	150 mm Concrete Pipe	Double Line (Storm)	Basic	No
2	HV110_	150 mm Concrete Pipe	Double Line (Storm)	Basic	No
3	HV110_	150 mm Concrete Pipe	Double Line (Storm)	Basic	No

Figure 45. LandXML PipeNetwork models in Autodesk Civil3D

Importing the PipeNetwork model produced a wireframe representation model of pipe and manhole elements, as shown in figure 45. Volumetric properties were completely ignored, but pipe connectors were emphasised in an unexpected way. An attempt to import pile cap hull models resulted in correct representations of horizontal and tilted faces, but vertical faces were missing, as illustrated in figure 46. Example of properly imported vertical faces, for comparison, was demonstrated in figure 43. Metadata import displayed names, descriptions, diameter profiles and materials, while the pile cap metadata was limited to names and descriptions.



```
Enhanced 3D performance is not available for the current visual style.
```

Figure 46. LandXML Pile Cap TIN mesh model in Autodesk Civil3D

4.1.2.1.3 Tekla Structures

Tekla Structures, also owned by Trimble, is another infrastructure design platform included in this study due to its use in Jokeri and Crown Bridges projects. The platform supports both LandXML and IFC.

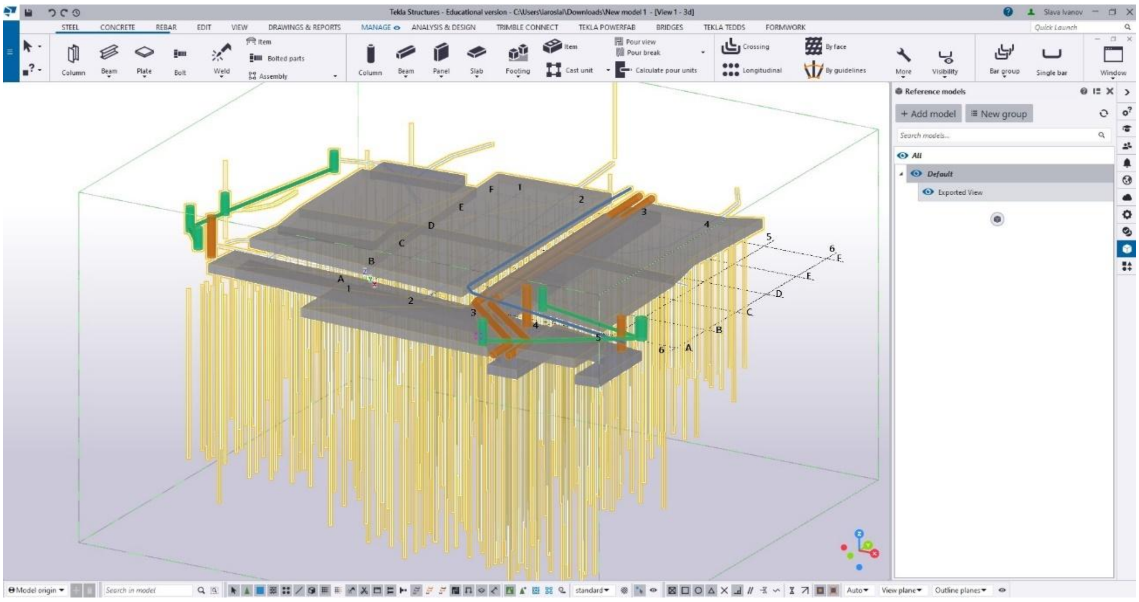
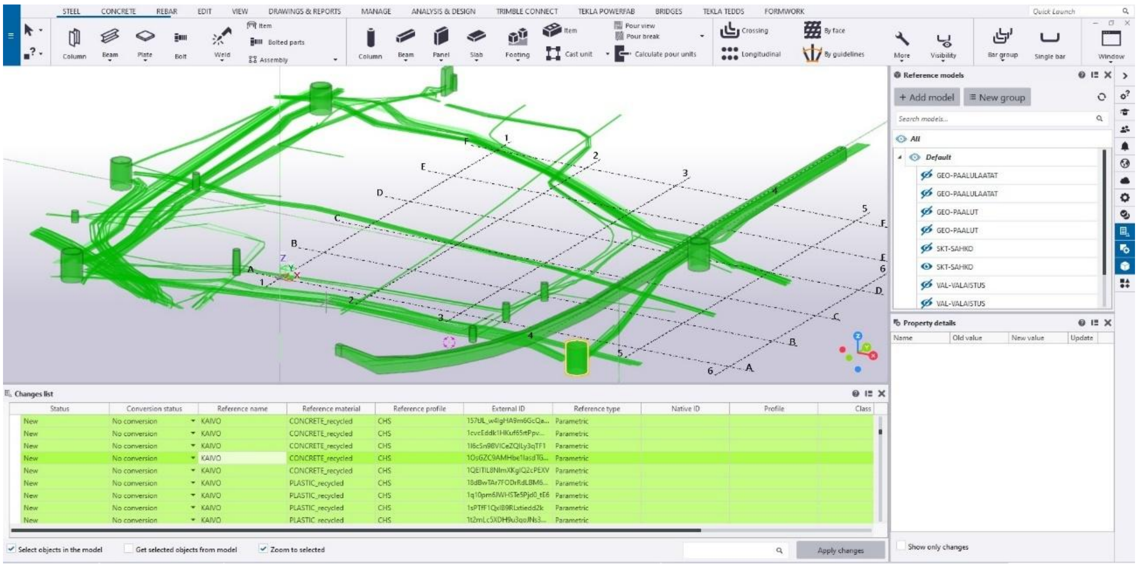


Figure 47. LandXML models in Tekla Structures

Tekla performed well while importing geometries from both formats. As shown in figure 47, LandXML pipe network types were recognised correctly and colour-coded appropriately, although metadata import was completely unsuccessful. IFC import preserved all geometric details and surface styles, with generic metadata properties of entities, such as names, descriptions and materials, as shown in figure 48. However, more detailed property sets were not imported.

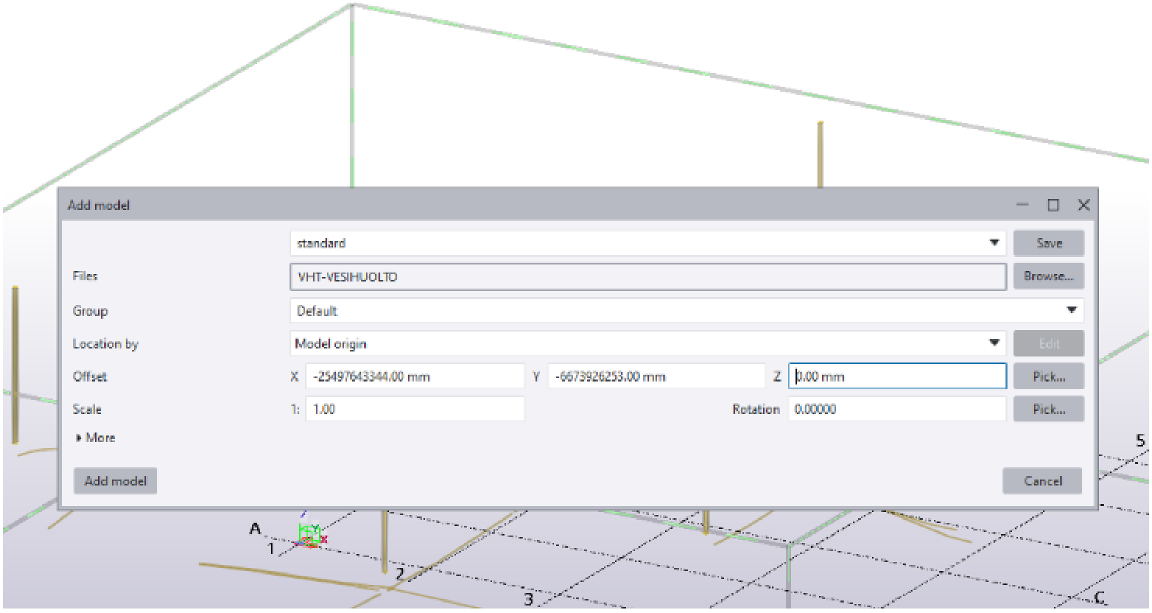


Status	Conversion status	Reference name	Reference material	Reference profile	External ID	Reference type	Native ID	Profile	Class
New	No conversion	KAIVO	CONCRETE_recycled	CHS	157UL_w4lgHA9m6GcQa...	Parametric			
New	No conversion	KAIVO	CONCRETE_recycled	CHS	1cvcEddk1HKuf65rtPpv...	Parametric			
New	No conversion	KAIVO	CONCRETE_recycled	CHS	1l6c5n98VICeZQILy3qTF1	Parametric			
New	No conversion	KAIVO	CONCRETE_recycled	CHS	1OsGZC9AMHbe1lasdTG...	Parametric			
New	No conversion	KAIVO	CONCRETE_recycled	CHS	1QEITIL8NImXKglQ2cPEXV	Parametric			
New	No conversion	KAIVO	PLASTIC_recycled	CHS	18dBwTAr7FODrRdLBM6...	Parametric			
New	No conversion	KAIVO	PLASTIC_recycled	CHS	1q10pm6JWHSTe5Pjd0_tE6	Parametric			
New	No conversion	KAIVO	PLASTIC_recycled	CHS	1sPTfF1QxIB9RLxtiedd2k	Parametric			
New	No conversion	KAIVO	PLASTIC_recycled	CHS	1t2mLc5XDH9u3qoJNs3...	Parametric			

☑ Select objects in the model ☐ Get selected objects from model ☑ Zoom to selected

Figure 48. IFC model property view in Tekla Structures

One notable limitation observed in Tekla was inability to automatically align the workspace with the model's coordinates. Manual origin translation had to be applied to each model individually, as illustrated in figure 49. Since all models share the same coordinate system, to ensure proper alignment, they must also share the same point of origin for proper alignment.



Add model

standard

Files: VHT-VESIHUOLTO

Group: Default

Location by: Model origin

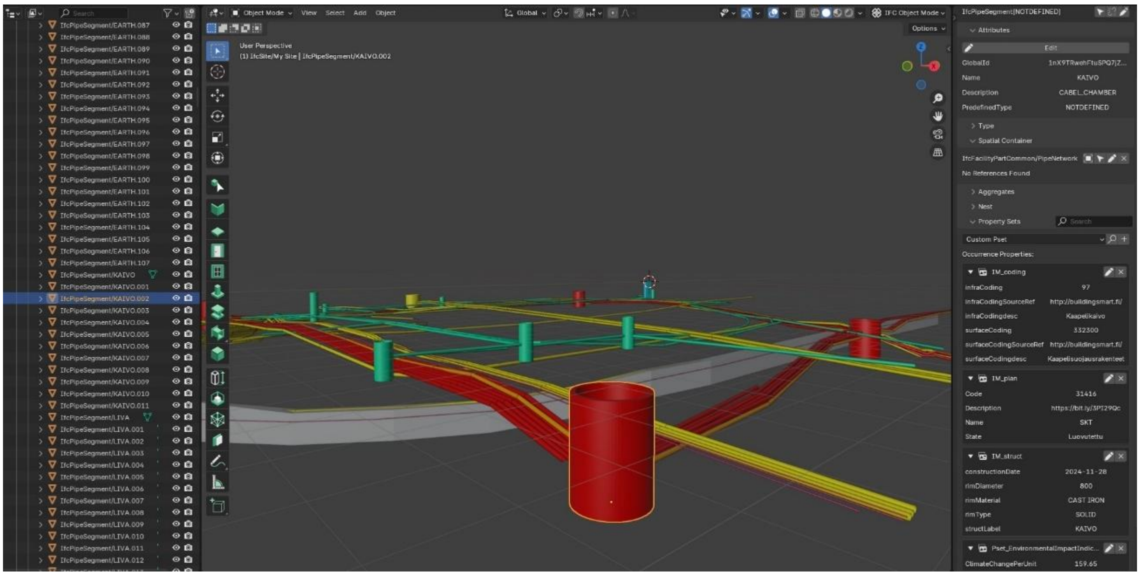
Offset: X -25497643344.00 mm Y -6673926253.00 mm Z 0.00 mm

Scale: 1: 1.00 Rotation 0.00000

Figure 49. Coordinate translation in Tekla Structures

4.1.2.1.4 Blender Bonsai

Blender is a multifunctional free software platform that supports BIM modelling through the Bonsai add-on. It was selected due to its extensive toolsets specially tailored for BIM modelling, open-source policy and excellent accessibility for both beginners and professionals. Bonsai does not support LandXML, therefore the testing was performed using IFC format only.



GlobalId	1nX9TRwehFtu5PQ7jZ...
Name	KAIVO
Description	CABEL_CHAMBER
PredefinedType	NOTDEFINED

IfcFacilityPartCommon/PipeNetwork

No References Found

Occurrence Properties:

IM_coding

infraCoding	97
infraCodingSourceRef	http://buildingsmart.fi/
infraCodingdesc	Kaapelikaivo
surfaceCoding	332300
surfaceCodingSourceRef	http://buildingsmart.fi/
surfaceCodingdesc	Kaapelisuojausrakenteet

IM_plan

Code	31416
Description	https://bit.ly/3PI29Qc
Name	SKT
State	Luovutettu

IM_struct

constructionDate	2024-11-28
rimDiameter	800
rimMaterial	CAST IRON
rimType	SOLID
structLabel	KAIVO

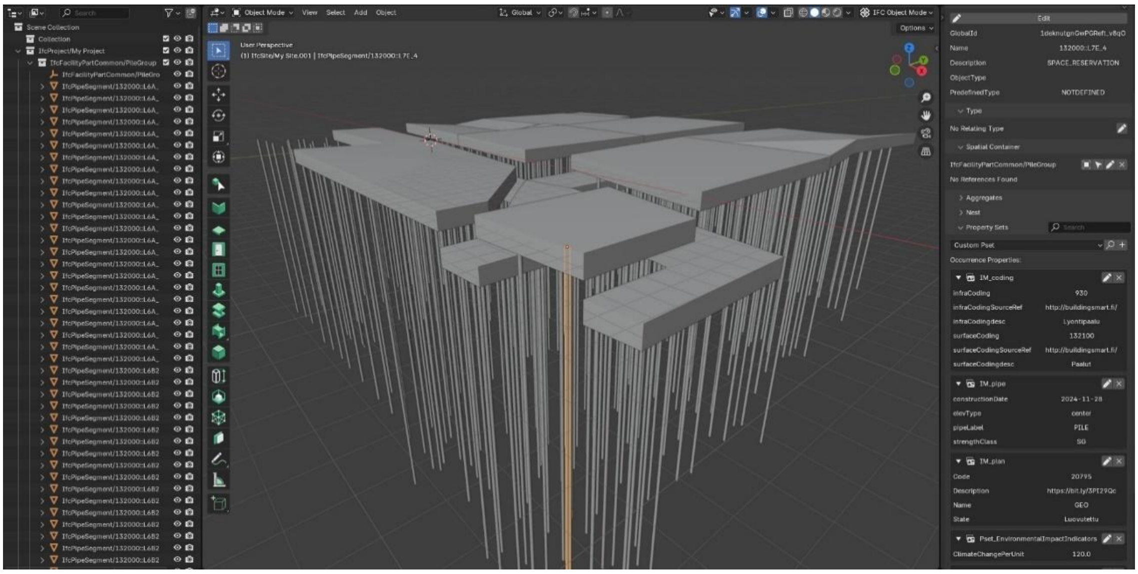
Pset_EnvironmentalImpactIndic...

ClimateChangePerUnit	159.65

Figure 50. Utility line models in Blender Bonsai

Importing models into Blender Bonsai was straightforward for both utility line and deep foundation datasets, which are shown in figures 50 and 51 respectively. All details, including colour profiles and origin points of each object, were rendered accurately.

Blender Bonsai is recognised for its comprehensive IFC schema support, featuring all the latest updates. This was confirmed during testing: the platform provided access to all generic properties of each entity, displayed both standard and custom-built metadata sets, and correctly interpreted inner relational hierarchies of the models.



GlobalId	1deknutgnGwPGRefI_v8qO
Name	132000::L7E_4
Description	SPACE_RESERVATION
ObjectType	
PredefinedType	NOTDEFINED

No Relating Type

IfcFacilityPartCommon/PileGroup

No References Found

IM_coding

infraCoding	930
infraCodingSourceRef	http://buildingsmart.fi/
infraCodingdesc	Lyontipaalu
surfaceCoding	132100
surfaceCodingSourceRef	http://buildingsmart.fi/
surfaceCodingdesc	Paalut

IM_pipe

constructionDate	2024-11-28
elevType	center
pipeLabel	PILE
strengthClass	S0

IM_plan

Code	20795
Description	https://bit.ly/3PI29Qc
Name	GEO
State	Luovutettu

Pset_EnvironmentalImpactIndicators

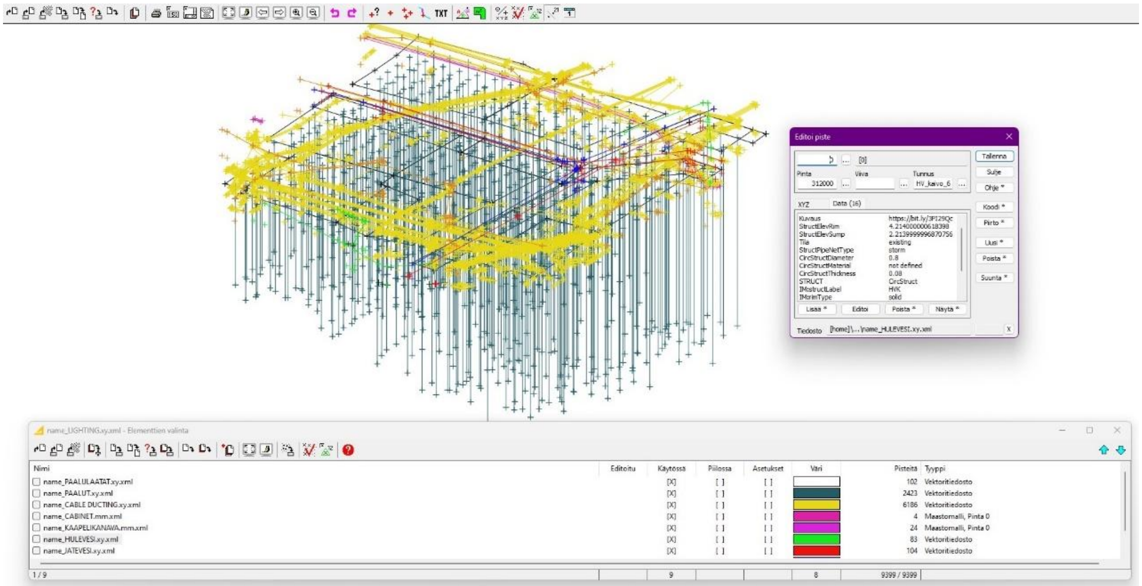
ClimateChangePerUnit	120.0

Figure 51. IFC foundation models in Blender Bonsai

4.1.2.1.5 3DWin

3DWin is an information modelling tool from the Novatron product catalogue. This software is widely used in the Finnish BIM industry and in both Jokeri and Crown Bridges projects. Known for its broad format compatibility and popularity among land surveyors, it supports import of both data exchange formats featured in this study and provides integrated access to InfraBIM and InfraRAK code libraries.

Visualisation engine of 3Dwin is distinctively wireframe-based and does support vertical mesh faces. Models containing vertical triangulations or overlapping triangles often result in partial rendering or failed import. For example, in LandXML pile cap models, triangle meshes are completely missing, as illustrated in figure 52. On the other hand, the metadata interpreter offered extensive access to both standardised and user-defined properties.



[0]

Pinta 312000

Viiva

Tunnus HV_kaivo_6

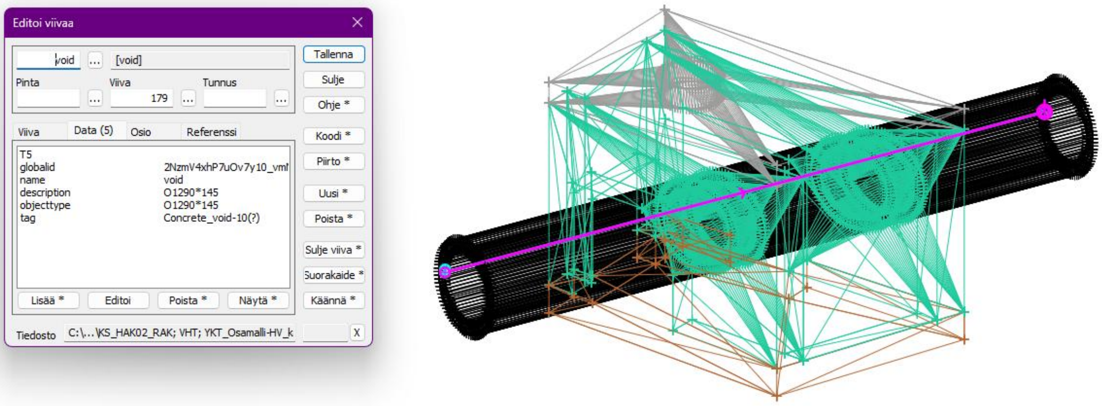
Kuvaus	https://bit.ly/3PI29Qc
StructElevRim	4.21400000618398
StructElevSump	2.2139999996870756
Tila	existing
StructPipeNetType	storm
CircStructDiameter	0.8
CircStructMaterial	not defined
CircStructThickness	0.08
STRUCT	CircStruct
IM:structLabel	HVK
IM:rimType	solid

Tiedosto [home]\...\name_HULEVESI.xy.xml

Nimi	Editoitu	Käytössä	Piilossa	Asetukset	Väri	Pisteitä	Tyyppi
☐ name_PAALULAATAT.xy.xml		[X]	[ ]	[ ]	⬜	102	Vektoritiedosto
☐ name_PAALUT.xy.xml		[X]	[ ]	[ ]	⬛	2423	Vektoritiedosto
☐ name_CABLE DUCTING.xy.xml		[X]	[ ]	[ ]	🟨	6186	Vektoritiedosto
☐ name_CABINET.mm.xml		[X]	[ ]	[ ]	🟪	4	Maastomalli, Pinta 0
☐ name_KAAPELIKANAVA.mm.xml		[X]	[ ]	[ ]	🟪	24	Maastomalli, Pinta 0
☐ name_HULEVESI.xy.xml		[X]	[ ]	[ ]	🟩	83	Vektoritiedosto
☐ name_JATEVESI.xy.xml		[X]	[ ]	[ ]	🟥	104	Vektoritiedosto
1 / 9		9			8	9399 / 9399	

Figure 52. LandXML models in 3DWin

While Inframodel 4.2 import was largely functional, 3DWin encountered unexpected errors while reading IFC4.3 models, likely due to incomplete support for the latest schema version. For demonstration purposes, an older IFC2.3 example model was imported instead. Unlike LandXML, IFC visualisation displayed all triangles of the generated mesh, regardless of spatial orientation, as illustrated in figure 53, and generic entity metadata was accessible through the property window.



void [void]

Pinta

Viiva 179

Tunnus

T5	
globalid	2NzmV4xhP7uOv7y10_vml
name	void
description	O1290*145
objecttype	O1290*145
tag	Concrete_void-10(?)

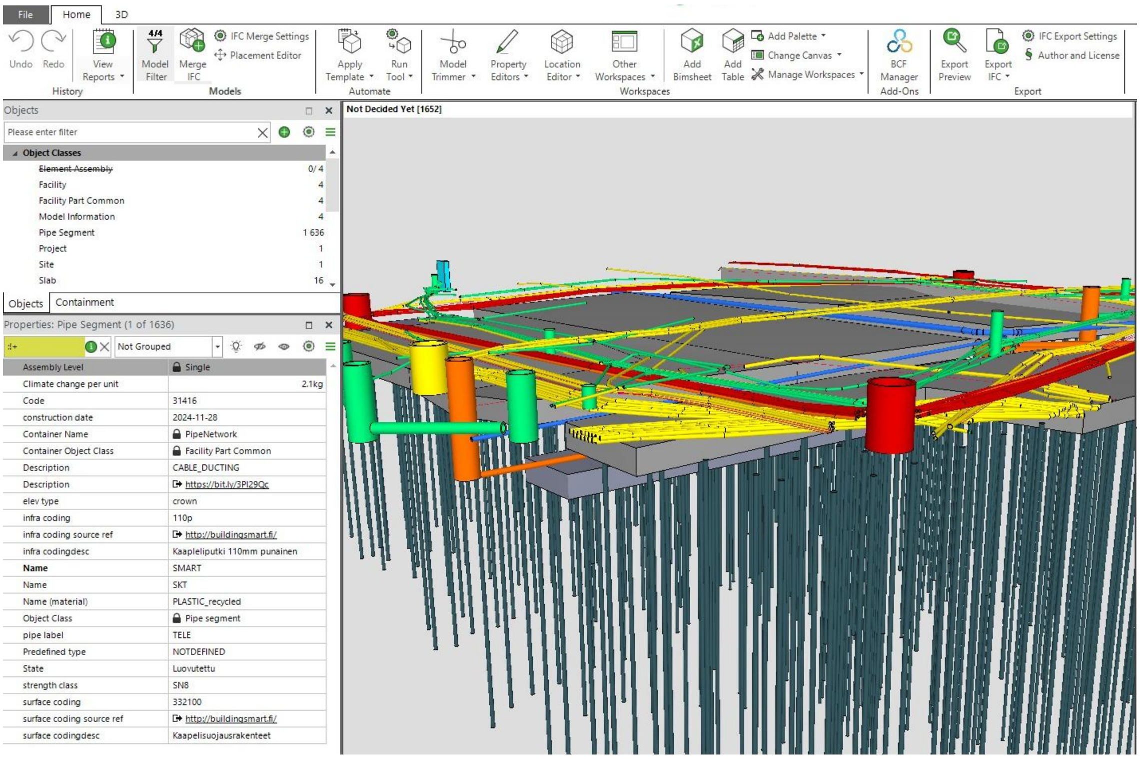
Tiedosto C:\...\KS_HAK02_RAK; VHT; YKT_Osamalli-HV_k

Figure 53. IFC2X3 visualisation in 3DWin

4.1.2.1.6 SimpleBIM

SimpleBIM is a multifunctional BIM platform primarily designed for simple model manipulation, such as cleaning and restructuring, rather than full-scale modelling. Nevertheless, it supports basic editing and export and was implemented in the Crown Bridges project, which justified its inclusion in this study. The platform supports IFC format only.

As illustrated in figure 54, SimpleBIM demonstrated excellent visualisation capabilities, rendering all geometric details correctly. Metadata handling was equally strong, the platform displayed both schema-prescribed or arbitrary information.



Object Classes	
~~Element Assembly~~	0/ 4
Facility	4
Facility Part Common	4
Model Information	4
Pipe Segment	1 636
Project	1
Site	1
Slab	16

Properties: Pipe Segment (1 of 1636)

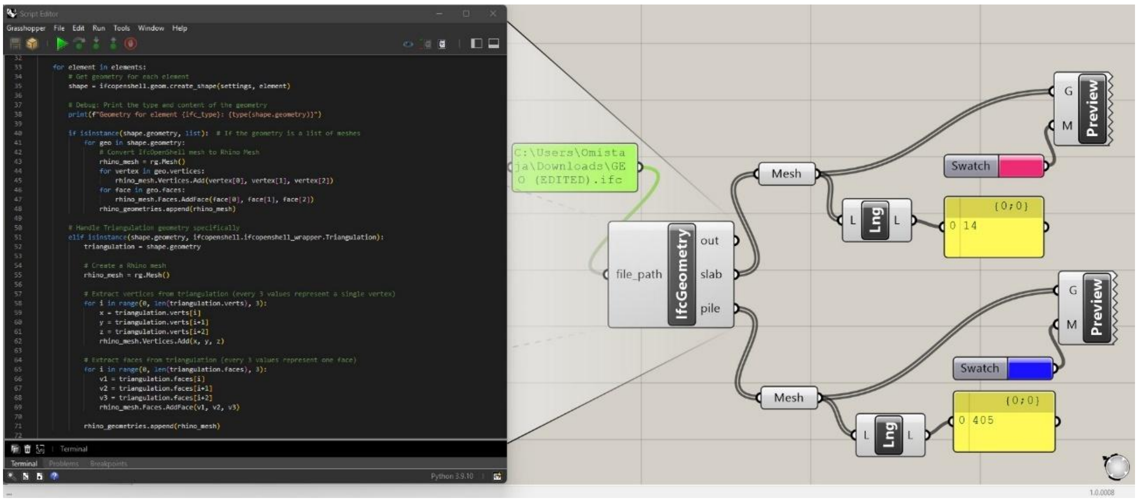
Assembly Level	Single
Climate change per unit	2.1kg
Code	31416
construction date	2024-11-28
Container Name	PipeNetwork
Container Object Class	Facility Part Common
Description	CABLE_DUCTING
Description	https://bit.ly/3PI29Qc
elev type	crown
infra coding	110p
infra coding source ref	http://buildingsmart.fi/
infra codingdesc	Kaapleliputki 110mm punainen
Name	SMART
Name	SKT
Name (material)	PLASTIC_recycled
Object Class	Pipe segment
pipe label	TELE
Predefined type	NOTDEFINED
State	Luovutettu
strength class	SN8
surface coding	332100
surface coding source ref	http://buildingsmart.fi/
surface codingdesc	Kaapelisuojausrakenteet

Figure 54. IFC models in SimpleBIM

4.1.2.1.7 Rhino

Rhino is a versatile 3D modelling solution applied widely across multiple industries. For this study, it was tested in combination with Grasshopper visual programming interface, due to their reputation of a powerful parametric design tool and popularity among BIM professionals.

Rhino modelling approach differs from other applications included in this study. Its functionalities are often managed through visual programming and Python scripting, as shown in figure 55. While this technically makes LandXML import possible, it requires programming effort and was therefore excluded from the present scope. Instead, IFC models were used to demonstrate the platform capabilities.



```
        for element in elements:
            # Get geometry for each element
            shape = ifcopenshell.geom.create_shape(settings, element)

            # Debug: Print the type and content of the geometry
            print(f"Geometry for element {ifc_type}: {type(shape.geometry)}")

            if isinstance(shape.geometry, list):  # If the geometry is a list of meshes
                for geo in shape.geometry:
                    # Convert IfcOpenShell mesh to Rhino Mesh
                    rhino_mesh = rg.Mesh()
                    for vertex in geo.vertices:
                        rhino_mesh.Vertices.Add(vertex[0], vertex[1], vertex[2])
                    for face in geo.faces:
                        rhino_mesh.Faces.AddFace(face[0], face[1], face[2])
                    rhino_geometries.append(rhino_mesh)

            # Handle Triangulation geometry specifically
            elif isinstance(shape.geometry, ifcopenshell.ifcopenshell_wrapper.Triangulation):
                triangulation = shape.geometry

                # Create a Rhino mesh
                rhino_mesh = rg.Mesh()

                # Extract vertices from triangulation (every 3 values represent a single vertex)
                for i in range(0, len(triangulation.verts), 3):
                    x = triangulation.verts[i]
                    y = triangulation.verts[i+1]
                    z = triangulation.verts[i+2]
                    rhino_mesh.Vertices.Add(x, y, z)

                # Extract faces from triangulation (every 3 values represent one face)
                for i in range(0, len(triangulation.faces), 3):
                    v1 = triangulation.faces[i]
                    v2 = triangulation.faces[i+1]
                    v3 = triangulation.faces[i+2]
                    rhino_mesh.Faces.AddFace(v1, v2, v3)

                rhino_geometries.append(rhino_mesh)
```

Figure 55. Grasshopper visual programming interface for Rhino, and Python script for IFC import

With IfcOpenShell library, models were imported smoothly, preserving all geometric details, as figure 56 illustrates. Additionally, the library enables extraction and structuring of all metadata types, inner hierarchies and other IFC-native properties.

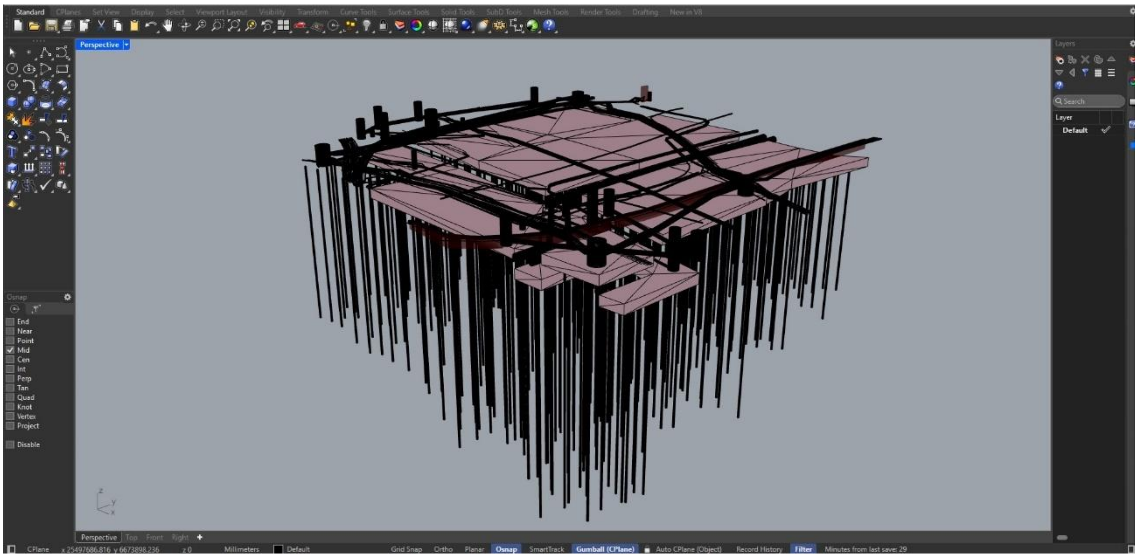


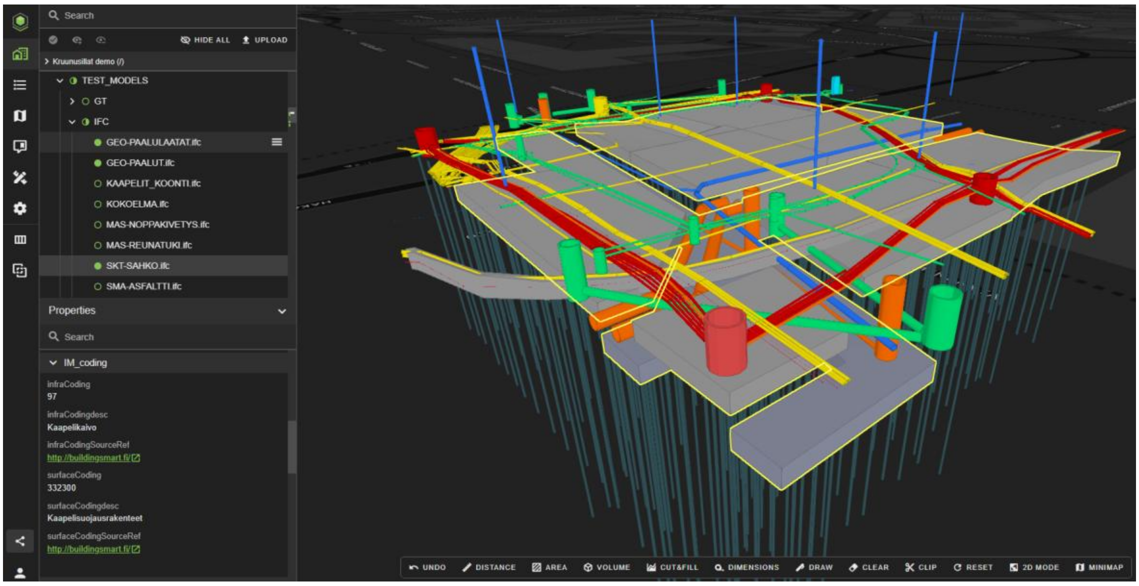
Figure 56. IFC models in Rhino

4.1.2.2 Model Viewers

4.1.2.2.1 Vektor.io

Vektor.io is a platform for BIM model viewing, analysing and collaboration. It was chosen for testing due to its highly efficient visualisation engine and successful implementation track at both Jokeri and Crown Bridges light rail projects. The platform can import a broad range of formats including LandXML and IFC.

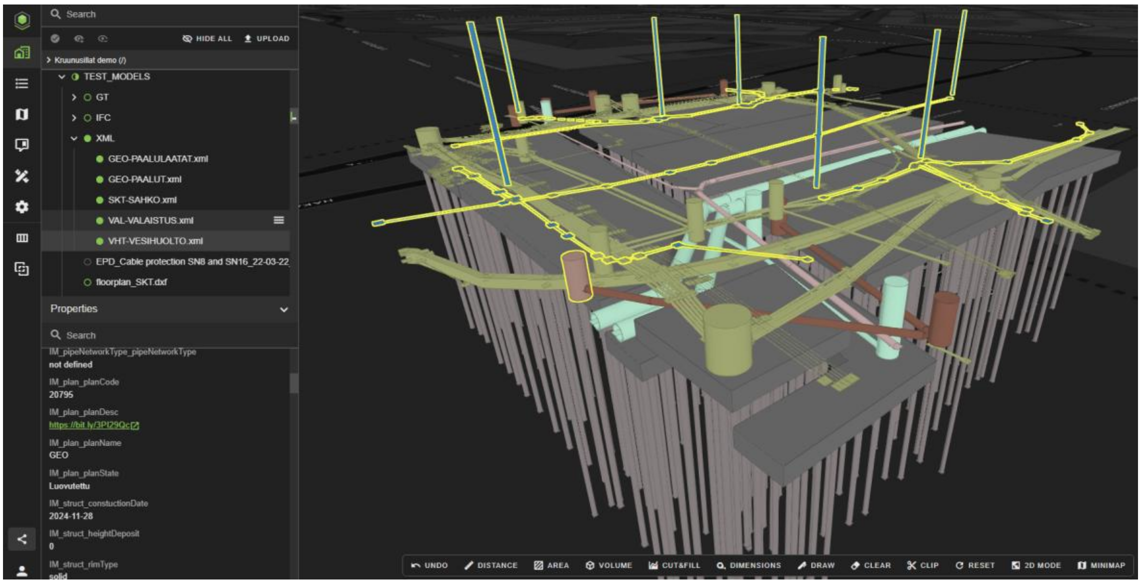
Both format imports were successful, as expected, with all elements present and accurately aligned with the background map. One minor difference was that the LandXML struct elements were rendered as closed cylinders, omitting the wall thickness and inner diameter. Another observation regarding LandXML was the unexpected rendering of pipe connectors, which are abstract elements and should not appear in the scene. IFC visualisation left no questions.



infraCoding
97
infraCodingdesc
Kaapelikaivo
infraCodingSourceRef
http://buildingsmart.fi/
surfaceCoding
332300
surfaceCodingdesc
Kaapelisuojausrakenteet
surfaceCodingSourceRef
http://buildingsmart.fi/

Figure 57. IFC models in Vektor.io

Metadata inspection confirmed that all entity-specific metadata fields, across all assigned presets, were correctly interpreted and displayed. In contrast to IFC, LandXML property listing also included fields inherited from parent elements of the entity, thus demonstrating accurate interpretation of relational hierarchy within the model. IFC visualisation is illustrated in figure 57 and LandXML in figure 58.



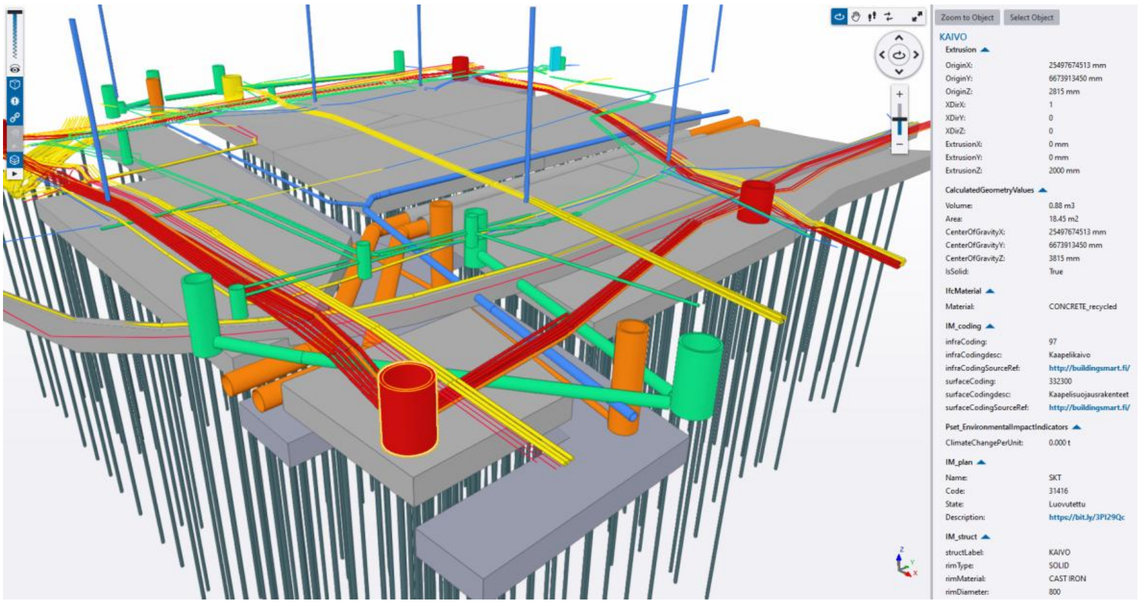
IM_pipeNetworkType_pipeNetworkType
not defined
IM_plan_planCode
20795
IM_plan_planDesc
https://bit.ly/3PI29Qc
IM_plan_planName
GEO
IM_plan_planState
Luovutettu
IM_struct_constuctionDate
2024-11-28
IM_struct_heightDeposit
0
IM_struct_rimType
solid

Figure 58. LandXML models in Vektor.io

4.1.2.2.2 Trimble Connect

Trimble Connect is a well-known platform for BIM model viewing, analysing and collaboration that belongs to the Trimble product family. It was chosen for testing due to its interaction with Trimble and Tekla design platforms and implementation in both Jokeri and Crown Bridges light rail projects. The platform can visualise both LandXML and IFC.

Both format imports resulted in correct visualisation of all geometric elements. One minor difference was that the LandXML pipe elements were rendered as closed cylinders, showing no wall thickness or inner diameter. IFC visualisation had no imperfections.



KAIVO

Extrusion
OriginX: 25497674513 mm
OriginY: 6673913450 mm
OriginZ: 2815 mm
XDirX: 1
XDirY: 0
XDirZ: 0
ExtrusionX: 0 mm
ExtrusionY: 0 mm
ExtrusionZ: 2000 mm

CalculatedGeometryValues
Volume: 0.88 m3
Area: 18.45 m2
CenterOfGravityX: 25497674513 mm
CenterOfGravityY: 6673913450 mm
CenterOfGravityZ: 3815 mm
IsSolid: True

IfcMaterial
Material: CONCRETE_recycled

IM_coding
infraCoding: 97
infraCodingdesc: Kaapelikaivo
infraCodingSourceRef: http://buildingsmart.fi/
surfaceCoding: 332300
surfaceCodingdesc: Kaapelisuojausrakenteet
surfaceCodingSourceRef: http://buildingsmart.fi/

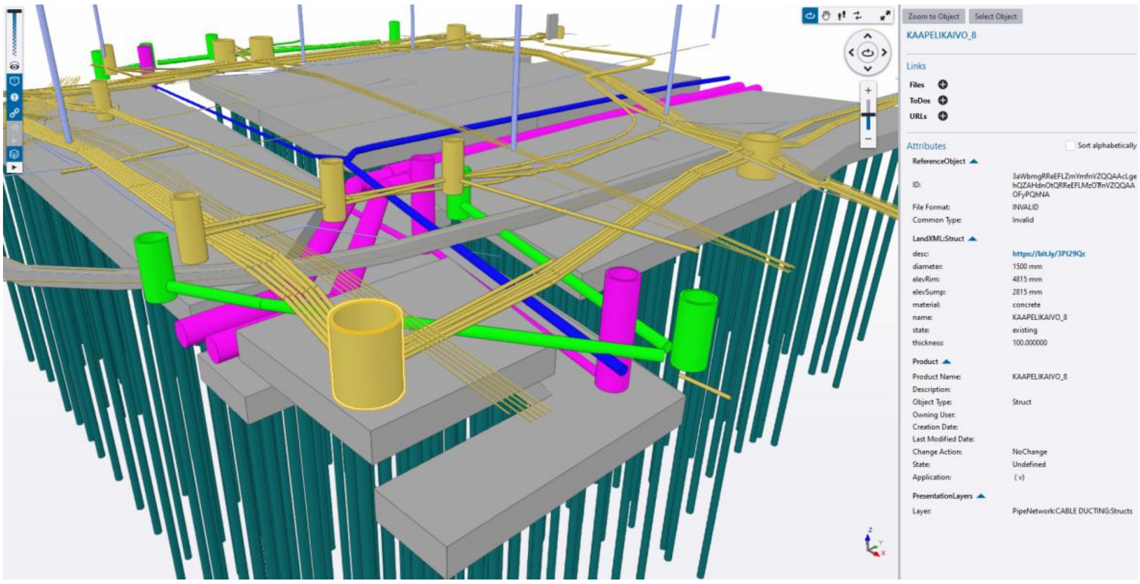
Pset_EnvironmentalImpactIndicators
ClimateChangePerUnit: 0.000 t

IM_plan
Name: SKT
Code: 31416
State: Luovutettu
Description: https://bit.ly/3PI29Qc

IM_struct
structLabel: KAIVO
rimType: SOLID
rimMaterial: CAST IRON
rimDiameter: 800

Figure 59. IFC models in Trimble Connect

Metadata inspection confirmed that all entity specific metadata fields in all presets, including those inherited from the parent element, were interpreted and displayed correctly for IFC. In contrast, LandXML property listing was limited to generic schema elements, leaving inframodel extensions out of scope. IFC visualisation is illustrated in figure 59 and LandXML in figure 60.



KAAPELIKAIVO_8

ID: 3aWbmgRReEFLZmYmfnVZQQAAcLgehQZAHdnOtQRReEFLMzOTfnVZQQAAOFyPQhNA
File Format: INVALID
Common Type: Invalid

desc: https://bit.ly/3PI29Qc
diameter: 1500 mm
elevRim: 4815 mm
elevSump: 2815 mm
material: concrete
name: KAAPELIKAIVO_8
state: existing
thickness: 100.000000

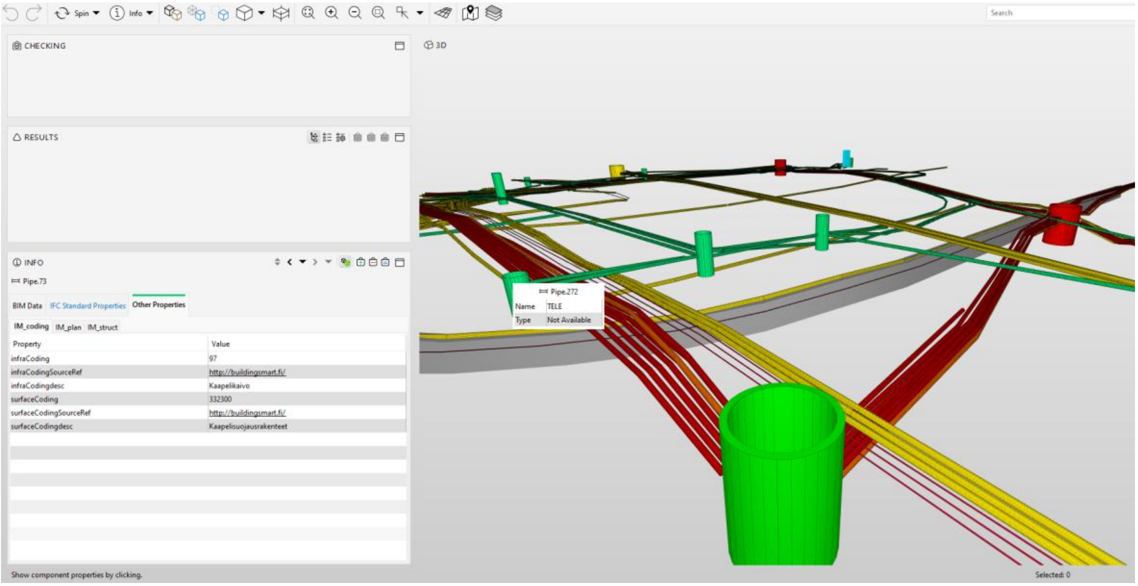
Product Name: KAAPELIKAIVO_8
Description:
Object Type: Struct
Owning User:
Creation Date:
Last Modified Date:
Change Action: NoChange
State: Undefined
Application: (v)

Layer: PipeNetwork:CABLE DUCTING:Structs

Figure 60. LandXML models in Trimble Connect

4.1.2.2.3 Solibri and Dalux

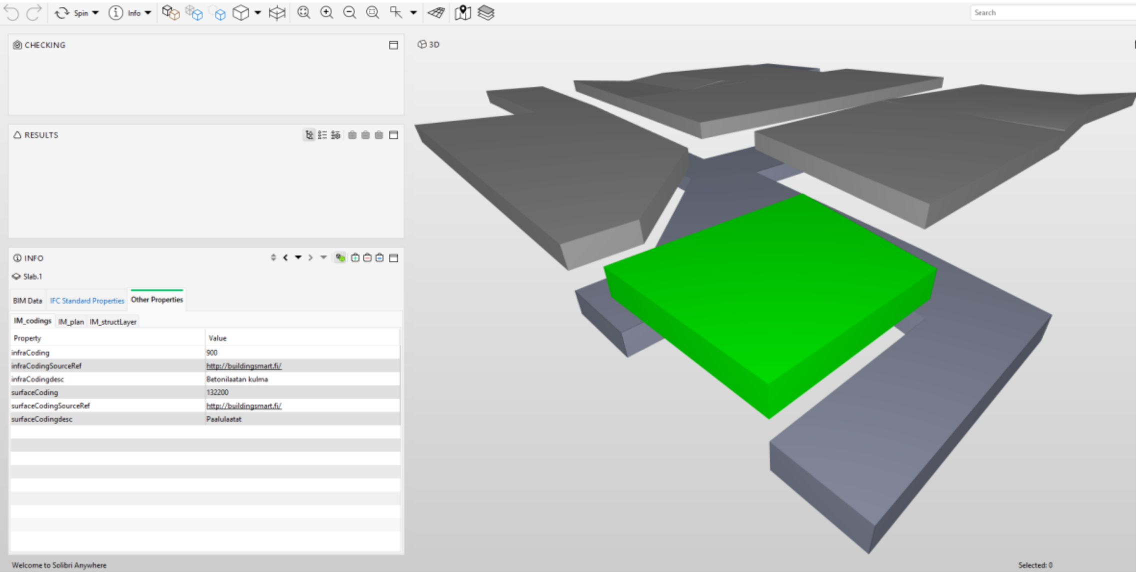
Solibri and Dalux are two separate platforms for BIM model viewing, sharing and analysing. Both applications were originally intended for the building assets data rather than infrastructure, although Dalux recently advanced to meet the infrastructure branch needs. The platforms were included into this study to establish a connection between infrastructure and building construction software types. Testing featured free versions of both platforms. The platforms do not support LandXML import, despite announced LandXML import for Dalux, therefore the testing was limited to IFC only.



Property	Value
infraCoding	97
infraCodingSourceRef	http://buildingsmart.fi/
infraCodingdesc	Kaapelikaivo
surfaceCoding	332300
surfaceCodingSourceRef	http://buildingsmart.fi/
surfaceCodingdesc	Kaapelisuojausrakenteet

Figure 61. Utility line models in Solibri

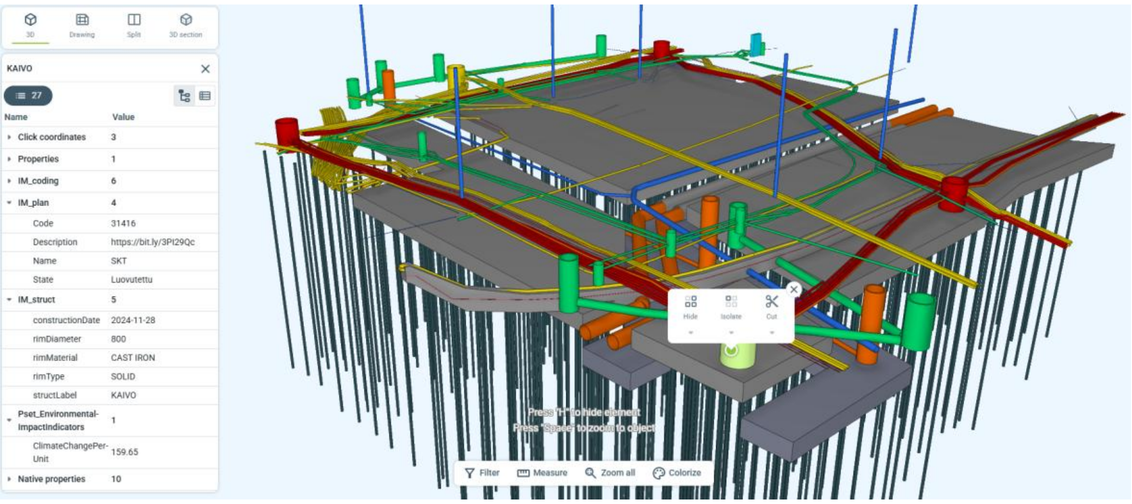
Rendering was excellent on both platforms, with no loss of any detail. One limitation of the free version of Solibri is that it allows viewing only one model at a time, while Dalux supports multi-model browsing.



Property	Value
infraCoding	900
infraCodingSourceRef	http://buildingsmart.fi/
infraCodingdesc	Betonilaatan kulma
surfaceCoding	132200
surfaceCodingSourceRef	http://buildingsmart.fi/
surfaceCodingdesc	Paalulaatat

Figure 62. Pile cap models in Solibri

All embedded metadata was accessible and accurately categorised in both applications. Utility line and pile cap model visualisations in Solibri are illustrated in figures 61 and 62 respectively, and test models rendered in Dalux are shown in figure 63.



KAIVO

27

Name	Value
Click coordinates	3
Properties	1
IM_coding	6
IM_plan	4
Code	31416
Description	https://bit.ly/3PI29Qc
Name	SKT
State	Luovutettu
IM_struct	5
constructionDate	2024-11-28
rimDiameter	800
rimMaterial	CAST IRON
rimType	SOLID
structLabel	KAIVO
Pset_Environmental-ImpactIndicators	1
ClimateChangePer-Unit	159.65
Native properties	10

Press "H" to hide element
Press "Space" to zoom to object

Figure 63. IFC test models in Dalux

4.1.2.2.4 SOVA3D

The most distinctive example of viewing application in this research is Sova3D. This is a platform for model viewing, primarily developed for municipal-level use. It was chosen for testing due to its recent adoption by the City of Helsinki and integration with Lupapiste-construction permit issuance service. In terms of supported format range, Sova3D is intended to visualise both LandXML and IFC format. Although considering, that the solution is still relatively new and is under ongoing development, some features may not be yet available.

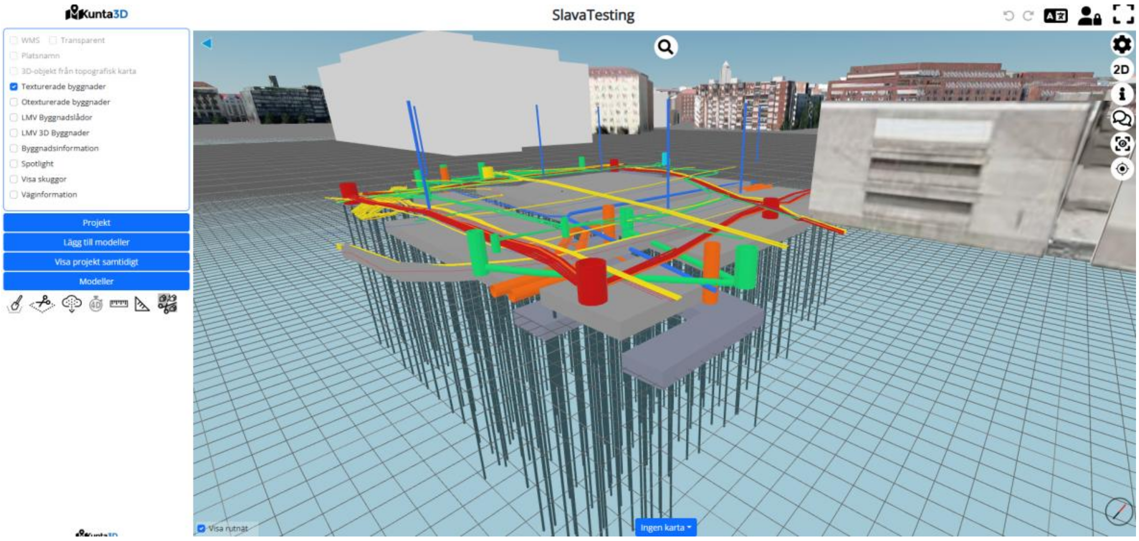


Figure 64. IFC models rendered in Sova3D

Both format imports were visualised successfully, including all geometric elements. However, the platform does not currently provide access to metadata properties. A brief inquiry confirmed that metadata management is under development and will be available in the upcoming releases. IFC visualisation is illustrated in figure 64 and LandXML in figure 65.

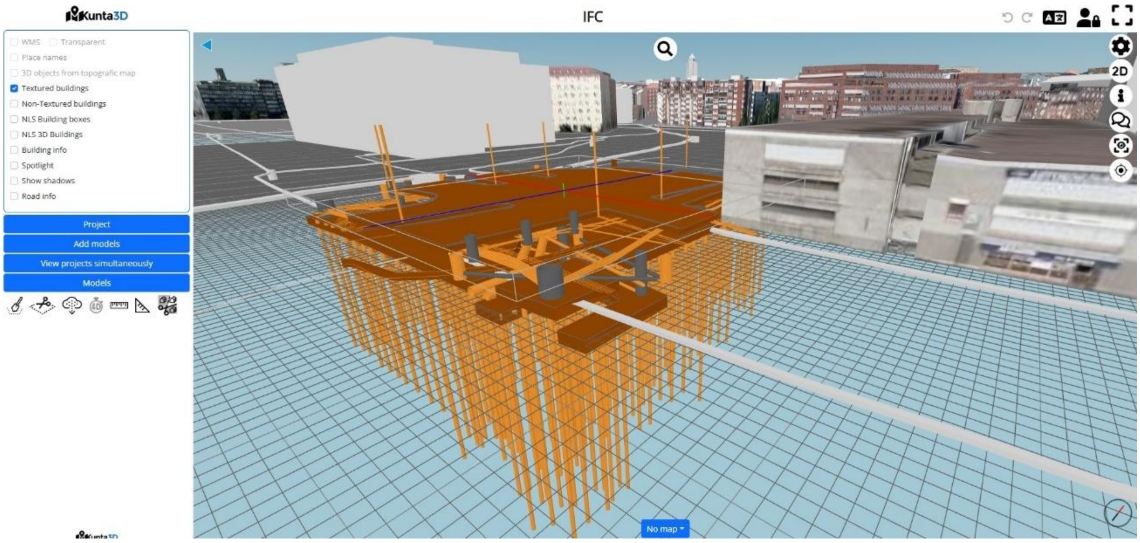


Figure 65. LandXML models rendered in Sova3D

4.1.2.3 Land Surveying and Site Management

4.1.2.3.1 Novatron Xsite

The Xsite machine control system is a toolkit that enables model-based operation for construction vehicles. It was included in this study due to its successful use history in Finnish civil construction projects, including Jokeri and Crown Bridges. Xsite system architecture handles BIM Models through a tablet installed in the vehicle cabin, as shown in figure 66. The tablet application, however, does not support IFC, therefore the testing was conducted using LandXML models. [79.]

Figure 66. Xsite machine control tablet in use [79].

Model readability evaluation was performed using Landnova desktop simulator. Since the system's primary role is to support onsite works, such as excavation or grading, metadata handling is not provided, and the model testing was focused on 3D geometry rendering and the platform's ability to use models for excavation referencing.

LandXML import was successful for both extruded and hull geometries, including relevant color-coding. An additional observation was that pipe connectors were rendered as flat plates. As demonstrated in figure 67, the system recognised the objects boundaries and elevation levels, enabling their use as operation guides. The bottom segment of the tablet interface displayed a cross section view together with calculated distance between the reference structure and the bucket edge.

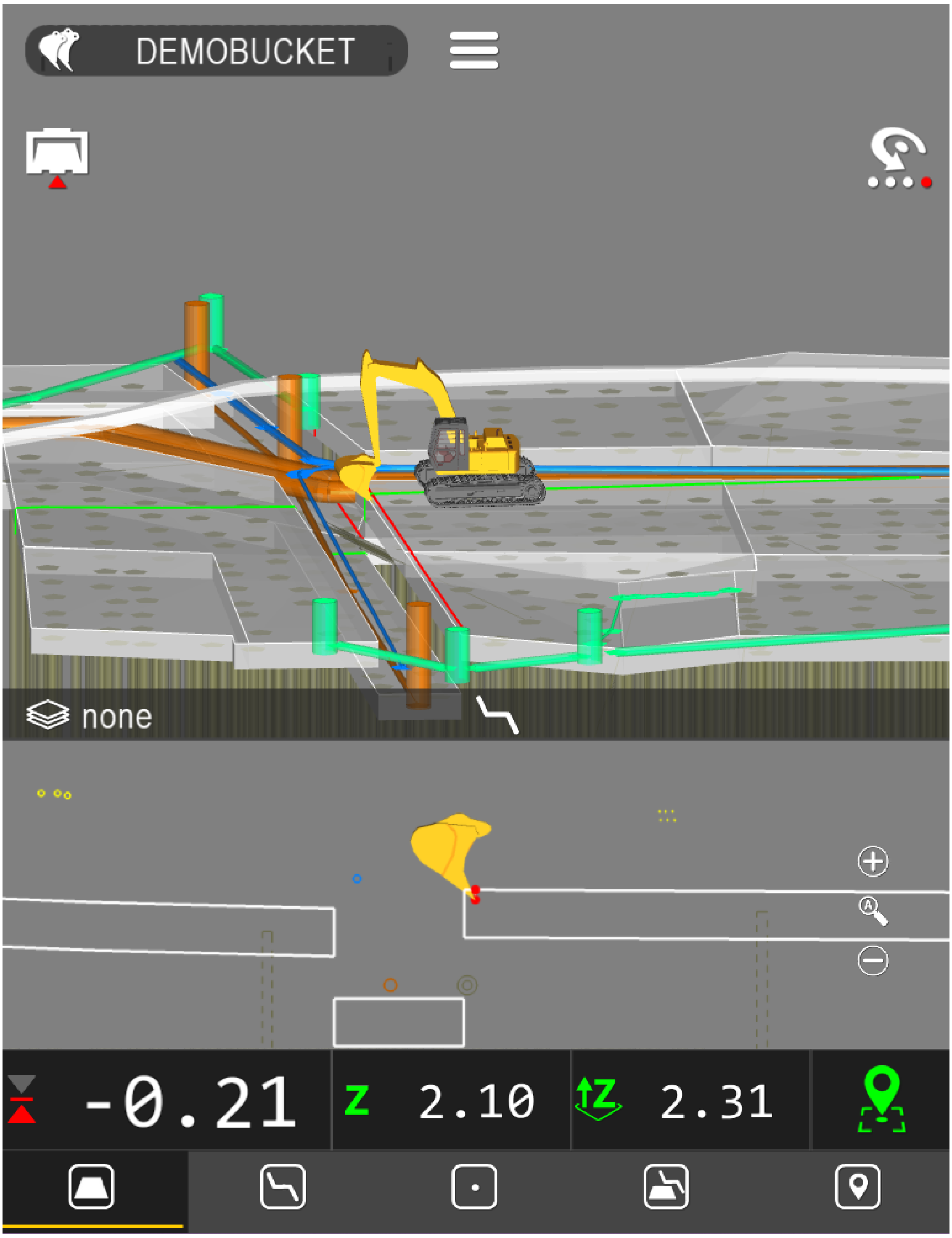
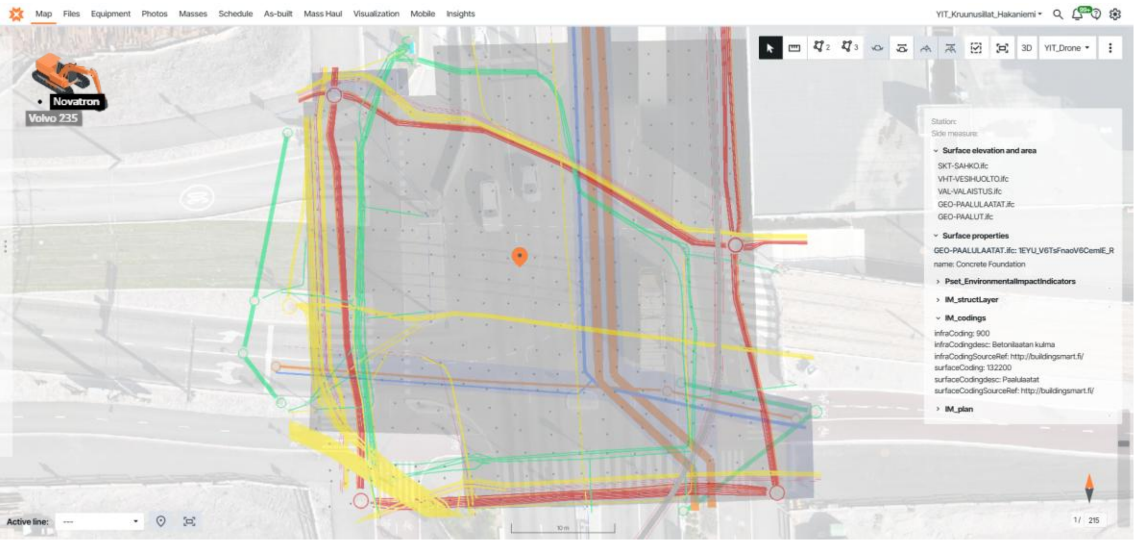


Figure 67. LandXML models in Landnova machine control simulator

4.1.2.3.2 Infrakit

Infrakit is a comprehensive ecosystem for construction site management, with a strong track record in infrastructure construction projects, such as Jokeri and Crown Bridges. It also offers cross-system integration with machine control tools such as Xsite and supports both LandXML and IFC.

Infrakit features two model-based layout options: a 2D map interface for background drawings and vehicle coordination, and a 3D viewer for model inspection. Both layouts display metadata.



Station:
Side measure:

Surface elevation and area
SKT-SAHKO.ifc
VHT-VESIHUOLTO.ifc
VAL-VALAISTUS.ifc
GEO-PAALULAATAT.ifc
GEO-PAALUT.ifc

Surface properties
GEO-PAALULAATAT.ifc: 1EYU_V6TsFnaoV6CemlE_R
name: Concrete Foundation

Pset_EnvironmentalImpactIndicators

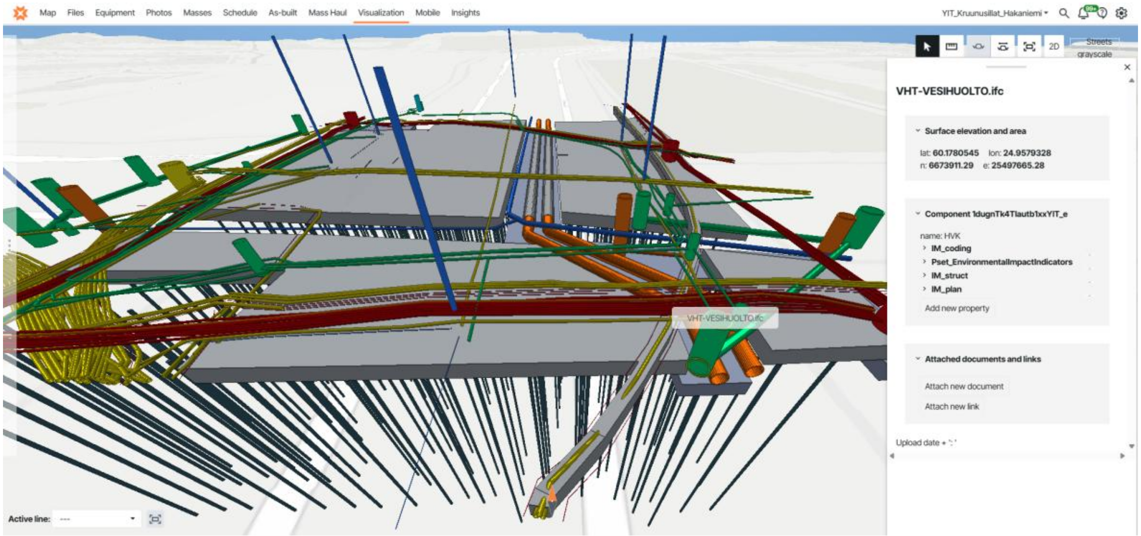
IM_structLayer

IM_codings
infraCoding: 900
infraCodingdesc: Betonilaatan kulma
infraCodingSourceRef: http://buildingsmart.fi/
surfaceCoding: 132200
surfaceCodingdesc: Paalulaatat
surfaceCodingSourceRef: http://buildingsmart.fi/

IM_plan

Figure 68. IFC models in Infrakit Map layout

IFC testing brought clear results in both modes. The 2D layout provided a schematic representation with correct dimensions and surface profiles, as shown in figure 68, while the 3D viewer rendered a conventional scene with all volumetric details preserved, as shown in figure 69. Metadata from imported entities was accessible in both cases.



VHT-VESIHUOLTO.ifc

Surface elevation and area

lat: 60.1780545 lon: 24.9579328
n: 6673911.29 e: 25497665.28

Component 1dugnTk4Tlautb1xxYIT_e

name: HVK
- IM_coding
- Pset_EnvironmentalImpactIndicators
- IM_struct
- IM_plan

Add new property

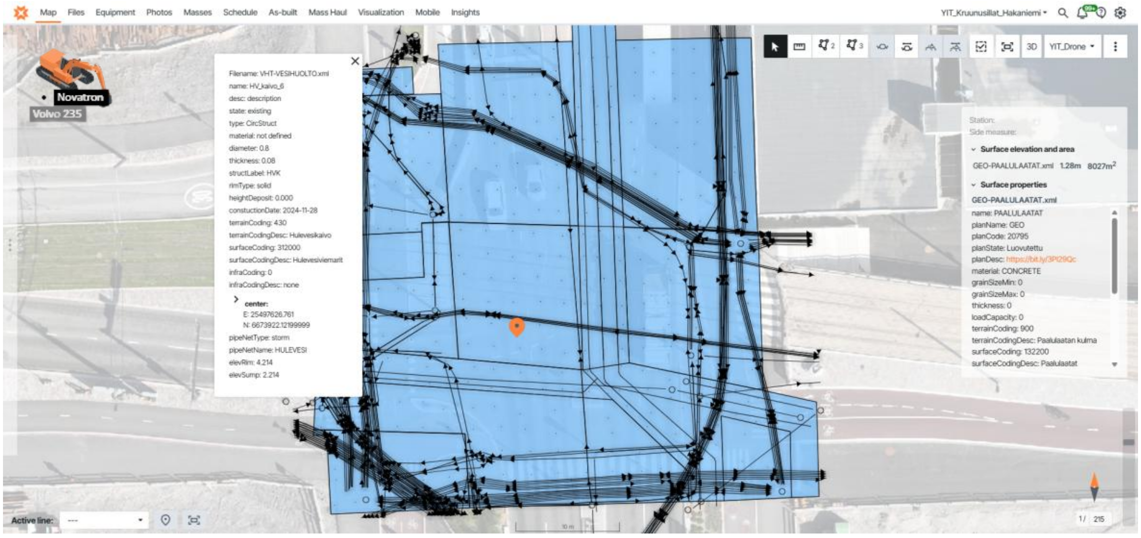
Attached documents and links

Attach new document
Attach new link

Upload date

Figure 69. IFC models in Infrakit 3D Visualisation layout

LandXML import results were mildly different. In the map view, pipe objects appeared as vectors and manholes as schematic symbols, without colour or lateral dimensions, as figure 70 confirms. Pile cap objects representation, in contrast, included accurate 2D dimensions and mesh colouring.



Filename: VHT-VESIHUOLTO.xml
name: HV_kaivo_6
desc: description
state: existing
type: CircStruct
material: not defined
diameter: 0.8
thickness: 0.08
structLabel: HVK
rimType: solid
heightDeposit: 0.000
constuctionDate: 2024-11-28
terrainCoding: 430
terrainCodingDesc: Hulevesikaivo
surfaceCoding: 312000
surfaceCodingDesc: Hulevesiviemarit
infraCoding: 0
infraCodingDesc: none
center:
E: 25497626.761
N: 6673922.12199999
pipeNetType: storm
pipeNetName: HULEVESI
elevRim: 4.214
elevSump: 2.214

Station:
Side measure:

Surface elevation and area

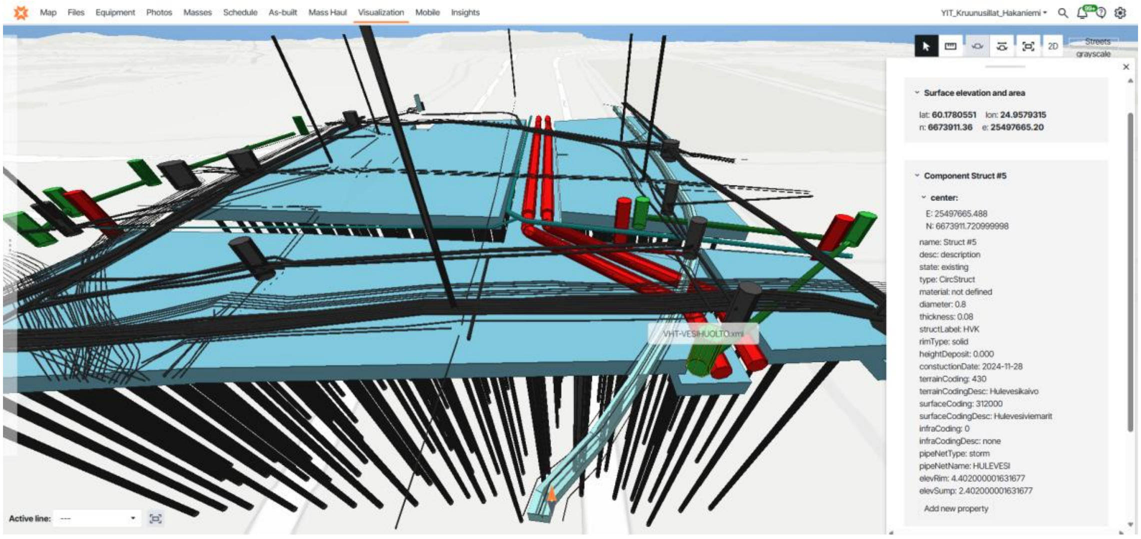
GEO-PAALULAATAT.xml 1.28m $8027 m^2$

Surface properties

GEO-PAALULAATAT.xml
name: PAALULAATAT
planName: GEO
planCode: 20795
planState: Luovutettu
planDesc: https://bit.ly/3Pl29Qc
material: CONCRETE
grainSizeMin: 0
grainSizeMax: 0
thickness: 0
loadCapacity: 0
terrainCoding: 900
terrainCodingDesc: Paalulaatan kulma
surfaceCoding: 132200
surfaceCodingDesc: Paalulaatat

Figure 70. LandXML models in Infrakit Map layout

The 3D layout view was similar to IFC, though extruded objects were simplified into closed cylinders with no wall thickness or inner diameter, as shown in figure 71. Also, metadata remained accessible in both layouts.



Surface elevation and area

lat: **60.1780551** lon: **24.9579315**
n: **6673911.36** e: **25497665.20**

Component Struct #5

center:
E: 25497665.488
N: 6673911.720999998
name: Struct #5
desc: description
state: existing
type: CircStruct
material: not defined
diameter: 0.8
thickness: 0.08
structLabel: HVK
rimType: solid
heightDeposit: 0.000
constuctionDate: 2024-11-28
terrainCoding: 430
terrainCodingDesc: Hulevesikaivo
surfaceCoding: 312000
surfaceCodingDesc: Hulevesiviemarit
infraCoding: 0
infraCodingDesc: none
pipeNetType: storm
pipeNetName: HULEVESI
elevRim: 4.402000001631677
elevSump: 2.402000001631677

Active line: ----

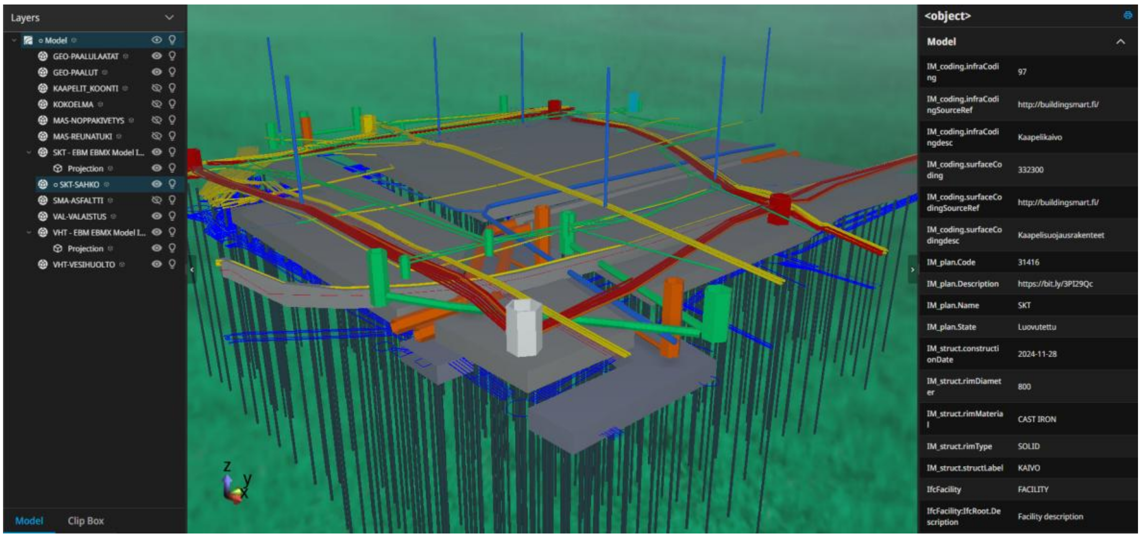
Figure 71. LandXML models in Infrakit 3D Visualisation layout

4.1.2.4 Asset Management and Maintenance

4.1.2.4.1 Cadmatic

Cadmatic eShare is a digital twin enabling platform, that integrates BIM models with sensor data, maintenance and asset management services. It was included into this study due to its potential relevance to the civil construction branch, as built assets technically can be managed and operated using this platform. Regarding its format compatibility, it imports IFC models only.

Model import into the system architecture is not a straightforward upload and requires customer support assistance. However, after the models have been uploaded it resulted in a clear 3D scene with all geometric details rendered accurately, as shown in figure 72. All metadata fields and presets were imported as well, and further import adjustments allow integration of hierarchies and cross-entity dependencies.



IM_coding.infraCoding	97
IM_coding.infraCodingSourceRef	http://buildingsmart.fi/
IM_coding.infraCodingdesc	Kaapelikaivo
IM_coding.surfaceCoding	332300
IM_coding.surfaceCodingSourceRef	http://buildingsmart.fi/
IM_coding.surfaceCodingdesc	Kaapelisuojausrakenteet
IM_plan.Code	31416
IM_plan.Description	https://bit.ly/3PI29Qc
IM_plan.Name	SKT
IM_plan.State	Luovutettu
IM_struct.constructionDate	2024-11-28
IM_struct.rimDiameter	800
IM_struct.rimMaterial	CAST IRON
IM_struct.rimType	SOLID
IM_struct.structLabel	KAIVO
IfcFacility	FACILITY
IfcFacility:IfcRoot.Description	Facility description

Figure 72. IFC models in Cadmatic eShare

4.1.3 Overview of Geometry and Metadata Readability

This overview concludes the first and the most extensive research category of this thesis. The main and the most apparent observation concerns the differences in model rendering across platforms. IFC models are consistently visualised in the same way, regardless of the software used. Extruded elements always have properly defined section profiles and the surfaces always retain their colouring. In contrast LandXML rendering varies unpredictably. In some platforms, pipes elements are shown as closed cylinders while manholes have circle hollow section, whereas in others the opposite occurs. Certain solutions fail to render vertical or overlapping TIN mesh faces, even when they are included in the model for a reason.

Another inconsistency is that some applications display pipe connectors, despite these being abstract reference entities that should not appear in the scene. In terms of LandXML syntax, this misinterpretation may arise because both elements are defined within the Struct class. The distinction, however, is that a manhole entity must contain a child element that specifies the relevant volumetric parameters, for instance CircStruct or RectStruct, whereas a pipe connector entity instead adopts a child element named Connection, which contains no volumetric properties and explicitly signals that its parent entity is abstract. [60.]

Further testing revealed that only a subset of examined applications performed any LandXML validation. This was assessed through attempting to remove pipe connectors entirely, even though schema defines them as mandatory. While some applications refused to render the corrupt model, others ignored schema compliance altogether and rendered model as usual.

This indicates a significant problem for the infrastructure domain: a lack of standardisation in LandXML import. One explanation is that, unlike IFC, there are no officially recommended tools, endorsed by a governing body like buildingSMART. As a result, the software developers often create their own LandXML interpreters. By contrast, IFC benefits from schema – aligned libraries such as IfcOpenShell and ThatOpenEngine.

A less obvious yet equally important finding is related to the entity identifiers. The unique ID used in IFC ensures that two elements with identical parameters can be distinguished. LandXML, however does not use mandatory identifiers. Instead, each object must have a unique Name element. Maintaining this uniqueness is difficult, particularly in native editors where names are assigned manually. The same challenge appears in metadata: if two metadata properties belonging to the same entity share the same name - even when located in different presets - many applications display only one of them, treating the other as redundant.

The third major observation applies primarily to the modelling applications, which often struggle with displaying metadata. This may be due to prioritising 3D modelling performance or underestimating the importance of metadata in favour of visual parameters. Whatever the reason is, metadata is a crucial component of as-built models, especially when they are used as initial data source in the design process. One possible solution to eliminate this issue, is a double screen workflow, with the modelling application on one screen and a viewer displaying the associated metadata on the other. An example of such combination, featuring Tekla Structures and Trimble Connect, is illustrated in figure 73.

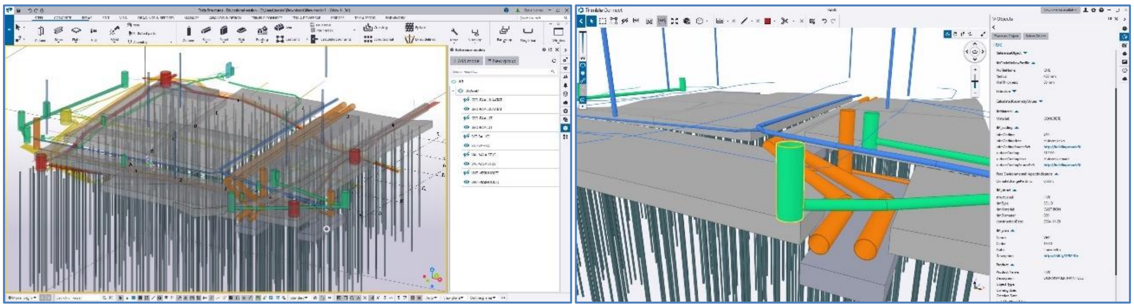


Figure 73. Double screen work interface example: a modelling application and a model viewer

It is also worth noting that the software selection in this study directly reflects the availability of solutions for specific tasks. While modelling and viewing are well covered, land surveying and asset management offer far fewer options.

Overall, the key takeaway from this research is that neither format guarantees readability across all platforms. While IFC demonstrated significantly broader applicability in terms of both geometry and metadata and shows strong potential to eliminate need for LandXML in the future, some applications still do not support IFC at all or only operate with older schema versions.

At present, this means that data cannot yet move seamlessly across all phases of asset lifecycle - design, construction, operation, demolition - without changing formats, but, on a positive note, there is at least one application for each phase that supports BIM-driven workflow. The overall performance of the formats is summarised in table 2. The compared features are extruded object visualisation, hull object visualisation, display of generic properties, and display of additional metadata presets. Green indicates excellent readability, yellow partial readability, red no visualisation and X format incompatibility.

Table 2. Format performance scores across platforms

		Modelling							Viewing					Site		Operation
		Novapoint	Civil3D	Tekla	Blender	3DWin	SimpleBIM	Rhino	Vektor	TrimbleConnect	Solibri	Dalux	Sova3D	Infrakit	Xsite	Cadmatic
XML (25.5)	Extruded (7.5)	🟨	🟨	🟩	X	🟨	X	X	🟩	🟩	X	X	🟩	🟩	🟩	X
	Hull (7.5)	🟨	🟨	🟩	X	🟨	X	X	🟩	🟩	X	X	🟩	🟩	🟩	X
	Generic (6.5)	🟩	🟨	🟨	X	🟩	X	X	🟩	🟩	X	X	🟨	🟩	🟥	X
	Psets(4)	🟥	🟥	🟥	X	🟩	X	X	🟩	🟨	X	X	🟨	🟩	🟥	X
IFC (47)	Extruded (13)	🟩	X	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	X	🟩
	Hull (13)	🟩	X	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	X	🟩
	Generic(11.5)	🟨	X	🟩	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	X	🟩
	Psets (9.5)	🟥	X	🟥	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	X	🟩

4.2 Suitability for Structure Gauge Analysis

Structure gauge analysis suitability examination employed qualitative and comparative research methods. The qualitative techniques were applied to assess the general applicability of as-built models within a structure gauge inspection workflow, while the comparative element evaluated their performance compared to the point cloud approach, with reference to the earlier work described in chapter 3.

The testing was primarily human-driven. The qualitative segment involved manually importing models into the selected platform, visually assessing the displayed geometries, and conducting a clash analysis in both the 3D viewport and the cross-section view. The comparative segment, in turn, examined model and pointcloud datasets in terms of storage space, visual properties, and suitability for the structural gauge check. Observations and findings were documented throughout, followed by deriving the conclusions.

4.2.1 Research Data

Structure gauge analysis primarily concerns superstructure elements that may intrude into the clearance envelope. Among the available data, the traffic sign models were found particularly suitable for this purpose. In addition to the as-built models, structure gauge envelope models were also produced as part of the quality assurance documentation for the Crown Bridges Light Rail project. The following models were included in this examination:

- Traffic Signs (3D elements)
- Maximum Static Rolling Stock Outline (Envelope model)
- Clearance Base Operating Standard (Envelope model)
- Minimum Structure Outline (Envelope model)

For research clarity, the testing area was selected from a straight section of the track alignment. Traffic sign models were extracted from superstructure as-built dataset and consist of series of signs distributed along the track, supplemented by annotation drawings. The models were produced using Plainview software.

The clearance envelope models were produced from three input components:

- Track alignment models, obtained from as-designed data
- As-built rail geometry models, obtained by land survey
- Dimension scheme for the clearance envelopes produced by the project owners.

The modelling process relied on 3DWin for alignment manipulations, and Plainview application for generating the envelope meshes. The applied envelope dimensions are illustrated in figure 74.

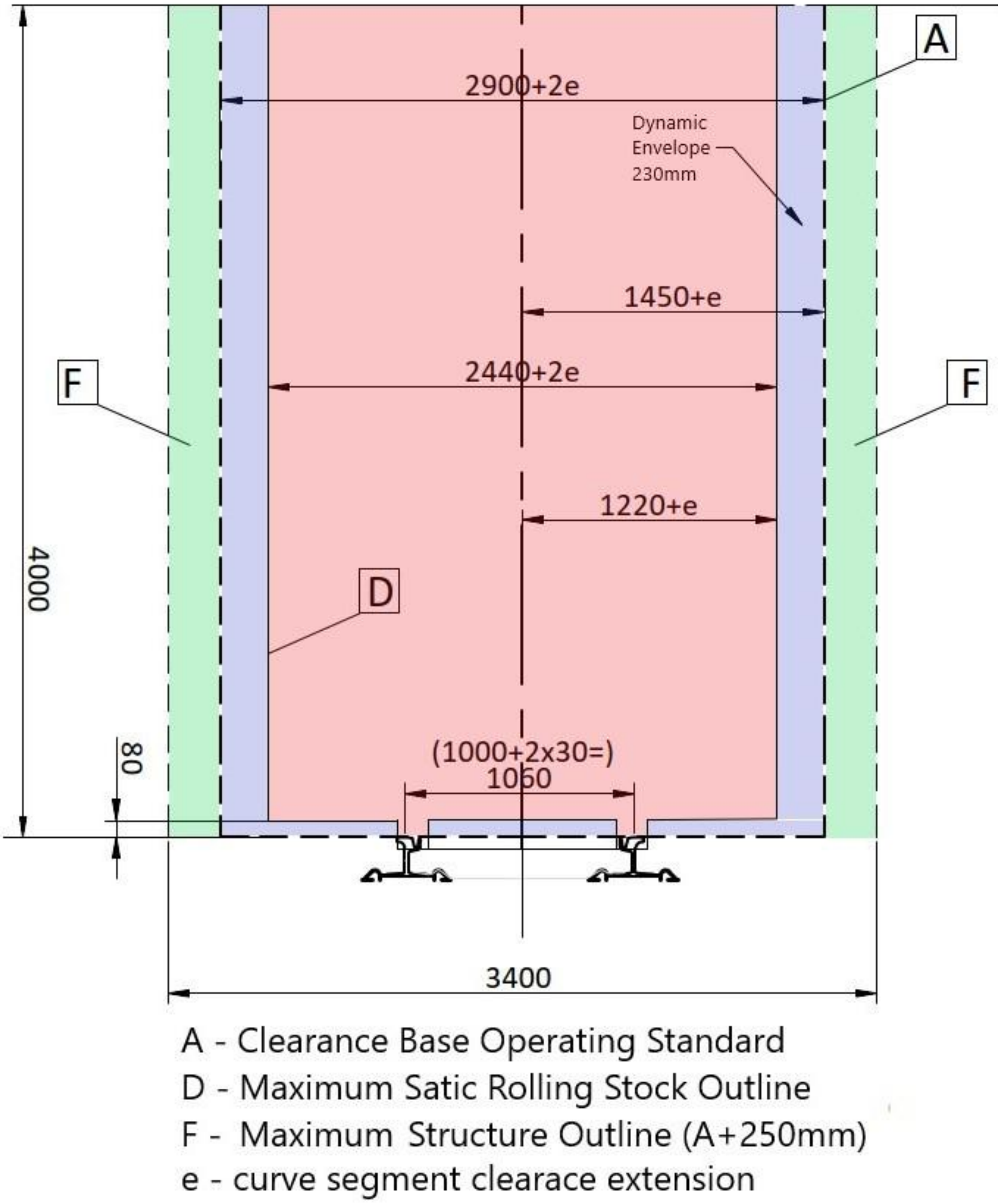


Figure 74. Structure gauge dimensions applied in light rail projects in Finland [80].

The Maximum Static Outline refers to the space occupied by a streetcar while stopped or moving at crawling speeds and is applied primarily for station inspections. The Clearance Base Standard, in turn, defines the space occupied during regular operation, and is applied to the stretches between the stations. If clearance base standard gauge is by mistake used at for the station inspection, the platform structures will intrude into this envelope. [80.] Finally, the minimum structure outline represents a project-specific gauge, defined as the clearance base + 250mm, as an additional safety margin. While structures are technically permitted to cut into this envelope, each case requires inspection followed by a formal approval. A schematic 2D view of traffic sign model and an envelope model used in this test, is illustrated in figure 75.

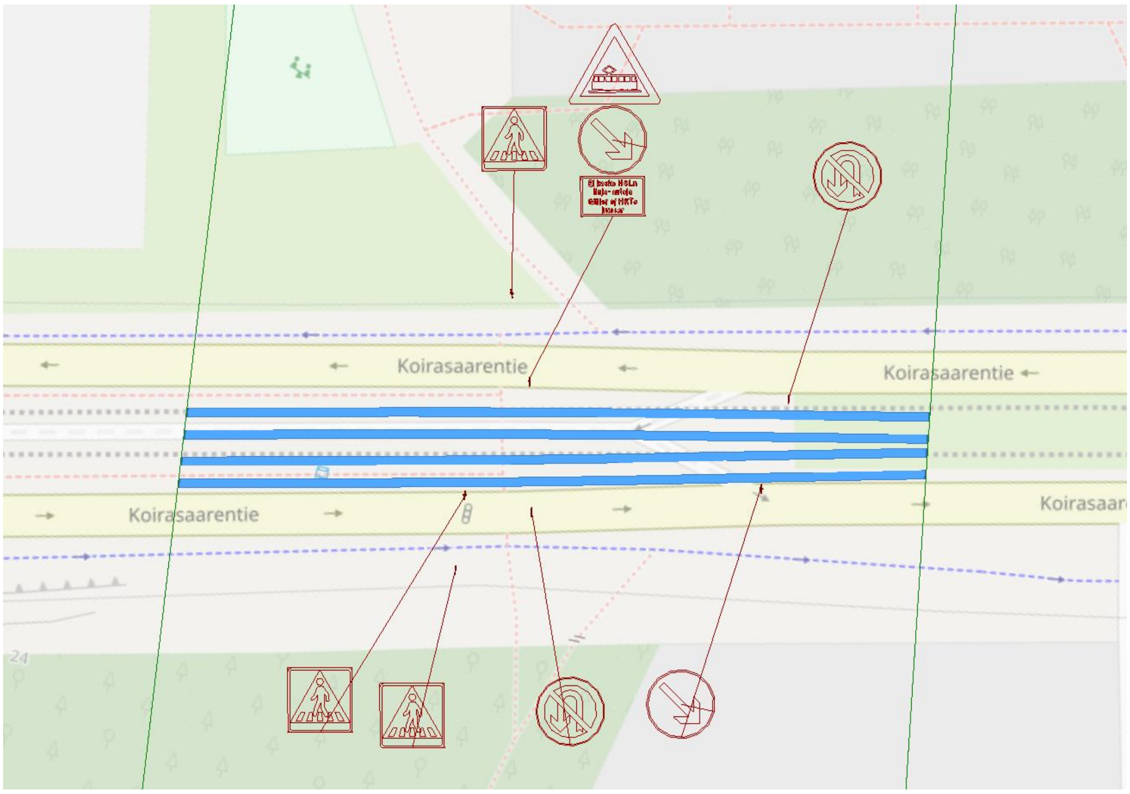


Figure 75. A schematic view of the traffic sign and clearance envelope model combination

Curved parts of the track were excluded from the examination for better comprehensibility. However, it is worth mentioning that the light rail design guidelines provide the list of clearance coefficients, for various curve radii. The sharper the curve, the greater the extension, with maximum value of 600mm at radii of 19-25m. [80.] A schematic representation of this behaviour is presented in figure 76.

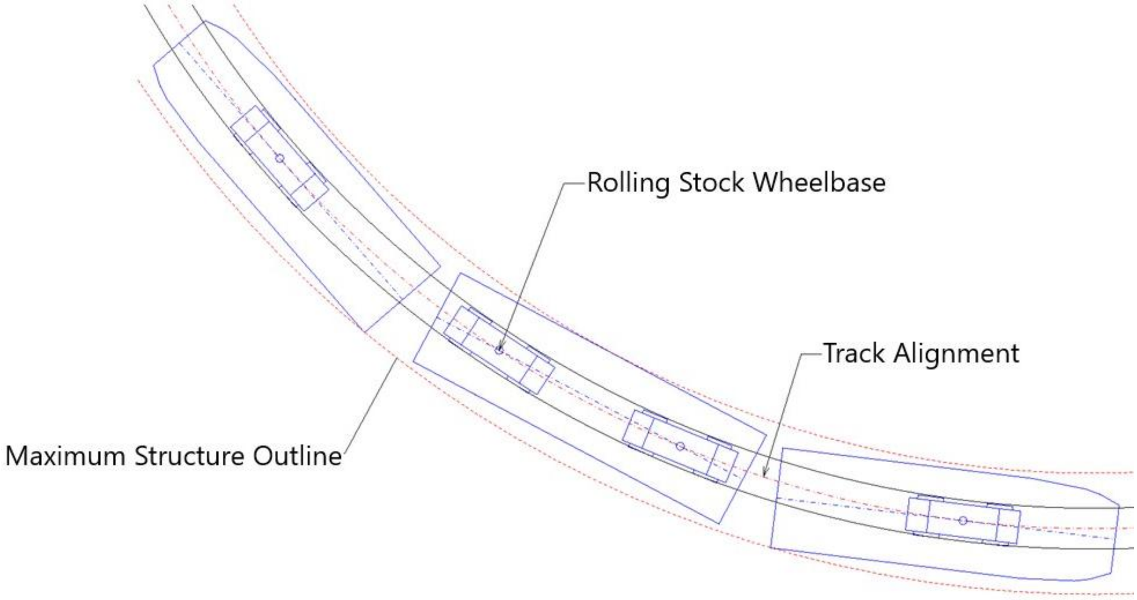


Figure 76. A schematic view of the traffic sign and clearance envelope model combination [80].

Both the envelope and traffic sign models are stored in LandXML format and essentially are compliant with Inframodel specification. Geometrically, they are represented as TIN meshes and technically can be converted into IFC, preserving all properties, if required in the future.

For an explicit demonstration of differences between BIM and point cloud representations of the same assets, an additional dataset was prepared. A longitudinal region of Jokeri Light Rail point cloud and a collection of Crown Bridges Light Rail as-built elements were aligned in an arbitrary location. The full dataset used in this comparison consisted of:

- Built infrastructure Point Cloud (LAS format)
- Built element as-Built models set (LandXML)

4.2.2 Research Framework

Vektor.io was found suitable for this analysis, as it provides all the necessary functionalities and has been actively applied in the project. In addition to efficient rendering of 3D models, alignment models and point clouds, the software offers a range of supplementary tools, including as cross-section views, shading and transparency settings, and distance measurement utilities.

4.2.2.1 Visual Inspection

Since no apparent structure gauge violations were detected from the original as-built model, the test was simulated by deliberately enlarging the dimensions of one of the traffic signs. This modification resulted in part of the sign element cutting into the minimum structure outline envelope, as shown in figure 77.

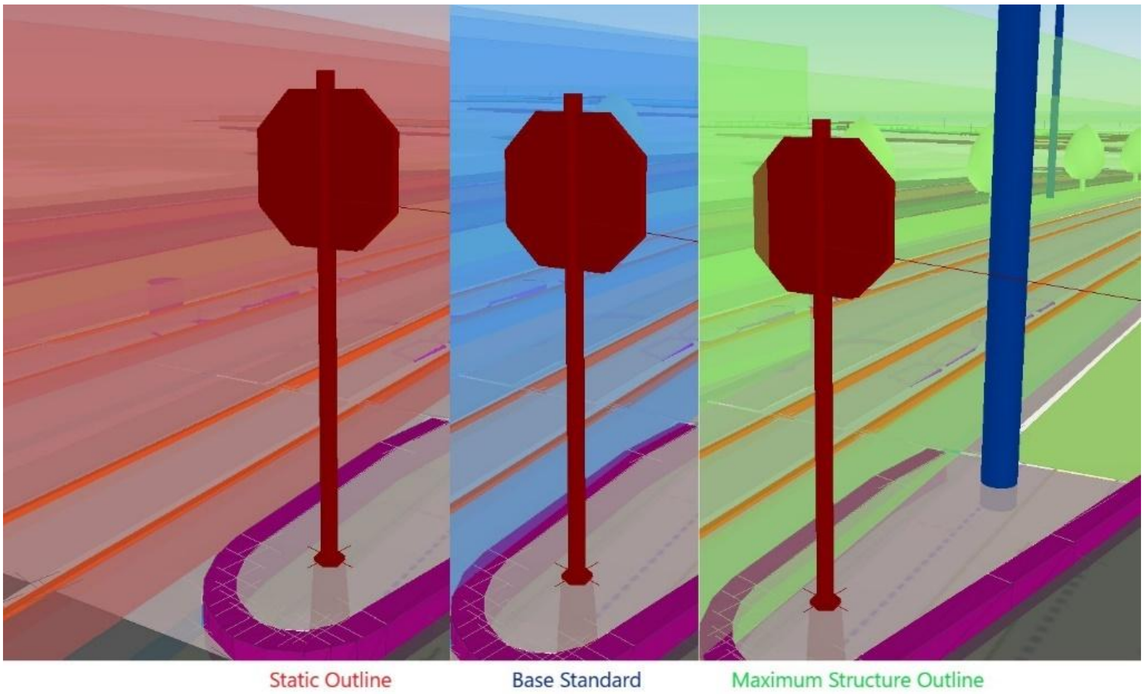


Figure 77. A traffic sign examination against the envelope models

A closer inspection of the cross-section view confirmed a gauge violation by approximately 0.100 m. Additional distance measurements performed in the 3D viewport provided a more precise value of 0.103 m. Both measurement approaches are presented in figure 78.

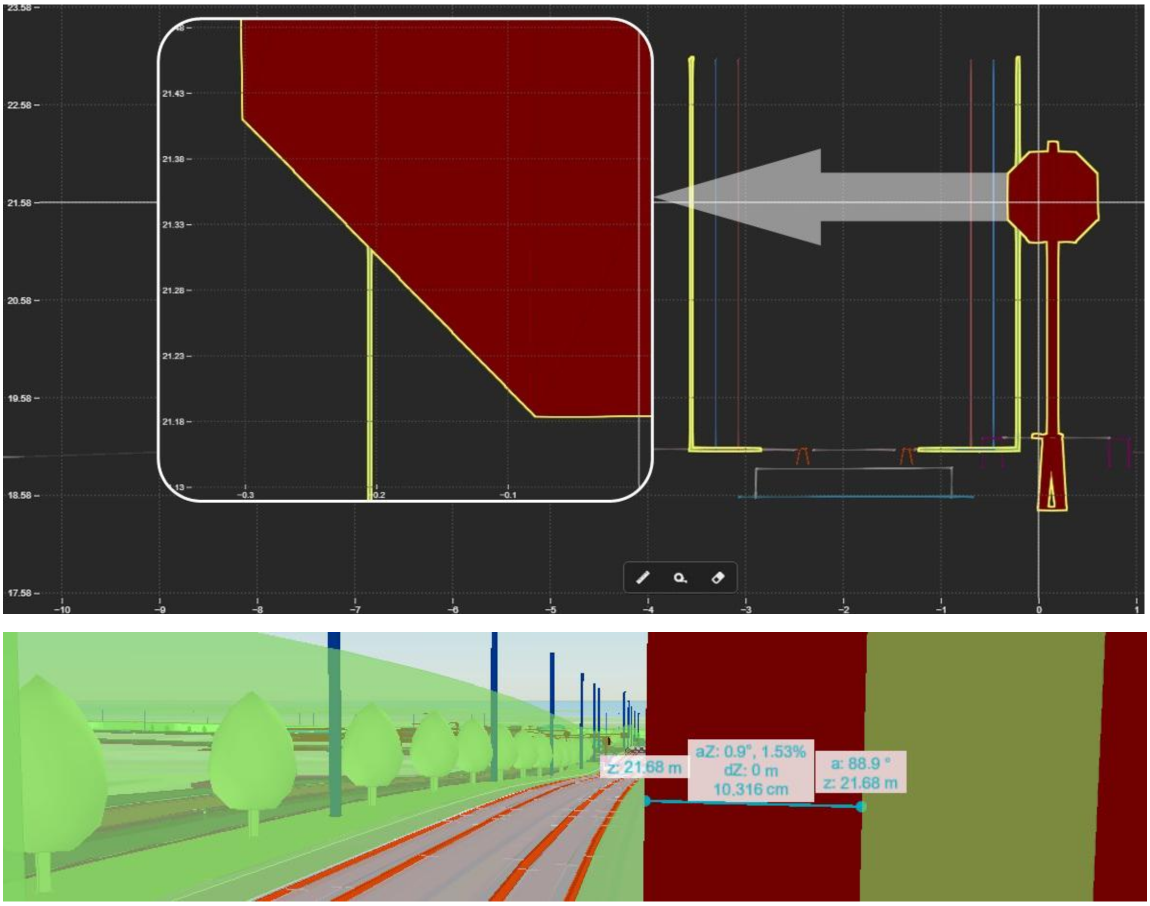


Figure 78. Cross-section view and 3D distance measurement tools in Vektor.io

4.2.2.2 Model Comparison Against Pointcloud

The reference point cloud was obtained through laser scanning, as described in chapter 3.3.3. The corresponding as-built elements were extracted from the models and aligned with the point cloud using Plainview platform.

A size comparison revealed a substantial difference between the datasets. The point cloud section occupied approximately 6.5 GB, whereas the equivalent as-built dataset was only 8.5 MB, meaning the point cloud required roughly 1000 times more storage space.

For the visual comparison the as-built elements were divided into two shape types: permanent and impermanent, based on their ability to change shape with time. Permanent shape objects, represented in this study by a cabinet and a railing, showed strong correspondence in both placement and geometry, as demonstrated in figure 79. Furthermore, since the models allow subsequent modifications, imperfections can be corrected within a modelling application.

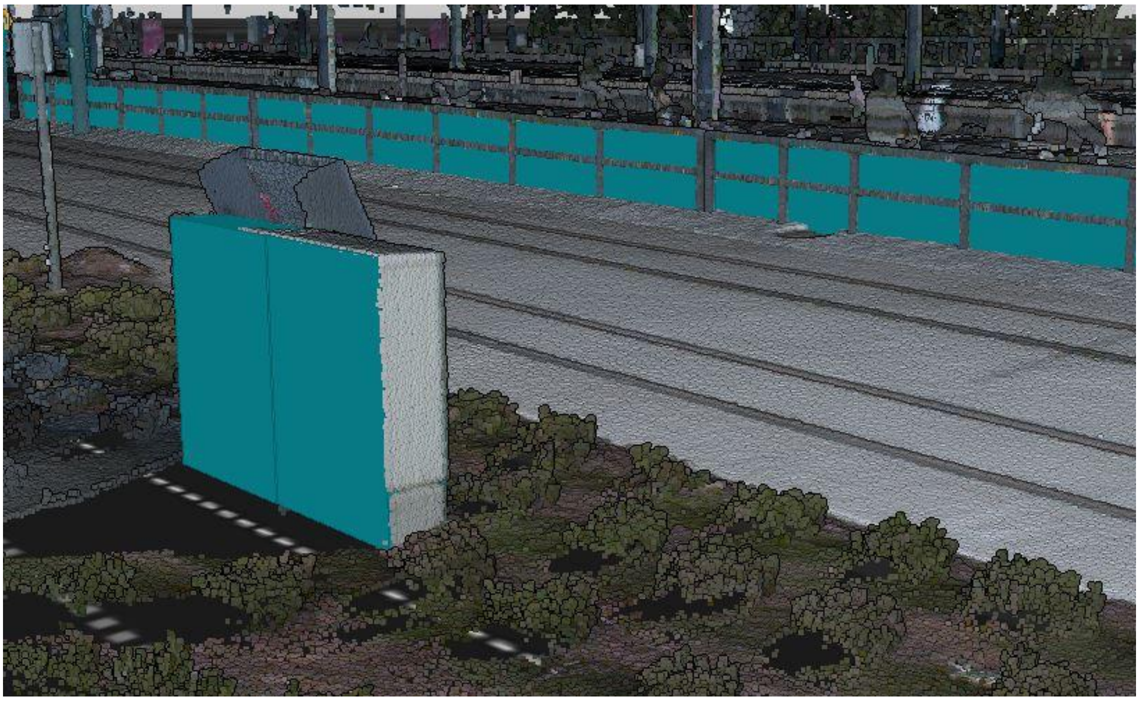


Figure 79. The point cloud and the BIM model comparison

Impermanent shape objects, in turn, were represented by the trees. These elements highlight a common issue: although they may not interfere with the clearance gauge at the time of commissioning, natural growth may cause later conflicts, potentially requiring re-modelling and re-inspection. The comparison of the tree models with the point cloud is illustrated in figure 80. For operational safety purposes, it is also possible to model the trees at their expected mature size, thereby simulating future conditions.

Figure 80. The point cloud and the BIM model comparison

After confirming the suitability of both data classes for the structure gauge analysis, their performance was further assessed through cross-section comparison. The results are presented in figure 81. Visual inspection showed no major differences in applicability for this task. The as-built cross-section appeared more clear and suitable for measurement purposes, but in both datasets the gauge violation could be easily identified.

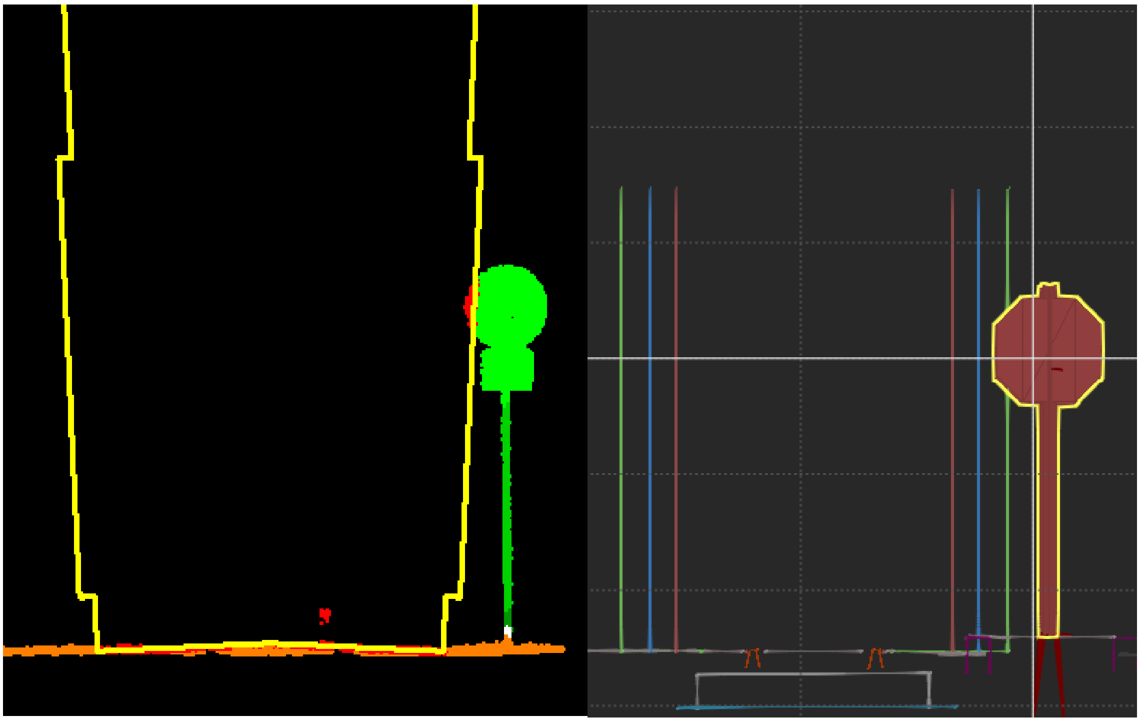


Figure 81. Structure gauge cross-sections derived from a point cloud [75] and **an as-built model**

4.2.3 Overview of Structure Gauge Analysis Suitability

The key observation of this research chapter is that as-built models represent a highly suitable dataset for structural gauge analysis. They are visually clear, compact in size, and relatively easy to produce without deviating from standard construction practices. While their geometric precision may be lower than that of a point cloud or a scan-to-BIM mesh, the additional accuracy is not critical for this type of analysis. The maximum structure outline already incorporates an additional 250mm safety margin, meaning that even minor intrusions into this envelope trigger a thorough inspection.

It should nevertheless be emphasised that potential conflicts must be revealed during the design phase, but due to potential human errors or automation misalignments, some issues may become apparent at later stages of construction. For this reason, the structure gauge analysis remains a mandatory procedure in a project delivery.

Another important conclusion is that both gauge envelope and as-built models can be produced in open formats, which technically allows the analysis to be performed on any BIM viewing platform. Certain solutions that support mobile deployment, enable the analysis to done on-site.

Finally, the main advantage of the model-based data is its low editability threshold. For instance, if a present traffic sign intrudes into the clearance envelope, a new smaller version can be modelled within minutes, to confirm that simple replacement would resolve the issue. The same principle applies to other built environment modifications, which can be simulated before implementation. Editability also enables localised inspections - modelling of only a small part of the asset - to verify that the latest adjustments comply with the safety requirements.

4.3 Digital Twin Readiness

Digital twin readiness examination was focused entirely on exploratory activities and can be classified as supplementary research. Its aim was to find a simple and implementable method for linking BIM models with sensor data streams. The success criterion was defined as achieving a dynamic connection between the model and continuously updated sensor readings.

The exploration was primarily human-driven, supported by the use of large language models (LLM) for solution seeking. The research workflow combined 3D modelling, lightweight programming, solution prototyping and iterative testing, followed by drawing a conclusion.

4.3.1 Research Data

The main challenge in assessing digital twin readiness was to identify an accessible and functioning sensor data architecture that could share readings in an open-source manner. After considering several alternatives, Sensoto and Metropolia SmartLab were selected for further evaluation.

Sensoto is a free web-based platform that provides access to open public sensor data worldwide. Available sensor categories include air and atmosphere, light and radiation, traffic, and water [81]. For this study, as-built models from the Crown Bridges project's dataset were used. A light pole model was found particularly suitable for linking local air quality data, as such a structure could realistically host an air quality sensor. The model was in LandXML format, and compliant with Inframodel 4.2 specification.

Metropolia SmatLab, in turn, is a sensor-equipped research facility located at Metropolia UAS, Myllypuro Campus. Available sensor categories include temperature, humidity, CO_2, water and energy consumption. This system was examined using corresponding SmartLab as-built model provided by Metropolia UAS. The asset was modelled in older IFC version 2.3, and a schematic 2D view of this model is presented in figure 82.

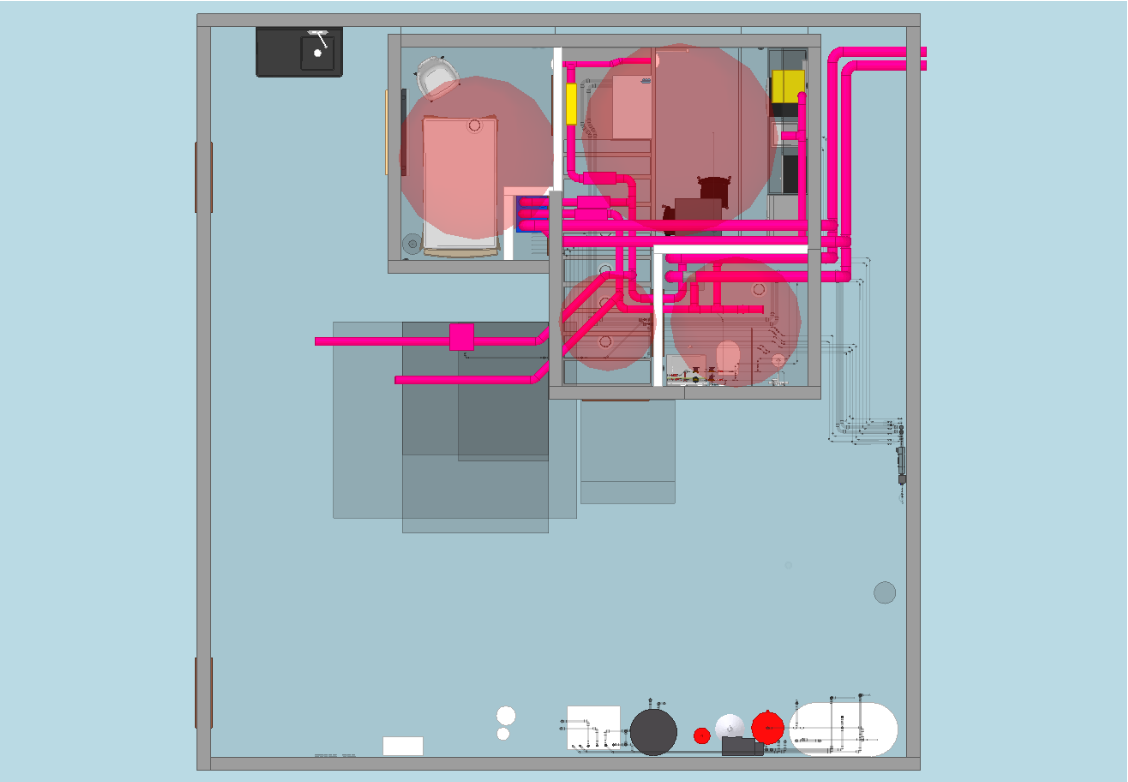
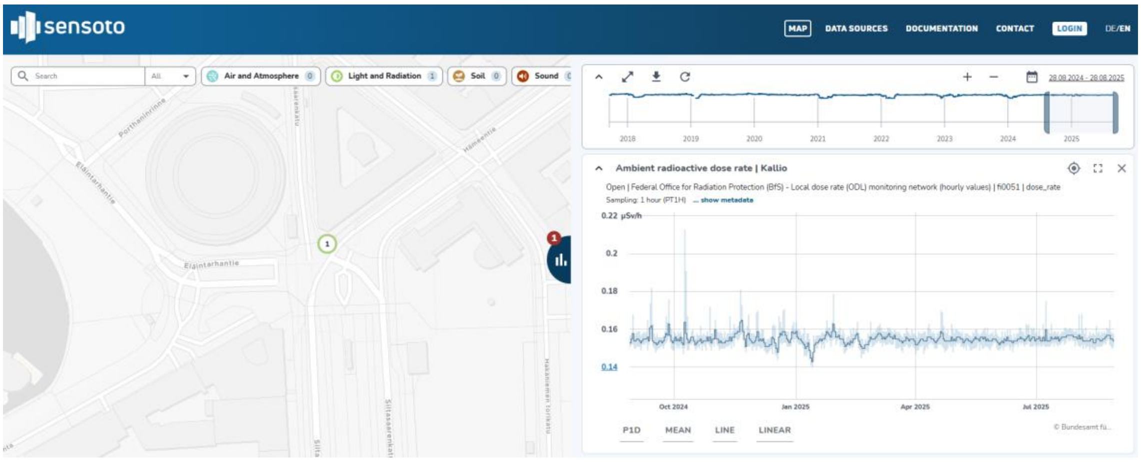


Figure 82. A schematic view of Metropolia SmartLab used in this study

4.3.2 Research Framework

4.3.2.1 Open Sensor Web

The integration concept of Sensoto was straightforward: connecting external open sensor data to a BIM model through an external link reference. A similar approach had previously been applied for linking quality assurance documentation. This required neither extensive model modifications nor custom solution development, as it used Sensoto's original viewing interface, which includes a map view with selected sensor's location and adjustable measurement history window, as shown in figure 83.



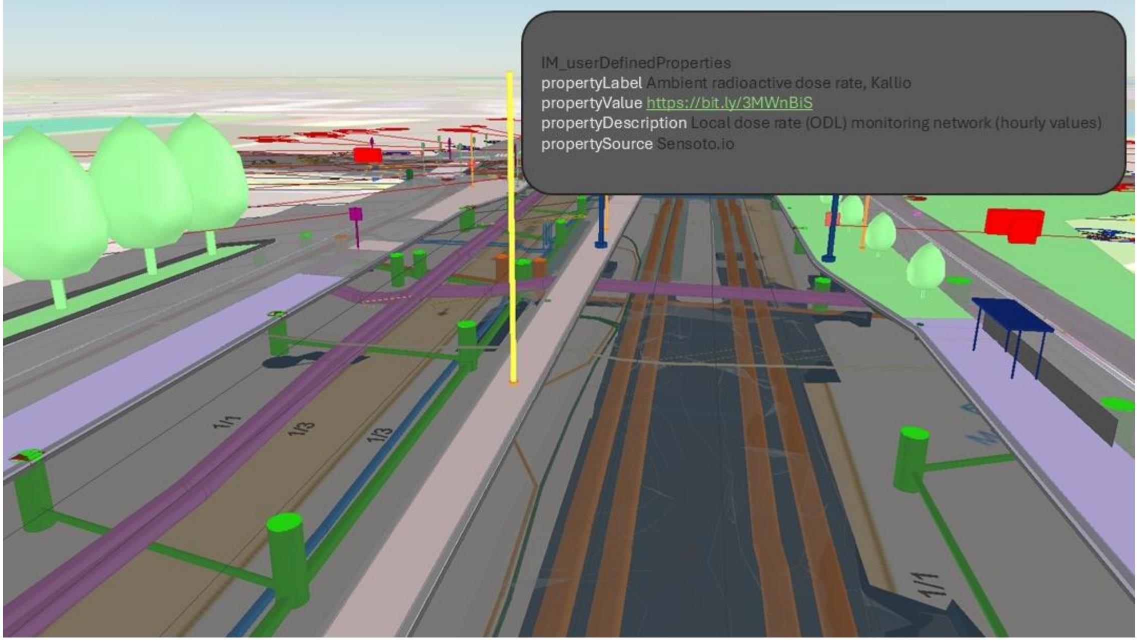
Ambient radioactive dose rate | Kallio

Open | Federal Office for Radiation Protection (BfS) - Local dose rate (ODL) monitoring network (hourly values) | fi0051 | dose_rate

Sampling: 1 hour (PT1H) ... show metadata

Figure 83. Sensor data workflow example in Sensoto

The original uniform resource locator (URL) provided by Sensoto was shortened using bit.ly service. Since neither LandXML nor Inframodel specifications natively support sensor integration, the arbitrary IM_userDefinedProperties preset was used to embed the link. Along with the link itself, the preset included sensor name, type and data source. The property set readability was confirmed in Vektor.io, as illustrated in figure 84.



IM_userDefinedProperties
propertyLabel Ambient radioactive dose rate, Kallio
propertyValue https://bit.ly/3MWnBiS
propertyDescription Local dose rate (ODL) monitoring network (hourly values)
propertySource Sensoto.io

Figure 84. Light pole as-built model with embedded open sensor data in Vektor.io

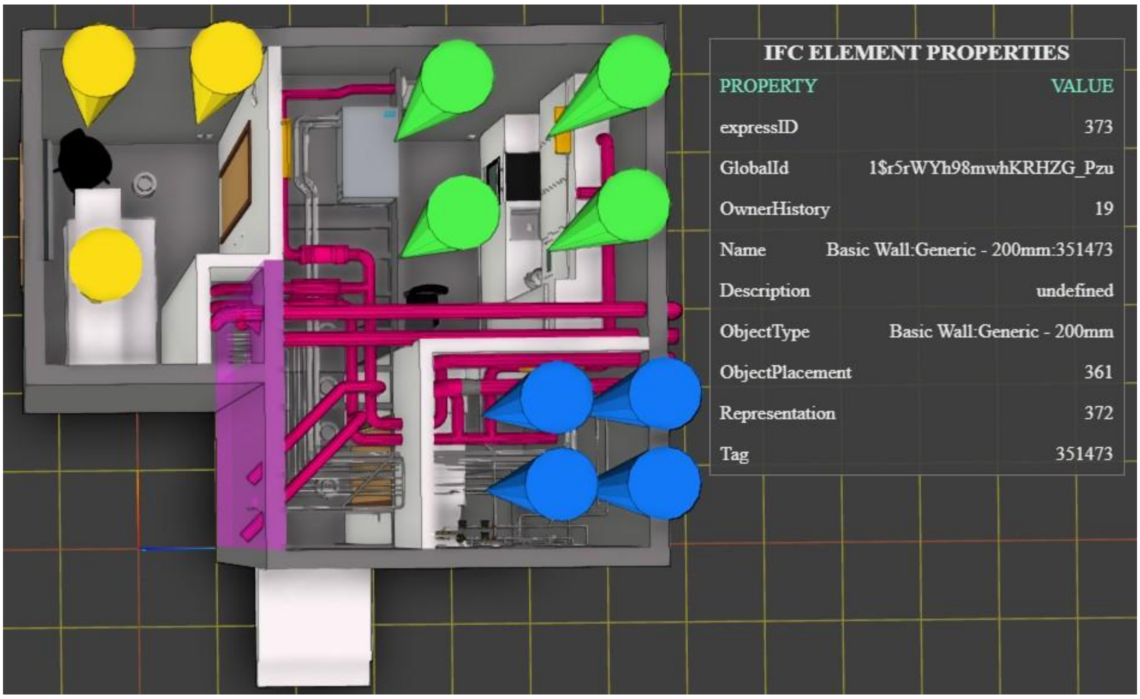
4.3.2.2 SmartLab

SmartLab sensors are accessible through the Message Queuing Telemetry Transport (MQTT), a lightweight publish/subscribe protocol, designed for the Internet of Things (IoT) applications [82]. Subscribing to a sensor requires MQTT broker address and MQTT topic, both of which were provided by Metropolia UAS.

Since the SmartLab system is significantly more complex than Sensoto, it required an equivalent approach. As a result, custom lightweight solution was developed to combine the following features:

- Browser-based interface
- BIM model loader and viewer
- Entity metadata display
- MQTT client
- Real-time sensor data visualisation
- Readings analysis and history database

For simplicity, Metropolia SmartLab BIM model was stripped from all excessive detail, resulting in a lightweight version of itself. Supplementary pointer entities were added to mark the sensor locations and store MQTT broker and topic data. The application was built using ifc.js, HTML and CSS, enabling 3D geometry and metadata rendering, as illustrated in figure 85.



IFC ELEMENT PROPERTIES

PROPERTY	VALUE
expressID	373
GlobalId	1$r5rWYh98mwhKRHZG_Pzu
OwnerHistory	19
Name	Basic Wall:Generic - 200mm:351473
Description	undefined
ObjectType	Basic Wall:Generic - 200mm
ObjectPlacement	361
Representation	372
Tag	351473

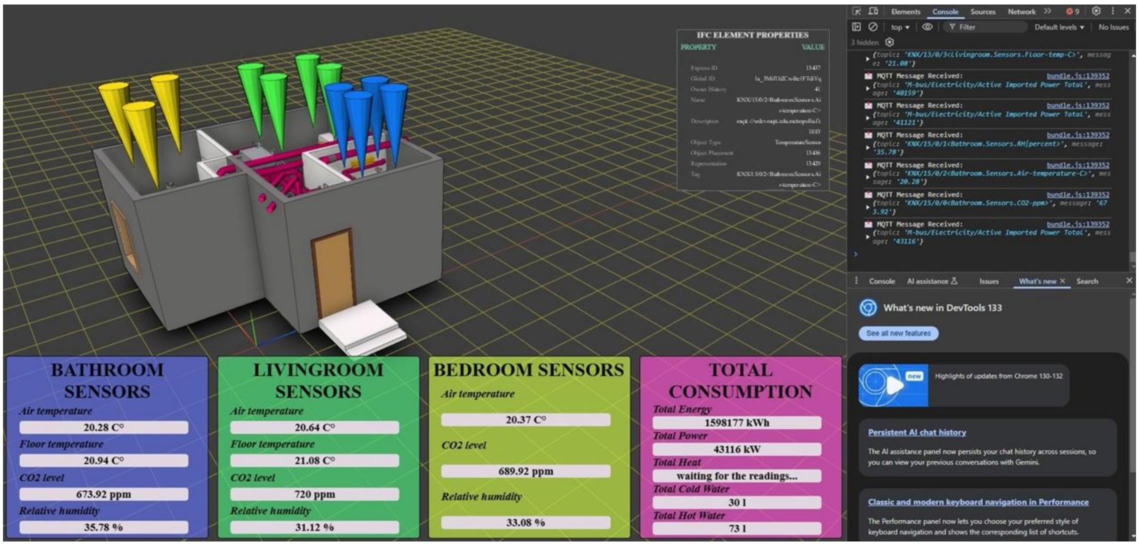
Figure 85. Metropolia SmartLab in custom-built web model viewer

Since IFC2.3 schema did not include the IfcSensor element, the IfcEquipmentElement was applied instead. Sensor identifiers were embedded into Name, Description and Tag elements, as shown in listing 11. While this was a non-standard approach that featured deprecated IFC entity, it allowed a fully functioning demonstration.

```
#13437=IFCEQUIPMENTELEMENT('1a_3Mif1b2Cwihe1FTdiYq', #41,
'KNX/15/0/2<Bathroom.Sensors.Air-temperature-C>',
'mqtt://xrdevmqtt.edu.metropolia.fi:1883', 'TemperatureSensor', #13436,
#13429, 'KNX/15/0/2<Bathroom.Sensors.Air-temperature-C>');
```

Listing 11. Improvised sensor element IFC2X3 encoding

An MQTT client was established using mqtt.js and WebSockets. figure 86 illustrates a midway prototype, that connected the BIM viewer with an MQTT client. Received readings were grouped by room and category within the main interface and logged into the browser console.



```
{topic: 'KNX/13/0/3<Livingroom.Sensors.Floor-temp-C>', message: '21.08'}
MQTT Message Received:                                              bundle.js:139352
{topic: 'M-bus/Electricity/Active Imported Power Total', message: '40159'}
MQTT Message Received:                                              bundle.js:139352
{topic: 'M-bus/Electricity/Active Imported Power Total', message: '41121'}
MQTT Message Received:                                              bundle.js:139352
{topic: 'KNX/15/0/1<Bathroom.Sensors.RH|percent>', message: '35.78'}
MQTT Message Received:                                              bundle.js:139352
{topic: 'KNX/15/0/2<Bathroom.Sensors.Air-temperature-C>', message: '20.28'}
MQTT Message Received:                                              bundle.js:139352
{topic: 'KNX/15/0/0<Bathroom.Sensors.CO2-ppm>', message: '673.92'}
MQTT Message Received:                                              bundle.js:139352
{topic: 'M-bus/Electricity/Active Imported Power Total', message: '43116'}
```

Persistent AI chat history

The AI assistance panel now persists your chat history across sessions, so you can view your previous conversations with Gemini.

Classic and modern keyboard navigation in Performance

The Performance panel now lets you choose your preferred style of keyboard navigation and shows the corresponding list of shortcuts.

BATHROOM SENSORS
Air temperature: 20.28 C°
Floor temperature: 20.94 C°
CO2 level: 673.92 ppm
Relative humidity: 35.78 %

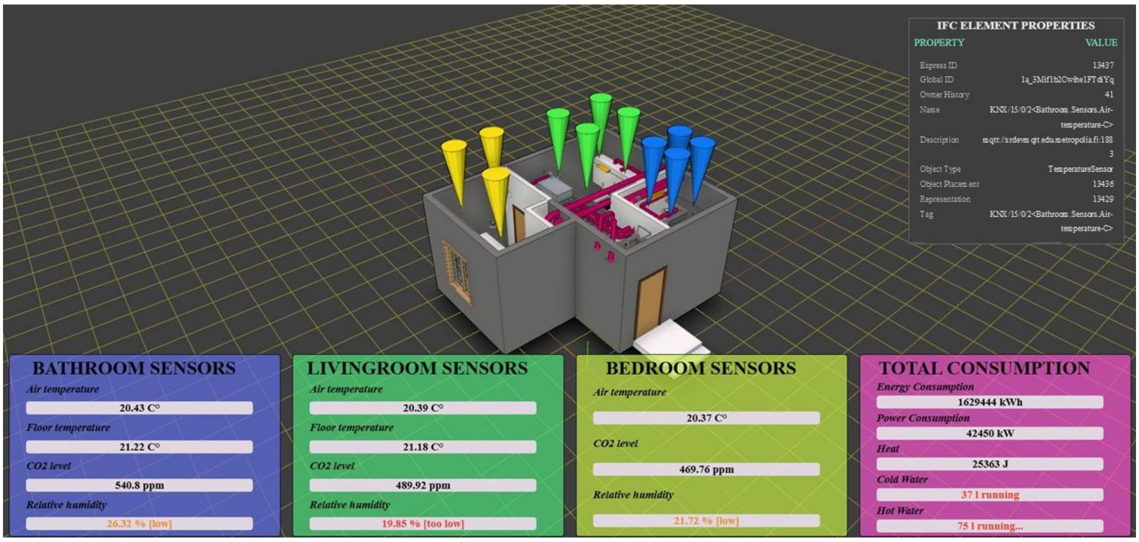
LIVINGROOM SENSORS
Air temperature: 20.64 C°
Floor temperature: 21.08 C°
CO2 level: 720 ppm
Relative humidity: 31.12 %

BEDROOM SENSORS
Air temperature: 20.37 C°
CO2 level: 689.92 ppm
Relative humidity: 33.08 %

TOTAL CONSUMPTION
Total Energy: 1598177 kWh
Total Power: 43116 kW
Total Heat: waiting for the readings...
Total Cold Water: 30 l
Total Hot Water: 73 l

Figure 86. IFC model viewer with integrated MQTT client

To complete the workflow, further features were added: threshold annotations for the condition readings and warning annotation for the consumption readings, highlighted with relevant colours. The final version of the application’s main page is shown in figure 87.



IFC ELEMENT PROPERTIES

PROPERTY	VALUE
Express ID	13437
Global ID	1a_3Mif1b2Cwihe1FTdiYq
Owner History	41
Name	KNX/15/0/2<Bathroom.Sensors.Air-temperature-C>
Description	mqtt://xrdevmqtt.edu.metropolia.fi:1883
Object Type	TemperatureSensor
Object Placement	13436
Representation	13429
Tag	KNX/15/0/2<Bathroom.Sensors.Air-temperature-C>

BATHROOM SENSORS
Air temperature: 20.43 C°
Floor temperature: 21.22 C°
CO2 level: 540.8 ppm
Relative humidity: 26.32 % [low]

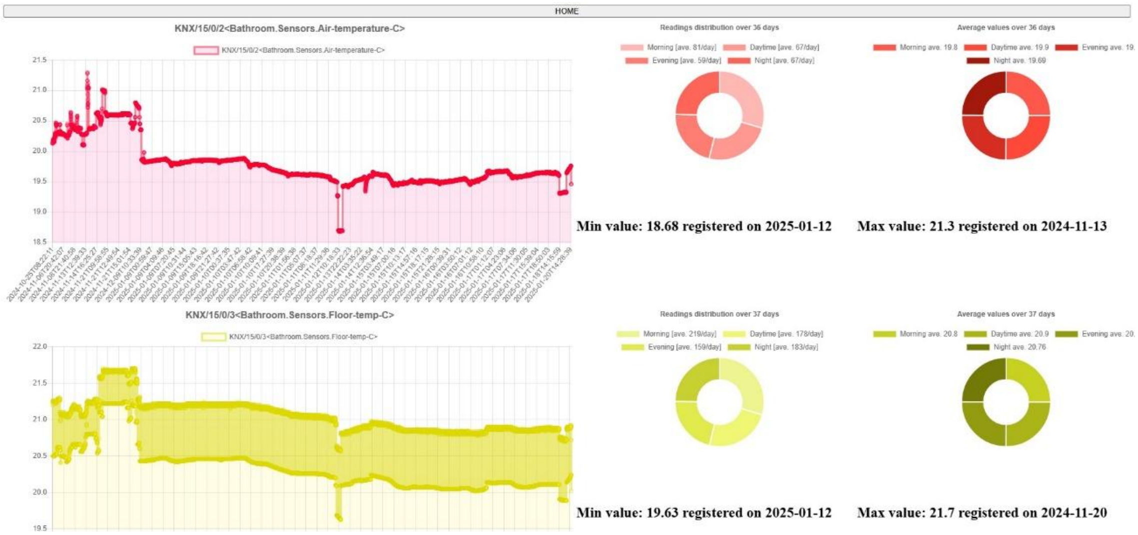
LIVINGROOM SENSORS
Air temperature: 20.39 C°
Floor temperature: 21.18 C°
CO2 level: 489.92 ppm
Relative humidity: 19.85 % [too low]

BEDROOM SENSORS
Air temperature: 20.37 C°
CO2 level: 469.76 ppm
Relative humidity: 21.72 % [low]

TOTAL CONSUMPTION
Energy Consumption: 1629444 kWh
Power Consumption: 42450 kW
Heat: 25363 J
Cold Water: 37 l running
Hot Water: 75 l running...

Figure 87. Main application view with annotated readings

Room-specific insight pages were also included into the framework. They displayed historical readings, average and extreme values, and distribution insights. An example for bathroom sensors is presented in figure 88.



Min value: 18.68 registered on 2025-01-12

Max value: 21.3 registered on 2024-11-13

Min value: 19.63 registered on 2025-01-12

Max value: 21.7 registered on 2024-11-20

Figure 88. Insight page for the bathroom sensors

4.3.3 Overview of Digital Twin Readiness

The performed examinations demonstrated that both sensor integration approaches are feasible, though each comes with a set of distinct conditions. Open sensor data proved to be simple to implement and required no programming skills, but it came with a number of limitations. Its functionality depends entirely on third-party platforms, which allow no customisation, and available sensor types are largely generic, such as weather or air quality. A potential use case could be a bridge model, enriched with air temperature, wind speed and relative humidity data, as these factors directly influence asset's structural health monitoring and maintenance scheduling.

In contrast, MQTT-based architecture, required programming knowledge but offered significantly greater flexibility and scalability. It enabled integration of any sensor type and supported custom data analytics, advanced visualisation, triggers and predictive insights. Although the demonstration was limited to the IFC format, the strategy applied in this study can be projected onto other schemas, enabling similar workflows for both standard LandXML and Inframodel-extended models.

Finally, it should be noted that reliable use of sensor data requires integrity and continuity control, as well as stable and secure sensor architecture. Establishing such systems extends beyond the competence of a general BIM engineer and requires collaboration with IoT specialists.

4.4 Suitability for AI driven Examination

Similar to the previous research segment, this part of the study relies on exploratory research methods and complements the main investigation. Its purpose is to explore potential applications of artificial intelligence algorithms directly on as-built models. The intended outcome is the development of a workflow capable of inspecting models, detecting possible anomalies in geometry or metadata, and generates predictive insights.

This study phase was divided into two parts: Anomaly Detection and Predictive Insight Generation. Both parts relied heavily on AI techniques, though human-driven input was not excluded. The research process involved extensive use of large language models for solution seeking and selection, while prediction accuracy was further evaluated using AI-based tools. The overall workflow combined information gathering, data cleaning, AI-related Python scripting, prototyping, and testing with sanity checks. Observations were documented throughout the process and compiled into a summarised conclusion.

4.4.1 Research Data

To ensure the versatility in testing, both LandXML and IFC formats were included into the examination. The chosen case model was an as-built stormwater pipe network. This model was selected for two reasons: first, because stormwater systems are among the most common infrastructure assets and typically present in most facilities, and second, because they contain rich embedded metadata that can be utilised for AI-enhanced examination. These characteristics ensure that the solutions and workflows developed during this research can be projected on other infrastructure BIM models. A schematic 2D view of the selected case model is illustrated in figure 89.

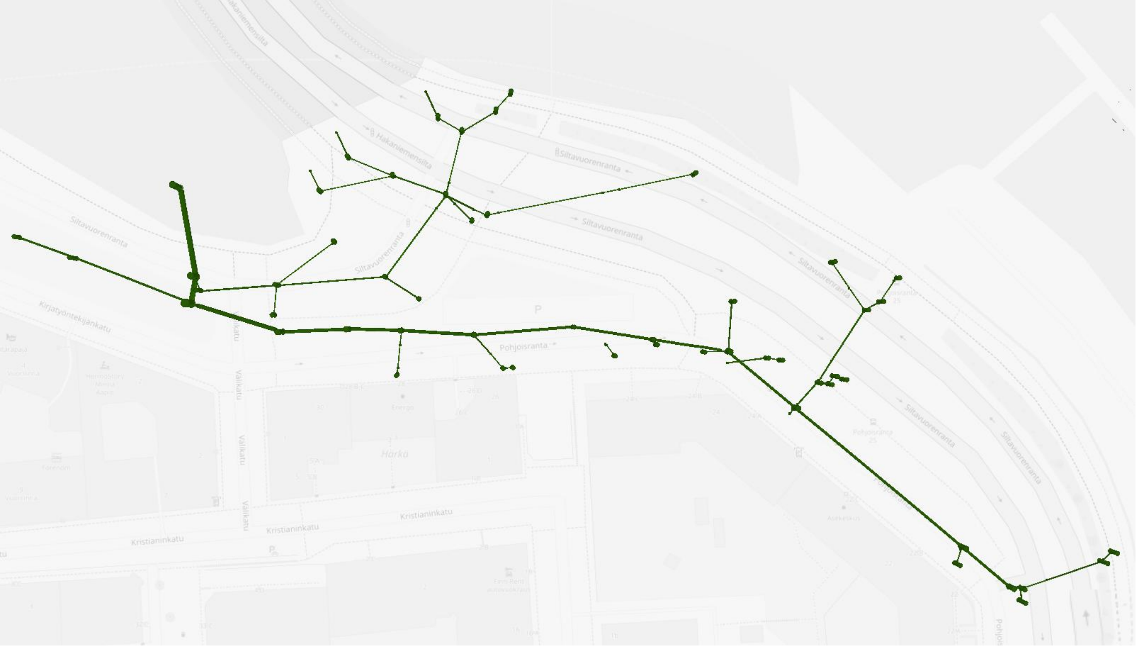


Figure 89. Stormwater as-built model used for AI examination

The anomaly detection phase focused on the LandXML version of the model. Metadata properties were assessed in terms of suitability, and after a thorough evaluation the *slope* parameter of the pipe elements was chosen as primary focus. A key factor that influenced this choice was that the slope relates simultaneously to geometry and metadata, offering a concise yet comprehensive demonstration of how anomalies can be detected across multiple parameter categories.

In addition, as-built modelling depends on land surveying, which may be affected by measurement errors, including false elevation values. Therefore, identifying abnormally steep or flat slopes can indicate potential measurement inaccuracies possibly revealing bigger issues, such as equipment malfunction of miscalibration.

For predictive insight generation, the IFC version of the model was used. Generating forecasts involves machine learning and requires a sufficiently large database for model training. Moreover, a reliable database enables validation through sanity check. These considerations led to the selection of environmental impact data for this part of the study, specifically, the CO_2 emission. This parameter is assigned to each element type and stored in the standard Pset_EnvironmentalImpactIndicators dataset. The emission values were extracted from the database described in chapter 2.2.3.3, which allowed model training on sufficient and reliable data and provided reference values for sanity checks.

4.4.2 Research Framework

4.4.2.1 Anomaly Detection

The anomaly detection process began with extracting slope values from all pipe elements in the model using xml.etree.ElementTree Python utility. Other widely used Python libraries, such as pandas and matplotlib, were also employed for data handling and visualisation.

The anomaly detection algorithm itself was based on the Isolation Forest method. The algorithm constructs multiple tree-like structures that progressively isolate individual data points by randomly splitting the dataset at each node. This randomness is crucial in this method, as it prevents data from fitting into a pattern. Consequently, the values that stand out of the majority are isolated in fewer steps. [83.] In Python, the method was implemented using the scikit-learn library.

The results are presented in figure 90. One crucial manually defined parameter for the Isolation Forest algorithm is the contamination rate. It specifies expected fraction of anomalies in the dataset. Results were generated for contamination rates of 1%, 5% and 10%. Since land survey data typically contains 1-5% errors, the 1% and 5% findings were considered particularly relevant. Figure 91 shows these anomalies projected back onto the as-built model with colour-coded highlights.

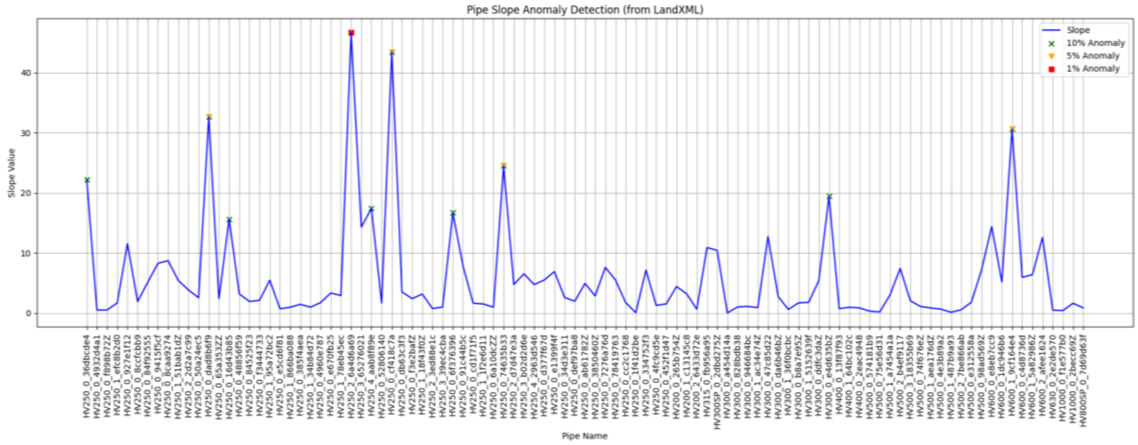


Figure 90. Slope anomalies detected by Isolation Forest

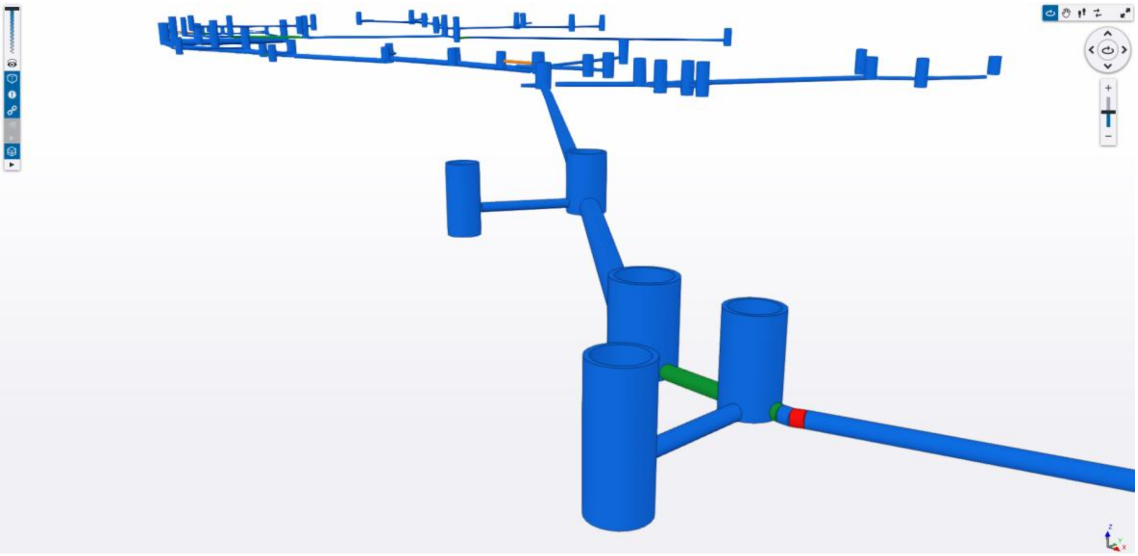


Figure 91. Detected anomalies highlighted in the original model shown in Trimble Connect

The above-presented analysis detected anomalies across the entire model. However, for more precise results, anomalies can also be inspected at the pipeline level, which refers to analysing a single pipeline composed of multiple pipe segments as an individual asset. Since a pipeline dataset is considerably smaller, the Isolation Forest method is not always necessary. A more suitable alternative in this case is the Local Outlier Factor (LOF), which relies on the nearest-neighbour distances. [84.]

An example of LOF applied on a single pipeline is illustrated in figure 92. The method identified an apparent winding in the pipeline route, which could possibly indicate a measurement error.

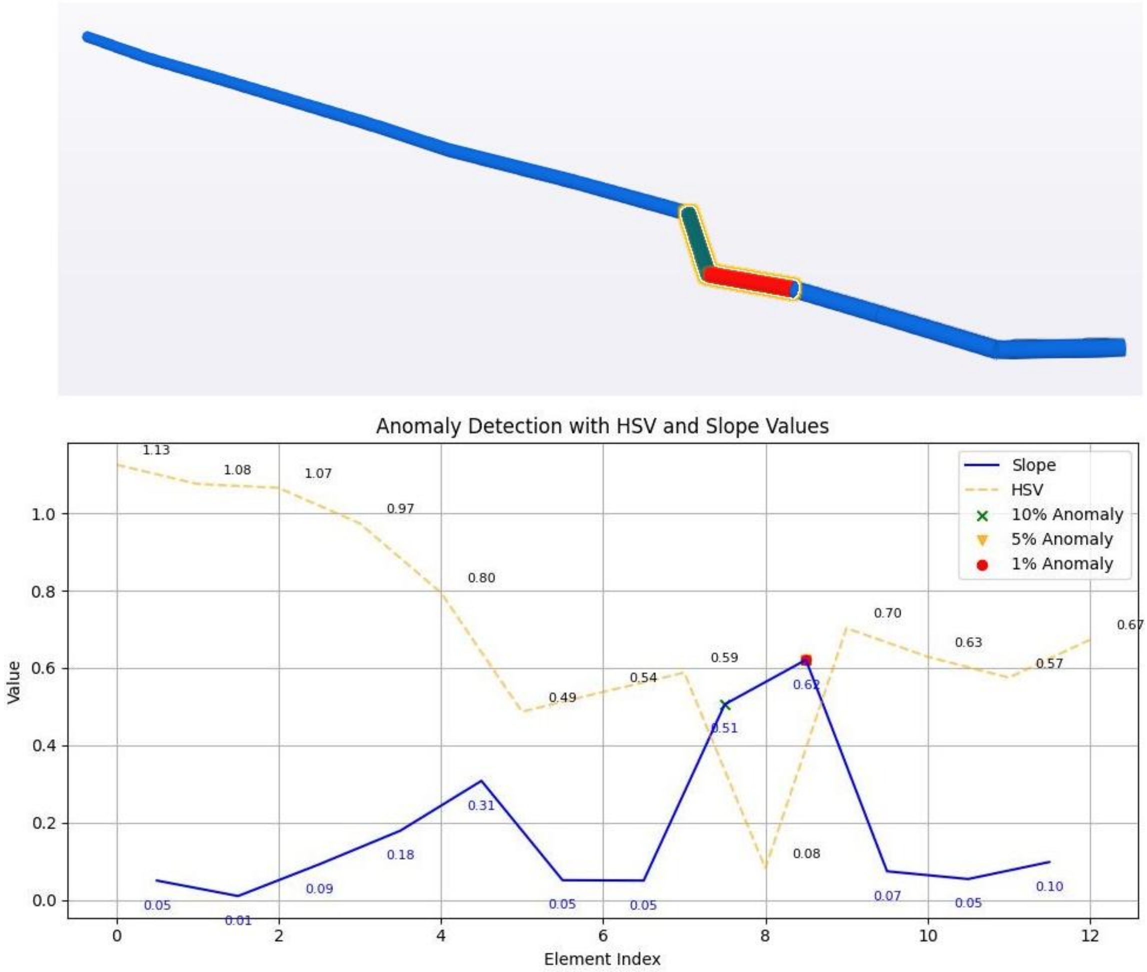


Figure 92. Local Outlier method applied on a single pipeline

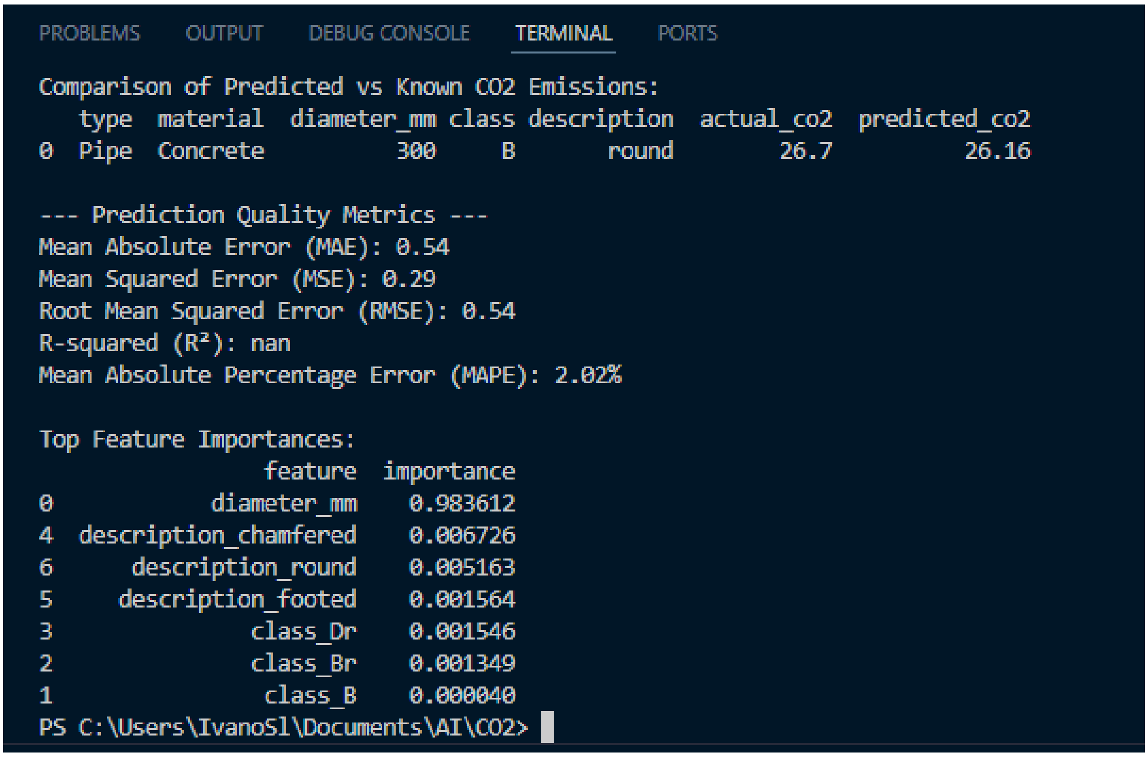
4.4.2.1 Predictive Insight Generation

The prediction analysis began with extracting all pipe types present in the IFC model, along with their CO_2 equivalent values from associated metadata sets. This task utilised the *IfcOpenShell* library, supported by *pandas*, *matplotlib, NumPy,* and *math* Python packages.

Since all pipe types initially contained correct emission values, one type's value was manually removed for testing purposes. The 300mm concrete pipe was chosen due to its common use in stormwater systems and its well-confirmed emission value, which made it perfectly suitable for prediction sanity checks and absolute error calculations.

The predictive algorithm combined two machine learning elements: the Random Forest Regressor and the One Hot Encoder. Random Forest regression uses ensembles of decision trees to generate predictions, and, similarly to Isolation Forest, it applies randomness principle to avoid overfitting. [85.] One Hot Encoder, in turn, converts property variables into a binary format, suitable for training machine learning models. [86.] Both utilities were imported from the scikit-learn library.

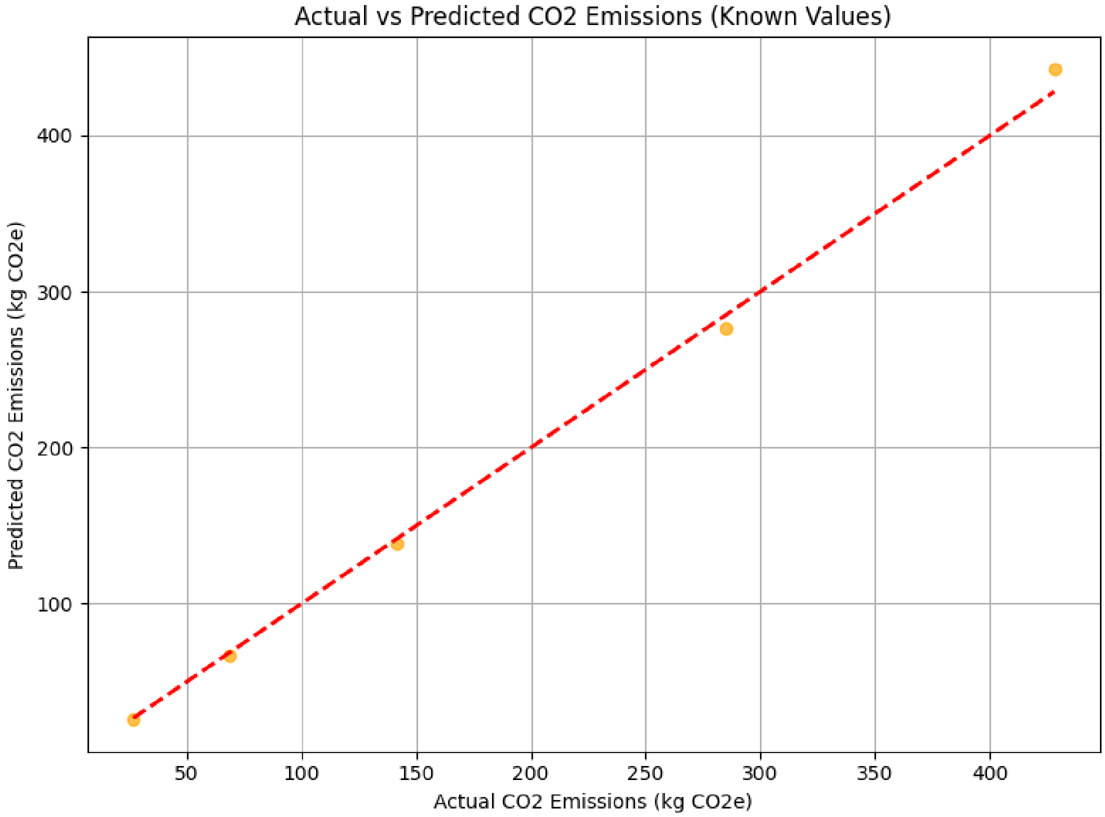
For model training, emission database entities were sorted by Type, Material, Size, Class, Description and CO2 equivalent. As illustrated in figure 93, the predicted result for the 300mm concrete pipe emission equivalent differed from the actual value by -0.54, resulting in absolute error of 2.02%. In other words, the model achieved an accuracy of approximately 98%.

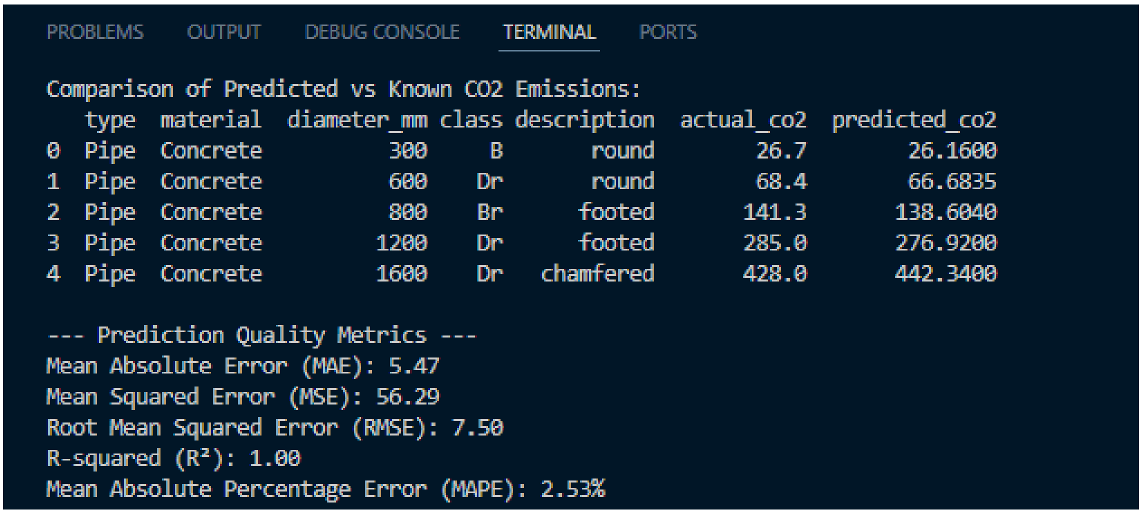


```
Comparison of Predicted vs Known CO2 Emissions:
   type  material  diameter_mm class description  actual_co2  predicted_co2
0  Pipe  Concrete          300     B       round        26.7          26.16

--- Prediction Quality Metrics ---
Mean Absolute Error (MAE): 0.54
Mean Squared Error (MSE): 0.29
Root Mean Squared Error (RMSE): 0.54
R-squared (R²): nan
Mean Absolute Percentage Error (MAPE): 2.02%

Top Feature Importances:
                 feature  importance
0            diameter_mm    0.983612
4  description_chamfered    0.006726
6      description_round    0.005163
5     description_footed    0.001564
3               class_Dr    0.001546
2               class_Br    0.001349
1                class_B    0.000040
PS C:\Users\IvanoSl\Documents\AI\CO2> 
```

Figure 93. Random Forest Regressor prediction results, featuring One Hot Encoder

For additional validation, a more challenging prediction test was conducted. CO_2 equivalent values were removed from five object types simultaneously: 300, 600, 800,1200 and 1600mm concrete pipes. As shown in figure 94, the absolute error value increased slightly by 0.51%, reducing the overall model accuracy to approximately 97.5%.





```
Comparison of Predicted vs Known CO2 Emissions:
   type  material  diameter_mm class description  actual_co2  predicted_co2
0  Pipe  Concrete          300     B       round        26.7        26.1600
1  Pipe  Concrete          600    Dr       round        68.4        66.6835
2  Pipe  Concrete          800    Br      footed       141.3       138.6040
3  Pipe  Concrete         1200    Dr      footed       285.0       276.9200
4  Pipe  Concrete         1600    Dr   chamfered       428.0       442.3400

--- Prediction Quality Metrics ---
Mean Absolute Error (MAE): 5.47
Mean Squared Error (MSE): 56.29
Root Mean Squared Error (RMSE): 7.50
R-squared (R²): 1.00
Mean Absolute Percentage Error (MAPE): 2.53%
```

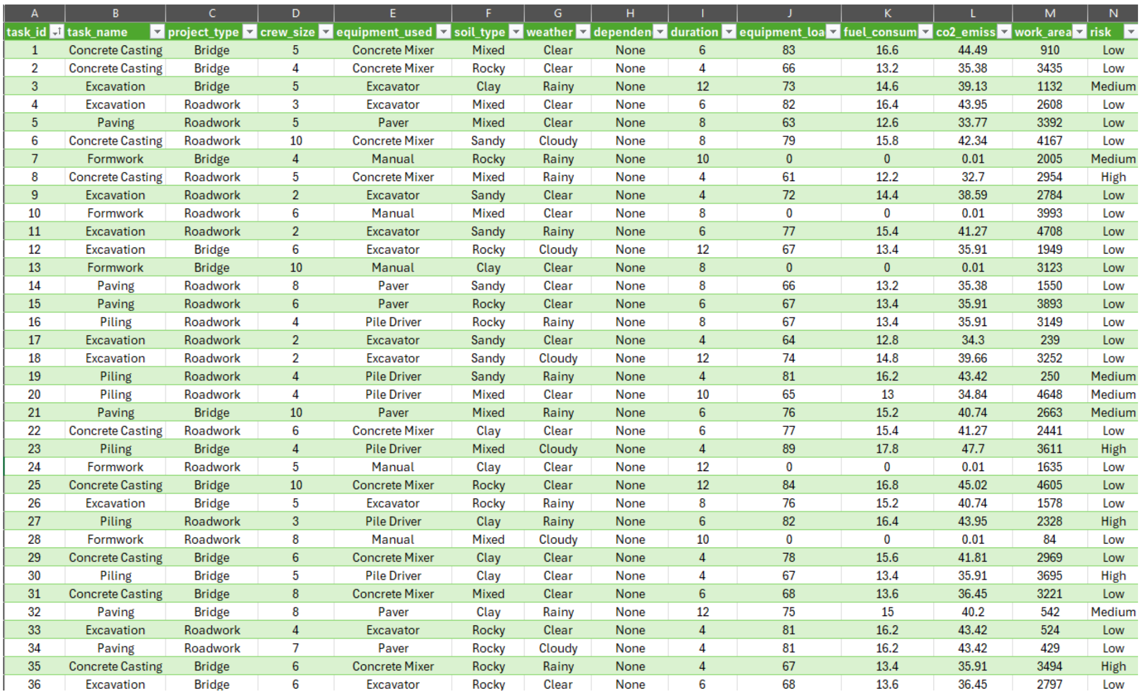
Figure 94. Random Forest Regressor sanity check results, featuring One Hot Encoder

4.4.3 Overview of AI suitability Examination

This chapter concludes the final experimental phase of the research. The study confirmed the feasibility of integrating BIM models with AI algorithms, with open data formats enabling seamless extraction of both geometric and metadata values for clustering and analysis.

The anomaly detection and value prediction approaches demonstrated in this chapter have clear potential to be incorporated into delivery workflows, enhancing schema validation and quality control procedures. Although, for simplicity, the algorithms were prototyped in Python, the same techniques can be implemented using JS modules, thereby opening additional integration opportunities. For example, the digital twin application described in chapter 4.3 could be extended to incorporate AI functionalities: the Isolation Forest could detect anomalies within the model or identify gaps or peaks in sensor measurements history, while Random Forest Regression could be applied to fill in the missing metadata or to forecast energy consumption.

The general principle of machine learning states that the more data is available, the more accurate predictive insights become. To emphasise the broader applicability of AI, three additional machine learning use cases relevant to infrastructure construction are included here, although they lay outside the scope of this thesis.



A	B	C	D	E	F	G	H	I	J	K	L	M	N
task_id	task_name	project_type	crew_size	equipment_used	soil_type	weather	dependen	duration	equipment_loa	fuel_consum	co2_emiss	work_area	risk
1	Concrete Casting	Bridge	5	Concrete Mixer	Mixed	Clear	None	6	83	16.6	44.49	910	Low
2	Concrete Casting	Bridge	4	Concrete Mixer	Rocky	Clear	None	4	66	13.2	35.38	3435	Low
3	Excavation	Bridge	5	Excavator	Clay	Rainy	None	12	73	14.6	39.13	1132	Medium
4	Excavation	Roadwork	3	Excavator	Mixed	Clear	None	6	82	16.4	43.95	2608	Low
5	Paving	Roadwork	5	Paver	Mixed	Clear	None	8	63	12.6	33.77	3392	Low
6	Concrete Casting	Roadwork	10	Concrete Mixer	Sandy	Cloudy	None	8	79	15.8	42.34	4167	Low
7	Formwork	Bridge	4	Manual	Rocky	Rainy	None	10	0	0	0.01	2005	Medium
8	Concrete Casting	Roadwork	5	Concrete Mixer	Mixed	Rainy	None	4	61	12.2	32.7	2954	High
9	Excavation	Roadwork	2	Excavator	Sandy	Clear	None	4	72	14.4	38.59	2784	Low
10	Formwork	Roadwork	6	Manual	Mixed	Clear	None	8	0	0	0.01	3993	Low
11	Excavation	Roadwork	2	Excavator	Sandy	Rainy	None	6	77	15.4	41.27	4708	Low
12	Excavation	Bridge	6	Excavator	Rocky	Cloudy	None	12	67	13.4	35.91	1949	Low
13	Formwork	Bridge	10	Manual	Clay	Clear	None	8	0	0	0.01	3123	Low
14	Paving	Roadwork	8	Paver	Sandy	Clear	None	8	66	13.2	35.38	1550	Low
15	Paving	Roadwork	6	Paver	Rocky	Clear	None	6	67	13.4	35.91	3893	Low
16	Piling	Roadwork	4	Pile Driver	Rocky	Rainy	None	8	67	13.4	35.91	3149	Low
17	Excavation	Roadwork	2	Excavator	Sandy	Clear	None	4	64	12.8	34.3	239	Low
18	Excavation	Roadwork	2	Excavator	Sandy	Cloudy	None	12	74	14.8	39.66	3252	Low
19	Piling	Roadwork	4	Pile Driver	Sandy	Rainy	None	4	81	16.2	43.42	250	Medium
20	Piling	Roadwork	4	Pile Driver	Mixed	Clear	None	10	65	13	34.84	4648	Medium
21	Paving	Bridge	10	Paver	Mixed	Rainy	None	6	76	15.2	40.74	2663	Medium
22	Concrete Casting	Roadwork	6	Concrete Mixer	Clay	Clear	None	6	77	15.4	41.27	2441	Low
23	Piling	Bridge	4	Pile Driver	Mixed	Cloudy	None	4	89	17.8	47.7	3611	High
24	Formwork	Roadwork	5	Manual	Clay	Clear	None	12	0	0	0.01	1635	Low
25	Concrete Casting	Bridge	10	Concrete Mixer	Rocky	Clear	None	12	84	16.8	45.02	4605	Low
26	Excavation	Bridge	5	Excavator	Rocky	Rainy	None	8	76	15.2	40.74	1578	Low
27	Piling	Roadwork	3	Pile Driver	Clay	Rainy	None	6	82	16.4	43.95	2328	High
28	Formwork	Roadwork	8	Manual	Mixed	Cloudy	None	10	0	0	0.01	84	Low
29	Concrete Casting	Bridge	6	Concrete Mixer	Clay	Clear	None	4	78	15.6	41.81	2969	Low
30	Piling	Bridge	5	Pile Driver	Clay	Clear	None	4	67	13.4	35.91	3695	High
31	Concrete Casting	Bridge	8	Concrete Mixer	Mixed	Clear	None	6	68	13.6	36.45	3221	Low
32	Paving	Bridge	8	Paver	Clay	Rainy	None	12	75	15	40.2	542	Medium
33	Excavation	Roadwork	4	Excavator	Rocky	Clear	None	4	81	16.2	43.42	524	Low
34	Paving	Roadwork	7	Paver	Rocky	Cloudy	None	4	81	16.2	43.42	429	Low
35	Concrete Casting	Bridge	6	Concrete Mixer	Rocky	Rainy	None	4	67	13.4	35.91	3494	High
36	Excavation	Bridge	6	Excavator	Rocky	Clear	None	6	68	13.6	36.45	2797	Low

Figure 95. Construction operations history log example

The first use case features predictive modelling based on historical construction data. Historical logs may contain records on operations such as piling or excavation, with details on used tools, work area, weather and fuel consumption, as illustrated in figure 95. A model trained on this data could predict future values such as required crew size, work duration, and fuel needs for similar excavation tasks. Such predictions can assist budget planning and scheduling, taking the workflows further towards automation. An example prediction output, for four piling scenarios is presented in figure 96.

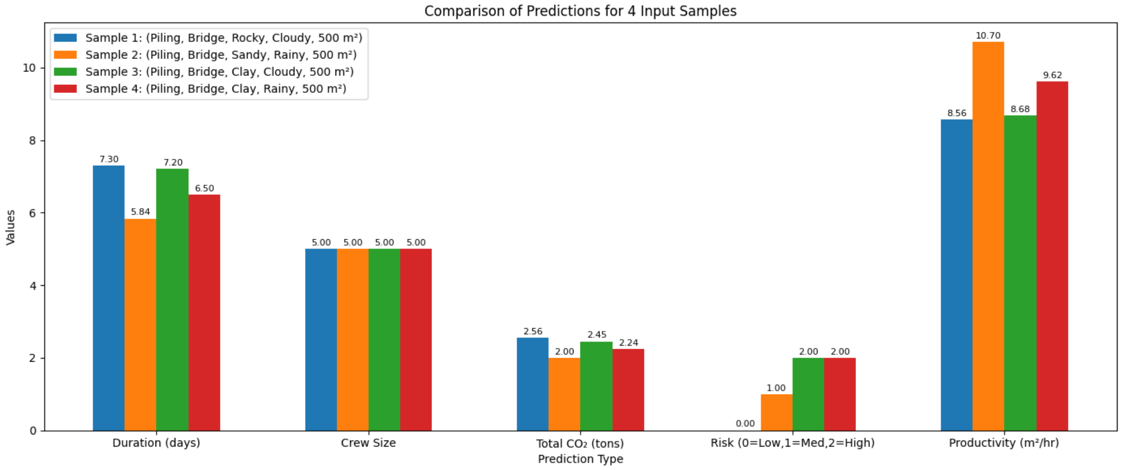
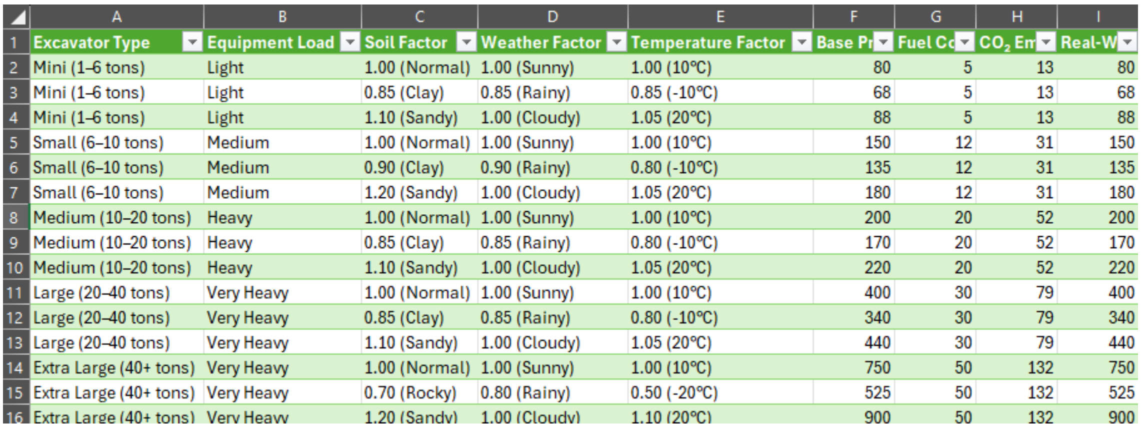


Figure 96. Example prediction for 4 piling setups

The second use case demonstrates process optimisation using static operational data. Example training data include vehicle performance statistics under various soil and weather conditions, as illustrated in figure 97.



	A	B	C	D	E	F	G	H	I
1	Excavator Type	Equipment Load	Soil Factor	Weather Factor	Temperature Factor	Base Pr	Fuel Co	CO$_2$ Em	Real-W
2	Mini (1–6 tons)	Light	1.00 (Normal)	1.00 (Sunny)	1.00 (10°C)	80	5	13	80
3	Mini (1–6 tons)	Light	0.85 (Clay)	0.85 (Rainy)	0.85 (-10°C)	68	5	13	68
4	Mini (1–6 tons)	Light	1.10 (Sandy)	1.00 (Cloudy)	1.05 (20°C)	88	5	13	88
5	Small (6–10 tons)	Medium	1.00 (Normal)	1.00 (Sunny)	1.00 (10°C)	150	12	31	150
6	Small (6–10 tons)	Medium	0.90 (Clay)	0.90 (Rainy)	0.80 (-10°C)	135	12	31	135
7	Small (6–10 tons)	Medium	1.20 (Sandy)	1.00 (Cloudy)	1.05 (20°C)	180	12	31	180
8	Medium (10–20 tons)	Heavy	1.00 (Normal)	1.00 (Sunny)	1.00 (10°C)	200	20	52	200
9	Medium (10–20 tons)	Heavy	0.85 (Clay)	0.85 (Rainy)	0.80 (-10°C)	170	20	52	170
10	Medium (10–20 tons)	Heavy	1.10 (Sandy)	1.00 (Cloudy)	1.05 (20°C)	220	20	52	220
11	Large (20–40 tons)	Very Heavy	1.00 (Normal)	1.00 (Sunny)	1.00 (10°C)	400	30	79	400
12	Large (20–40 tons)	Very Heavy	0.85 (Clay)	0.85 (Rainy)	0.80 (-10°C)	340	30	79	340
13	Large (20–40 tons)	Very Heavy	1.10 (Sandy)	1.00 (Cloudy)	1.05 (20°C)	440	30	79	440
14	Extra Large (40+ tons)	Very Heavy	1.00 (Normal)	1.00 (Sunny)	1.00 (10°C)	750	50	132	750
15	Extra Large (40+ tons)	Very Heavy	0.70 (Rocky)	0.80 (Rainy)	0.50 (-20°C)	525	50	132	525
16	Extra Large (40+ tons)	Very Heavy	1.20 (Sandy)	1.00 (Cloudy)	1.10 (20°C)	900	50	132	900

Figure 97. Vehicle stats example

The optimisation model then evaluates which operation setup is the most efficient in terms of schedule, workforce allocation, fuel consumption, and emissions. An illustrative optimisation output for 2000 m³ sandy soil excavation on a rainy 7°C Day, is provided in figure 98.

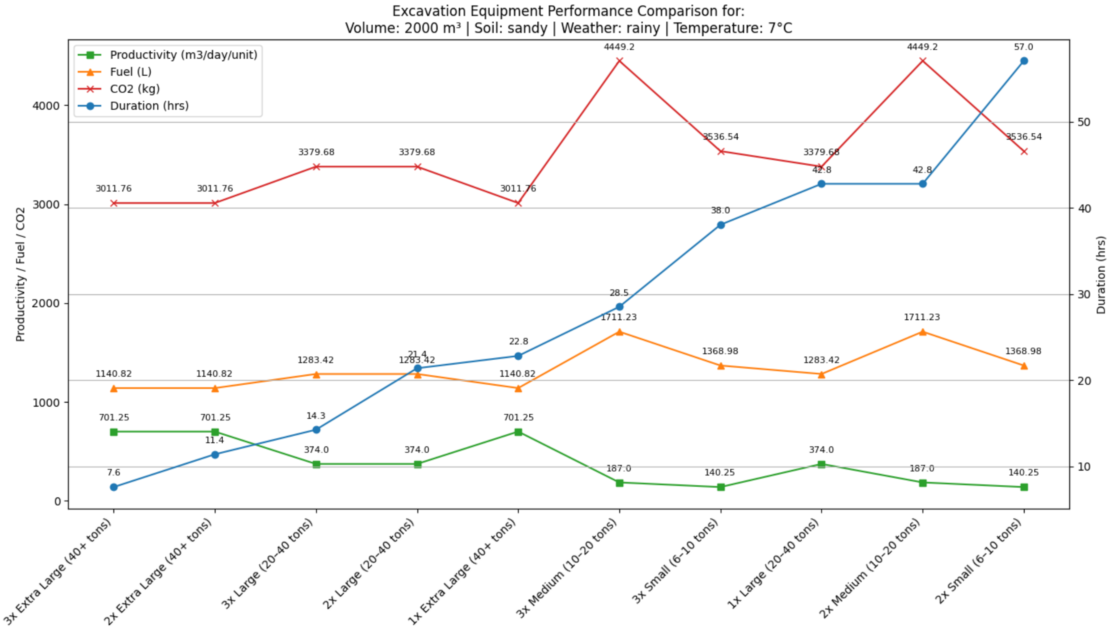
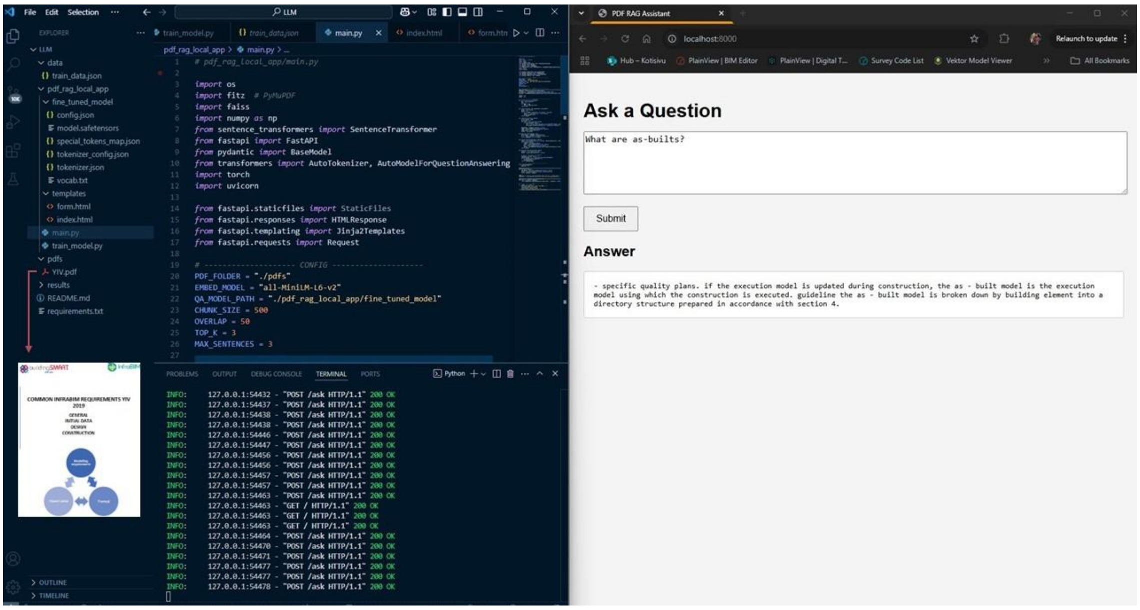


Figure 98. Example optimisation chart for 2000m^3 sandy soil excavation performed in rainy *7°C* conditions

The third example explores language models applications in closed data environments, specifically through a Retrieval-Augmented Generation (RAG) approach. RAG is a natural language processing engine that retrieves, analyses, and restructures information from a given knowledge base, in response to a user query. A suitable knowledge base may consist of both structured documents and unstructured text fragments. [87.] Figure 99 illustrates the implementation of Common Infra BIM requirements, a document described in chapter 2.2.1.1, as a knowledge base. The PDF document was used without modification, and the RAG algorithm provided context-specific answers, such as retrieving all definitions of as-Built Models in response to a prompt query.



```python
# pdf_rag_local_app/main.py

import os
import fitz  # PyMuPDF
import faiss
import numpy as np
from sentence_transformers import SentenceTransformer
from fastapi import FastAPI
from pydantic import BaseModel
from transformers import AutoTokenizer, AutoModelForQuestionAnswering
import torch
import uvicorn

from fastapi.staticfiles import StaticFiles
from fastapi.responses import HTMLResponse
from fastapi.templating import Jinja2Templates
from fastapi.requests import Request

# -------------------- CONFIG --------------------
PDF_FOLDER = "./pdfs"
EMBED_MODEL = "all-MiniLM-L6-v2"
QA_MODEL_PATH = "./pdf_rag_local_app/fine_tuned_model"
CHUNK_SIZE = 500
OVERLAP = 50
TOP_K = 3
MAX_SENTENCES = 3
```

```
INFO:     127.0.0.1:54432 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54437 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54438 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54438 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54446 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54447 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54456 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54456 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54457 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54457 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54463 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54463 - "GET / HTTP/1.1" 200 OK
INFO:     127.0.0.1:54463 - "GET / HTTP/1.1" 200 OK
INFO:     127.0.0.1:54463 - "GET / HTTP/1.1" 200 OK
INFO:     127.0.0.1:54464 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54470 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54471 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54477 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54477 - "POST /ask HTTP/1.1" 200 OK
INFO:     127.0.0.1:54478 - "POST /ask HTTP/1.1" 200 OK
```

Ask a Question

What are as-builts?

Answer

- specific quality plans. if the execution model is updated during construction, the as - built model is the execution model using which the construction is executed. guideline the as - built model is broken down by building element into a directory structure prepared in accordance with section 4.

Figure 99. Example use of Retrieval Augmented Generation

Although this approach requires further model training in order to evolve into a fully functioning BIM advisor, it demonstrates the potential for creating domain-specific AI assistants. With larger knowledge bases and incorporation of human expertise, such systems could largely contribute to design or quality assurance process.

In conclusion, this research demonstrates the considerable potential of AI applications in infrastructure. The predominance of open-source tools provides a strong foundation for the development of next-generation solutions, such as common data environments that integrate sensor inputs, evaluate its own content independently. Moreover, these solutions could extend their capabilities to legal advice, design consultancy, and AI-assisted project handover.

5 Discussion and Conclusions

This chapter presents the final conclusions of the thesis study. For clarity, the discussion follows the research perspectives and questions outlined in chapter 1. Each subchapter contains both conclusions and a discussion.

5.1 Initial Data Perspective

As mentioned before, the availability of reliable initial data was one of the primary triggers for as-built modelling development. The internal cohesion and overall integrity of the as-built deliverables influence subsequent modelling activities at multiple levels. Therefore, the findings and observations derived from this research perspective are expected to provide applicable insights for forthcoming infrastructure projects, specifically for the light rail undertakings.

As summarised in chapter 4.1 model readability index was significantly higher for viewing platforms compared to modelling applications, and neither format was compatible with all presented solutions. This leads to the conclusion that the as-built models produced for the Crown Bridges project are indeed suitable for use as initial data, although it comes with a set of conditions. The primary delivery format, LandXML, can be successfully imported into the modelling, viewing and surveying platforms most relevant for the infrastructure domain, enabling its use as a reference model. However, data exchange with building construction projects may require conversion into IFC format.

Metadata accessibility issues in design platforms can be addressed through double screen approach illustrated in chapter 4.2.3. Given that most projects rely on a combination of modelling and viewing tools, the application of these models as initial data is entirely feasible. With respect to format comparison, LandXML remains prevailing standard in Finnish infrastructure construction, making on-demand conversion to IFC a more efficient strategy than maintaining datasets in parallel formats. [88.]

Finally, the use of external URLs to link documentation to the as-built model presents a promising method for enabling model-based handover. However, due to its novelty, this method requires training on how to locate, use, and maintain these links. Future projects may need to develop more comprehensive protocols for managing external links.

The minimum acceptable level of information for as-built models encompasses basic volumetric properties and project-specific metadata, typically defined by an asset owner. Basic volumetric properties may include circular hollow-section profile, lengths and slopes for utility pipelines, and three-dimensional hull models for shallow foundations and cables. Project-specific metadata, on the other hand, can range from simple material names to extensive property sets, but in practice, more metadata is generally beneficial. [88.]

Although publications and regulatory provisions referenced in this study do not specify detailed metadata requirements, repeated mention of the bill of materials highlights the importance of material information. Therefore, it can be concluded that each entity should at least contain information on used materials.

Another critical topic is geometry complexity, over-detailed models, and excessive clustering. While excessive metadata rarely affects software performance, overly complex geometries or triangulation may make model usage extremely difficult or even impossible, potentially leading to platform crashes and requiring additional model simplification. Excessive clustering — combining multiple models into a single large file— may simplify transfers but results in an immense size model that requires dismantling. A practical approach is to maintain one model per discipline and per geographic section, such as a road segment between crossroads. [88.] This sectioning principle was applied in the Crown Bridges Light Rail project.

5.2 Operation and Maintenance Perspective

The application of as-built models in asset management was one of the explicit objectives of the Crown Bridges project. Current observations reveal strong interest coming from maintenance managers, complimented by desire to learn and adopt new practices. This subchapter concerns the overall suitability of as-built models for integration into asset management frameworks and provides insights to facilitate this adoption. Although digital twin readiness was originally evaluated under a different category, the related conclusions are presented here for consistency.

A primary criterion for operation and maintenance suitability was model readability within relevant software platforms. The study shows that the Crown Bridges as-built models can be compatible with such platforms if converted into IFC format. The limited number of available platforms narrowed the examination scope down to one scenario but also reflected the current state of the industry. Within these constraints, the dataset is confirmed suitable.

Another related consideration is digital twin readiness, as efficient asset management relies on the capacity to reflect the asset's current state. The Crown Bridges as-built models conform to the digital twin definitions given in the chapter 2.1.3: they are an information construct of the physical twin, they are annotated 3D models with correct dimensions, they include information on used materials, processes, they are able to incorporate information on service records and, most importantly, can be linked to the sensor data streams. These capabilities apply to both of the examined data exchange formats. Furthermore, the openness of the data formats enables the inclusion of maintenance-specific metadata. Examples include maintenance classification for streets, areas and green spaces, relevant codes and quantities. [89.]

The as-built models examined in this thesis qualify as digital twins according to the Crown Bridges project-specific definition given in chapter 2.1.3, but whether they qualify as digital twins on a general scale is rather debatable, as definitions of the concept vary widely. It is therefore most accurate to state that the models are applicable within digital twin frameworks, but their classification as digital twins should be determined locally, based on the definitions adopted within a specific workflow. At the managerial level, training should emphasise the strategic benefits of model-based maintenance. The limited availability of suitable solutions for model-based maintenance and operation platforms highlights the immaturity of the existing framework and the need for tool development and professional training, at both operational and managerial levels.

Training perspectives for operational staff should cover adoption and integration of delivered as-built models, model maintenance—which, in turn, may require advanced information modelling skills—and preparation of handover datasets. Such datasets may be highly task-specific, for instance, a BIM model updated to as-maintained status and accompanied by change logs and maintenance records. At the managerial level, training should emphasise the strategic benefits and expected value of model-based maintenance.

Although detailed infrastructure-focused reports on BIM implementation in asset management are still rare, evidence from building facility management suggests promising outcomes: adoption of BIM-based maintenance workflow reduced time spent on locating elements by 83%, shortened risk assessment by 50% and lowered overall annual costs by 5%. [90.]

5.3 Regulatory Compliance Perspective

While legal compliance was not an explicit research question, and is not yet a mandatory requirement, it is expected to become increasingly relevant in the future. Concluding this research perspective requires reviewing the definitions of as-built models given in referenced provisions. Reviewing reveals recurring references to geometry, location, deviations from the original designs, and basic property data. These specifications are rather superficial, obviously leaving further detail to project-level governance. Machine readability is another recurring requirement, and Finnish Built Environment Information System specifically identifies IFC as the required format.

Based on existing definitions and obligations, the Crown Bridges as-built models conform as far as is currently possible. Open format policy, adopted in the project, ensures their scalability for future regulatory updates.

It is advisable to monitor ongoing standardisation and legislative developments closely, as the available evidence leaves an impression that additional provisions are underway. Clarifying clauses in legislation, such as those authorising supplementary regulations given by the Ministry of Environment, solidify this implication. The closest foreseeable update may be the Act of the Built Environment Information System extending its jurisdiction to infrastructure assets, resulting in mandatory the as-built data submission into the system. In such a case, the dataset compliance verification would become an integral stage in the asset lifecycle, affecting delivery, maintenance and redevelopment phases.

5.4 Auxiliary Applications Perspective

This category initially focused on structure gauge analysis but was later extended to sensor data streaming and machine learning applications, reflecting the growing importance of IoT and AI in civil construction. As expected, the models performed exceptionally well in both structure gauge checks—successfully detecting envelope penetrations—and in AI integration tests, which produced reliable predictive results. Moreover, the structure gauge examination success proved the BIM model-driven method being more efficient than the point cloud-based. The success of these applications is largely attributable to the openness of the data formats, their readability, and their clear internal structure. The results described in chapters 4.3 and 4.4 demonstrate the current potential of these methods and suggest promising future perspectives.

Another significant observation is that, due to the availability of open-source tools, development does not need to rely on large software vendors. However, developing highly functional solutions in-house may require the expertise of dedicated professionals.

Future research could extend structure gauge analysis into animated simulations. Existing libraries already enable the development of 3D viewport in which detailed rolling stock models can be moved along alignments, replicating realistic turning angles of a vehicle. Comparable solutions are already employed by streetcar manufacturers as part of their internal workflows, therefore, similar tools integration into common data environments may benefit from collaboration with manufacturers. Machine learning algorithms, in turn, could be deployed as intelligent validators, capable of detecting missing metadata, retrieving correct values from databases or applying predictive methods when such values are unavailable.

Across all four research perspectives, clear interconnections can be identified. Initial data and maintenance belong to consequent phases of the asset lifecycle, and both derive from as-built deliverables. Legal aspect, in turn, affect them both, and the inclusion of as-built models in legislation highlights this interdependence. In return, regulatory development is often driven by advances in construction practices, creating a feedback loop. The same applies to artificial intelligence: it can be used to validate compliance with laws and standards, while those same regulations set new targets for AI tools development.

5.5 Contribution Statement

While most existing studies on BIM focus on building sector, this research extends the discussion further into the civil construction domain and contributes to both theoretical and practical aspects of as-built modelling. From a theory perspective, it provides one of the first systematic evaluations of infrastructure as-built modelling frameworks, addressing the asset lifecycle, industry practices, and regulatory environments. On the practical level, it demonstrates the immediate applicability of as-built models produced for a major civil construction project. By documenting format compatibility, metadata accessibility, workflow bottlenecks, and intersection with regulatory provisions, this study presents an applicable knowledge base for engineers, project owners, designers, land surveyors and facility managers.

Finally, this research bridges the gap between asset lifecycle phases with well-established model-based practices and those yet to adopt them. It shows that current as-built deliverables are already capable of addressing many practical challenges, while identifying areas where further development and standardisation are needed.

References

1 Crown Bridges Light Rail. Online. City of Helsinki. <https://kruunusillat.fi/en/home/>. 2021. Accessed 30 Aug 2025.

2 Gröndahl Miikka. City of Helsinki. Interviewed 7 May 2025.

3 Bolpagni, Michele; Gavina, Riccardo & Ribeiro, Diogo. Industry 4.0 for the Built Environment: Methodologies, Technologies and Skills. 2022. Cham: Springer.

4 BIM Dictionary. Online. BIMe Initiative. <https://bimdictionary.com/>. Accessed 20 Feb 2025.

5 Eastman, Chuck; Teicholz, Paul; Sacks, Rafael & Liston, Kathleen. BIM Handbook: A Guide to Building Information Modelling for Owners, Managers, Designers, Engineers and Contractors. 2008. Hoboken (NJ): John Wiley & Sons.

6 Ellis, George. The evolution of BIM dimensions: 3D, 4D, 5D & beyond [Internet]. Online. Autodesk. <https://www.autodesk.com/blogs/construction/bim-dimensions/>. 2024. Accessed 20 Feb 2025.

7 Common InfraBIM Requirements. Online. buildingSMART Finland. <https://drive.buildingsmart.fi/s/c5jX78gHo9mAQab>. 2019. Accessed 20 Feb 2025.

8 As-built model — definition. Online. Law Insider. <https://www.lawinsider.com/dictionary/as-built-model>. Accessed 20 Feb 2025.

9 As-Built BIM [Internet]. Online. Reality IMT. <https://realityimt.com/industry-insights/as-built-bim/>. 2018. Accessed 20 Feb 2025.

10 Grieves, Michael William. Virtually intelligent product systems: digital and physical twins. In: Complex Systems Engineering: Theory and Practice. 2019. Reston (VA): American Institute of Aeronautics and Astronautics.

11 Grieves, Michael William. Origins of the Digital Twin Concept. Working paper. 2016. Melbourne (FL): Florida Institute of Technology / NASA.

12 Dirnfeld, Ronen. Digital Twins in Railways: State of the Art, Opportunities, and Guidelines. 2022. Linnaeus University, Faculty of Technology.

13 buildingSMART International. Online. <https://www.buildingsmart.org/>. 2025. Accessed 20 Feb 2025.

14 RYTV Project Program. Online. buildingSMART Finland. <https://www.buildingsmart.fi/en_GB/rytv>. 2025. Accessed 5 Aug 2025.

15 International Organization for Standardization. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling. Part 1: Concepts and principles (ISO 19650-1:2018). 2018. Geneva: ISO.

16 International Organization for Standardization. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling. Part 3: Operational phase of the assets (ISO 19650-3:2020). 2020. Geneva: ISO.

17 International Organization for Standardization. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling. Part 2: Delivery phase of the assets (ISO 19650-2:2018). 2018. Geneva: ISO.

18 International Organization for Standardization. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling. Part 4: Information exchange (ISO 19650-4:2022). 2022. Geneva: ISO.

19 buildingSMART Validation Service. Online. buildingSMART International. <https://validate.buildingsmart.org/>. Accessed 20 Feb 2025.

20 What is the ISO 16739 standard? Online. BibLus. <https://biblus.accasoftware.com/en/what-is-the-iso-16739-standard/>. 2022. Accessed 20 Feb 2025.

21 International Organization for Standardization. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema (ISO 16739-1:2024). 2024. Geneva: ISO.

22 The evolution of IFC: The path to IFC5. Online. buildingSMART International. <https://www.buildingsmart.es/2024/12/03/the-evolution-of-ifc-the-path-to-ifc5/>. 2024. Accessed 20 Aug 2025.

23 IFC – Industry Foundation Classes. Online. OSArch. <https://wiki.osarch.org/index.php/IFC_-_Industry_Foundation_Classes>. Accessed 20 Feb 2025.

24 A New Way of Working. Online. buildingSMART International. <https://education.buildingsmart.org/a-new-way-of-working/>. Accessed 20 Feb 2025.

25 Inframodel – Data Exchange Format. Online. buildingSMART Finland. <https://wiki.buildingsmart.fi/fi/04_Julkaisut_ja_Standardit/Inframodel>. Accessed 20 Feb 2025.

26 International Organization for Standardization. Information management using building information modelling (BIM) — Part 5: Security minded approach to information management (ISO 19650-5:2020). 2020. Geneva: ISO.

27 Data Security in BIM: Understanding its Aspects, Risks, and Best Practices. Online. Interscale Education. <https://interscaleedu.com/en/blog/bim/data-security-in-bim/>. 2024. Accessed 20 Feb 2025.

28 Tiihonen, Pasi. Yleiskirje. Online. Kuntaliitto. <https://www.kuntaliitto.fi/yleiskirjeet/2023/laki-rakennetun-ympariston-tietojarjestelmasta-voimaan-112024-rakentamisen-ja>. 2023. Accessed 20 Feb 2025.

29 Construction Act 751/2023. Online. Finland. <https://finlex.fi/fi/laki/alkup/2023/20230751>. 2023. Accessed 20 Feb 2025.

30 Act on the Built Environment Information System 431/2023. Online. Finland. <https://finlex.fi/fi/laki/alkup/2023/20230431>. 2023. Accessed 20 Feb 2025.

31 RYHTI – Built Environment Information System. About the system. Online. SYKE. <https://ryhti.syke.fi/en/about-the-system/>. 2025. Accessed 20 Feb 2025.

32 RYHTI – Built Environment Information System map-based service. Online. SYKE. <https://ryhti-palvelut.syke.fi/kartta>. 2025. Accessed 20 Feb 2025.

33 Emissions database for infrastructure construction. Online. CO2DATA. <https://co2data.fi/docs/Palvelukuvaus.pdf>. 2024. Accessed 20 Feb 2025.

34 Emission database for infrastructure construction. Online. Carbon-neutral Finland. <https://www.hiilineutraalisuomi.fi/en-US/Tools_and_services/National_emission_database_CO2data/Infrastructure_construction_database/Emission_database_for_infrastructure_con(66652)>. 2024. Accessed 20 Feb 2025.

35 Borrmann, André; König, Markus; Koch, Christian & Beetz, Jakob. Building Information Modelling – Technology Foundations and Industry Practice. 2018. Cham: Springer.

36 Which BIM software to use. Online. BIM Corner. <https://bimcorner.com/which-bim-software-to-use/>. 2020. Accessed 20 Feb 2025.

37 What’s new in Solibri Office. Online. Solibri. <https://www.solibri.com/articles/release-highlights-release-24-5-0>. 2024. Accessed 20 Feb 2025.

38 How does land surveying work? Online. National Land Survey of Finland. <https://www.maanmittauslaitos.fi/en/topical_issues/how-does-land-surveying-work>. 2022. Accessed 20 Feb 2025.

39 Total Station Surveying vs GNSS Surveying: Understanding the Differences. Online. Global GPS Systems. <https://globalgpssystems.com/total-station/total-station-surveying-vs-gnss-surveying-understanding-the-differences/>. Accessed 20 Feb 2025.

40 Photogrammetry Processing. Online. Propeller Aero. <https://www.propelleraero.com/survey-processing/>. 2025. Accessed 20 Feb 2025.

41 Ibrahimkhil, Mohammad Haroon; Shen, Xiaodong & Barati, Kamran. Enhanced Construction Progress Monitoring through Mobile Mapping and As Built Modeling. In: Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC). 2021. Dubuque (IA): IAARC.

42 Droneinfo. Online. <https://www.droneinfo.fi/en/registration-and-theoretical-examination>. 2025. Accessed 20 Feb 2025.

43 Shahmoradi, Javad; Roghanchi, Pejman & Hassanalian, Mostafa. Drones in underground mines: Challenges and applications. In: 2020 Gulf Southwest Section Conference. 2020. ASEE Conferences.

44 Wu, Yuhang; Li, Meng & Xue, Fuzhan. Towards fully automatic Scan to BIM: A prototype method integrating deep neural networks and architectonic grammar. In: Proceedings of the 2023 European Conference on Computing in Construction & the 40th International CIB W78 Conference. 2023. Luxembourg: European Council on Computing in Construction.

45 Madubuike, Okechukwu Chukwuemeka; Anumba, Chimay J. & Papadonikolaki, Eleni. Leveraging Digital Twins for Enhanced Construction Project Delivery. In: Proceedings of the 2024 European Conference on Computing in Construction (EC3). 2024. Luxembourg: European Council on Computing in Construction.

46 Trimble Connect. Online. Trimble. <https://www.trimble.com/en/products/trimble-connect>. 2025. Accessed 20 Feb 2025.

47 Moghtadernejad, Sajad; Lounis, Zoubir & Zhang, Zhishen. Bridge deterioration modeling considering climate change effects. In: Infrastructure Resilience and Sustainability in Built and Natural Environments. Boca Raton (FL): CRC Press / Taylor & Francis.

48 Infrakit. Online. <https://www.infrakit.com/products/>. 2025. Accessed 20 Feb 2025.

49 Why Connected Is Key with Common Data Environments. Online. Trimble. <https://constructible.trimble.com/construction-industry/why-connected-is-key-with-common-data-environments>. Accessed 20 Feb 2025.

50 Open Sensor Web. Online. <https://www.opensensorweb.de/en>. 2025. Accessed 20 Feb 2025.

51 Those Open docs. Online. That Open Company. <https://docs.thatopen.com/>. 2025. Accessed 20 Feb 2025.

52 Maps. Online. Mapbox. <https://www.mapbox.com/maps>. 2025. Accessed 20 Feb 2025.

53 IfcOpenShell documentation. Online. IfcOpenShell. <https://docs.ifcopenshell.org/introduction.html>. 2025. Accessed 20 Feb 2025.

54 BIM files: the main BIM formats for design. Online. BibLus. <https://biblus.accasoftware.com/en/bim-files-the-main-bim-formats-for-design/>. 2025. Accessed 20 Aug 2025.

55 LandXML. Home page. Online. <http://www.landxml.org/>. Accessed 20 Feb 2025.

56 LandXML files. Online. Autodesk Help. <https://help.autodesk.com/view/INFDS/ENU/?guid=GUID-76CAEA4D-6353-4350-ADBB-129B70017957>. 2025. Accessed 20 Feb 2025.

57 What is LandXML? Online. LandXML. <http://www.landxml.org/about.aspx>. Accessed 20 Feb 2025.

58 Mankidi, Ajay Babu. Elucidation of Traffic Congestion at Seematti Round about, Kottayam. 2014. Pathamuttom, Kerala: Saintgits College of Engineering.

59 Sellarès, Toni. Terrains and Triangulations. Online. University of Girona, Department of Computer Science & Mathematics. <https://imae.udg.edu/~sellares/ComGeo/TIN.pdf>. 2013. Accessed 30 Aug 2025.

60 LandXML 1.2 schema. Online. <http://landxml.org/schema/LandXML-1.2/documentation/>. 2008. Accessed 20 Feb 2025.

61 Industry Foundation Classes. Online. buildingSMART International. <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>. 2025. Accessed 20 Feb 2025.

62 3D formats overview: STEP. Online. CAD Exchanger. <https://cadexchanger.com/blog/3d-formats-overview-step/>. 2021. Accessed 20 Feb 2025.

63 IFC Specifications Database. Online. buildingSMART International. <https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/>. 2025. Accessed 20 Feb 2025.

64 All you need to know about IFC 4.3 for infrastructure. Online. BIM Corner. <https://bimcorner.com/ifc-4-3-for-infrastructure/>. 2022. Accessed 20 Feb 2025.

65 ArchiCAD Help. Online. Graphisoft. <https://help.graphisoft.com/AC/19/INT/AC19Help/07_Interoperability/07_Interoperability-68.htm>. 2015. Accessed 20 Feb 2025.

66 Relations I used to create IFC 4.3 for infrastructure. Online. BIM Corner. <https://bimcorner.com/ifcrelationship-in-infrastructure-ifc-4-3/>. 2022. Accessed 20 Feb 2025.

67 IFC4X3 schema documentation: Shape representation. Online. buildingSMART. <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcShapeRepresentation.htm>. 2024. Accessed 20 Feb 2025.

68 Hasan, Ahmed M. M.; Torky, Ahmed A. & Rashed, Yasser F. Geometrically accurate structural analysis models in BIM centered software. 2019. Automation in Construction.

69 Tarina GT-formaatista. Online. Geocenter. <https://www.geocenter.fi/blogi/joulutarina-gt-formaattista/>. 2013. Accessed 20 Feb 2025.

70 Lehikoinen, Anniina. buildingSMART Finland. Interviewed 8 Apr 2025.

71 Kang, Tae Wook. Object composite query method using IFC and LandXML based on BIM linkage model. 2017. Automation in Construction.

72 IFC 5: What it is and how to manage the new openBIM standard for data exchange in the AEC sector. Online. BibLus. <https://biblus.accasoftware.com/en/what-is-ifc-5/>. 2025. Accessed 20 Aug 2025.

73 What is Jokeri Light Rail? Online. Raide-Jokeri. <https://raidejokeri.info/en/jokeri-light-rail-from-itakeskus-to-keilaniemi/>. Accessed 20 Aug 2025.

74 Structure Gauge and Kinematic Envelope. Online. RailSystem. <https://railsystem.net/structure-gauge-and-kinematic-envelope/>. Accessed 20 Aug 2025.

75 Jokeri Light Rail structure gauge final report (Not available for public access). Sweco. 2023. Sweco.

76 Finnish Transport Infrastructure Agency (Väylävirasto). Track technical instructions (RATO). Part 2 Track Geometry. Online. Väylävirasto. <https://vayla.fi/palveluntuottajat/ohjeluettelo>. Accessed 20 Aug 2025.

77 Raide-Jokerin keilaus? Hoituu! Online. GEOTRIM. <https://geotrim.fi/yritys/blogi/raide-jokerin-keilaus-hoituu/>. Accessed 20 Aug 2025.

78 TrimBIM technology. Online. Trimble. <https://developer.tekla.com/documentation/trimbim-technology>. 2024. Accessed 20 Aug 2025.

79 Using 3D machine control on smaller construction sites – easy, fast, and cost-efficient. Online. Novatron. <https://novatron.fi/en/using-3d-machine-control-on-smaller-construction-sites/>. Accessed 20 Aug 2025.

80 Light Rail Design Guidelines. Online. City of Helsinki. <https://raitiotieohje.fi/tilavaatimukset-2/>. Accessed 20 Aug 2025.

81 Sensor Data Hub. Online. Sensoto. <https://sensoto.io/en/>. Accessed 20 Feb 2025.

82 The standard for IoT messaging. Online. MQTT. <https://mqtt.org/>. Accessed 20 Feb 2025.

83 Anomaly detection using Isolation Forest. Online. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/anomaly-detection-using-isolation-forest/>. 2025. Accessed 30 Aug 2025.

84 Local Outlier Factor. Online. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/local-outlier-factor/>. 2025. Accessed 30 Aug 2025.

85 Random Forest Regression in Python. Online. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/random-forest-regression-in-python/>. 2025. Accessed 30 Aug 2025.

86 One Hot Encoding in Machine Learning. Online. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/ml-one-hot-encoding/>. 2025. Accessed 30 Aug 2025.

87 What is Retrieval Augmented Generation. Online. GeeksforGeeks. <https://www.geeksforgeeks.org/nlp/what-is-retrieval-augmented-generation-rag/>. 2025. Accessed 30 Aug 2025.

88 Kemppainen, Liisa. Sitowise. Interviewed 3 Apr 2025.

89 89 Vertainen, Virpi. City of Helsinki. Interviewed 16 Apr 2025.

90 First quantitative ROI analysis of the benefits of BIM for Facility Management. Online. Between the Poles. <https://geospatial.blogs.com/geospatial/2018/06/first-quantitative-roi-analysis-of-the-benefits-of-bim-for-fm.html>. 2018. Accessed 30 Aug 2025.