

Research into optimised and future railway infrastructure

S2R-CFM-IP3-01-2020 Innovation Action

Appendix 8.1.1 Track information model

Confidentiality level: CO

Project information

DISCLAIMER AND ACKNOWLEDGEMENT

This project has received funding from the European Union's Horizon 2020 Program Research and Innovation action under Grant Agreement No 101012456.

This document reflects the views of the author(s) and does not necessarily reflect the views or policy of the European Commission. Whilst efforts have been made to ensure the accuracy and completeness of this document, the In2Track3 consortium shall not be liable for any errors or omissions, however caused.

|  |  |  |
| --- | --- | --- |
| Document history | | |
| **Revision** | **Date** | **Description** |
| 1 | 2023-06-07 | Appendix |
| 2 | 2023-08-08 | Renumbering the document and its sub-chapters and figures |
|  |  |  |

**TABLE OF CONTENTS**

[Appendix 8.1.1-1: Task approach and methodology 3](#_Toc142390820)

[Phase 1: The development of a track-based building Information Model (TIM) 3](#_Toc142390821)

[1.1. Define model structure (systems and components) 3](#_Toc142390822)

[1.2. Define a data structure (types and information structure) 4](#_Toc142390823)

[1.3. Identify, collect (or create) BIM-Objects (3D/2D-geometries) needed to create the BIM model (Object Type Library) 4](#_Toc142390824)

[1.4. Collect input data for the model (sample data) 4](#_Toc142390825)

[1.5. Define appropriate tools for creating the artificial environment (BIM model) 5](#_Toc142390826)

[1.6. Evaluation of the design concept (gap analysis) 5](#_Toc142390827)

[Phase 2: Incorporation of the infrastructure regulations (TRVINFRA) 6](#_Toc142390828)

[2.1. Define the model's governing infrastructure regulations. 6](#_Toc142390829)

[2.2. Develop IT concepts to link the regulations to BIM objects. 6](#_Toc142390830)

[Phase 3: The development of a virtual environment as a digital information carrier 9](#_Toc142390831)

[3.1. Combine data and 3D objects to create BIM model 9](#_Toc142390832)

[Appendix 8.1.1-2: GAP analysis 10](#_Toc142390833)

[1. Method 10](#_Toc142390834)

[2. Analysis Result 11](#_Toc142390835)

[Appendix 8.1.1-3: Information model for The Built Environment 15](#_Toc142390836)

[1. Framework 15](#_Toc142390837)

[2. Implementation 19](#_Toc142390838)

[Appendix 8.1.1-4: Design process 23](#_Toc142390839)

[1. Framework 23](#_Toc142390840)

[2. Implementation 25](#_Toc142390841)

[3. Material 28](#_Toc142390842)

[Appendix 8.1.1-5: Information model for the governing requirements 29](#_Toc142390843)

[1. Framework 29](#_Toc142390844)

[2. Implementation 30](#_Toc142390845)

[Appendix 8.1.1-6: Semantic information modeling and linking 32](#_Toc142390846)

[1. Framework 32](#_Toc142390847)

[2. Implementation 33](#_Toc142390848)

[3. Material 36](#_Toc142390849)

[Appendix 8.1.1-7: The virtual environment 37](#_Toc142390850)

[1. Framework 37](#_Toc142390851)

[2. Implementation 38](#_Toc142390852)

[3. Material 39](#_Toc142390853)

[References 41](#_Toc142390854)

# Appendix 8.1.1-1: Task approach and methodology

The primary approach of the task is to identify technical challenges at an early stage and re-evaluate the methods or change working methods if necessary. This will ensure that the activities are executed efficiently to achieve the desired results. The task's objectives will be met using various methods, and the activities will be grouped into three distinct phases. The first phase aims to develop an artificial environment, namely a Building Information Model (BIM), to capture the necessary information regarding the Track superstructure. In the second phase, Trafikverket's relevant infrastructure regulations that govern Track structure will be defined and digitized using the “Semantic web” technology (W3C, 2015). Lastly, in phase three, a virtual environment will be created to serve as a digital information carrier. The proposed solutions aim to improve the management and maintenance of the Track superstructure by providing stakeholders with accurate and up-to-date information about the assets.

## Phase 1: The development of a track-based building Information Model (TIM)

### Define model structure (systems and components)

In this activity, an information model for the artificial environment (BIM model) of a Track superstructure will be established. Unified Modeling Language (UML) will be used to generate and configure the information model. The information model will be structured based on the following concepts:

1. Using the standard "IFC-Rail 4x3" (and/or “IFC 4” the current official release schema) model structure developed by the "buildingSMART" group. (Limited, 2022). The IFC conceptual model describe each railway object as a unique Concept inside the model, meaning it is described one time and for all inside the Model. However, the same object can be seen under multiple points of view. Three major modelling views (aspects) are adopted in the model, Structural (physical), Spatial and Functional. (IFC Rail Team, 2022)

Table 1 Modelling views and their use (IFC Rail Team, 2022)

**En bild som visar text, skärmbild, Teckensnitt, nummer

Automatiskt genererad beskrivning**

For the purpose of this task only the Structural (physical) and Spatial aspects will be used.

1. Using the information model for "Reference facility" developed by Trafikverket in connection with the feasibility study "Virtual Master facility" (Svärdby-Bergman). This model will be used as a frame-work for creating an “Object type Library”, a data-catalog for the Structural (physical) objects and the Spatial objects.
2. Using the information model for "Reference ID", Trafikverket's common hierarchy and reference designations system for assets in a facility.

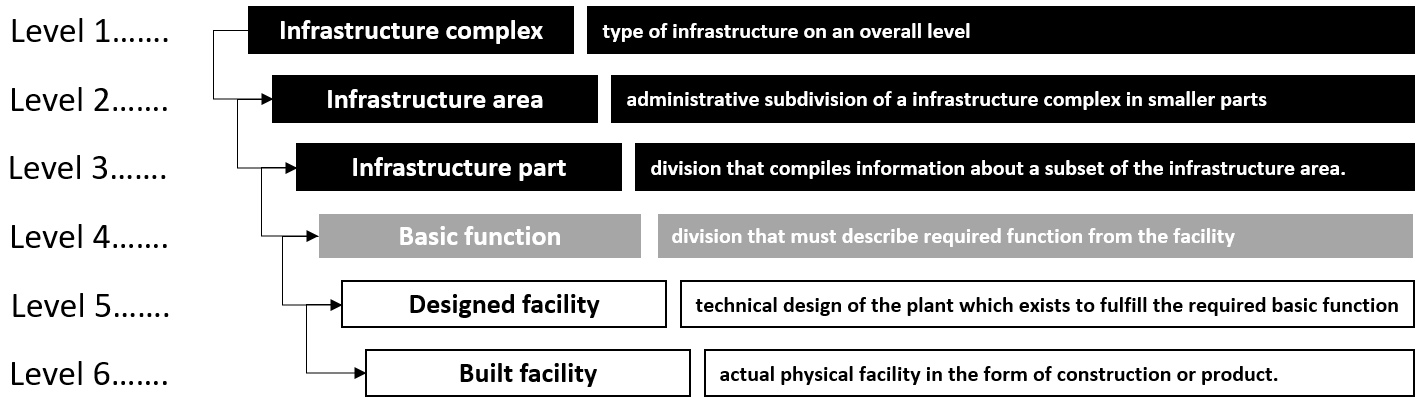


Figure Trafikverket's common hierarchy based on Trafikverket’s guidelines TDOK 2019:0026

The purpose of the information model is to identify the different systems and component types that exist in a Track Structure environment. The information model will describe a combination of spatial structure and a physical objects, displaying the relationships between objects and how they are (de)composed using “Aggregation” (i.e. part of, composed of) and “Generalization” type of relationships.

### Define a data structure (types and information structure)

In this activity, a definition will be established for the different properties that different types of facilities, systems or components have. The activity is performed in parallel with activity "[Phase 1.1](#_Define_model_structure)" and will generate data for the UML model. In addition to the geometric properties, properties defined in the feasibility study "Virtual Master Plant" will be used. Information stored in Trafikverket's various IT systems can then (theoretically) be linked to the BIM model in the next phases of the task with the help of a unique "Object Key".

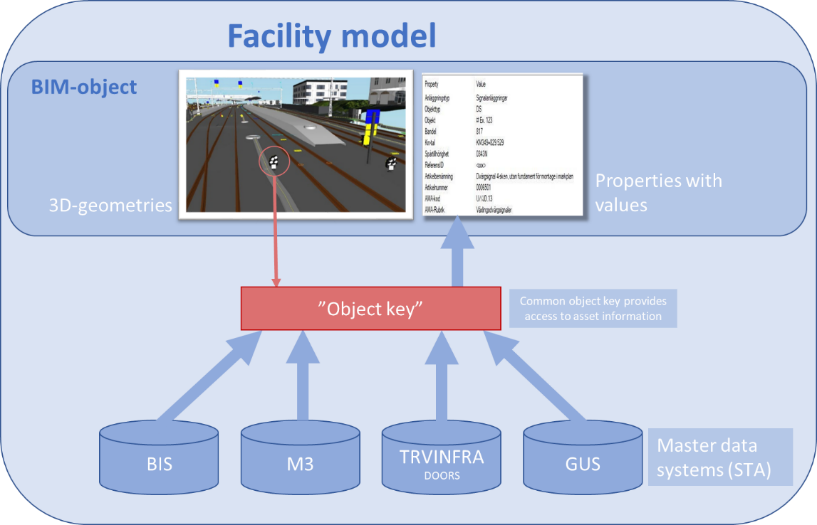


Figure An "object key" enables access to data from multiple systems.

### Identify, collect (or create) BIM-Objects (3D/2D-geometries) needed to create the BIM model (Object Type Library)

In this activity, BIM-objects (3D/2D-geometries) will be created or collected for all the components defined by the UML model in activity 1.1. The BIM objects must be parametric and contain the attributes corresponding to the properties defined in activity 1.2. Different types of CAD / BIM tools will be evaluated to perform this activity. For the purpose of this task, objects created in various Trafikverket’s early projects are collected and reused for the creation of the BIM model.

### Collect input data for the model (sample data)

In this activity, sample data is collected from the Trafikverket's various databases to use in the BIM model. The goal is that all the properties defined in the UML model will contain data. In the event that the information is missing from Trafikverket's databases, a fictitious but “reality-based” data will be created to ensure that all aspects of the UML model are tested in the BIM model to guarantee a high quality for the proof-of-concept model.

### Define appropriate tools for creating the artificial environment (BIM model)

In this activity, different types of CAD / BIM tools will be evaluated to define the most suitable tool that can be used to create a BIM model for the entire Track superstructure part of a railway Track. The tools must meet the following criteria to qualify for use as a BIM design tool:

* Is capable of performing parametric design
* Is capable of performing graphical programming to maximize the efficiency of the design process.
* Can export the design to an IFC file format.

Finally, a BIM model of a limited area will be created to use as a basis for the artificial environment.

### Evaluation of the design concept (gap analysis)

In this activity, a gap analysis will be used to follow up the progress and success of the various above-mentioned activities in Phase 1 and activities in Phases 2 and 3. The analysis will describe the desired result at the end of the project (2023) compared to the start of the project. The analysis will also be used to identify future measures and recommendations for further work. [Se Appendix 8.1.1-2](#_Appendix_B-_GAP)

## Phase 2: Incorporation of the infrastructure regulations (TRVINFRA)

### Define the model's governing infrastructure regulations.

During the project, the new regulations for New mainlines "Technical system requirements New mainlines" version 1.0 (TSK\_NS\_1.0) will used as base for the technical design. "Technical system requirements New mainlines" (TSK NS) is a Trafikverket’s document which together with existing regulations (TDOK / TRVINFRA) contains Trafikverket's technical requirements for planning, design, construction, operation and maintenance for the high-speed system on the Stockholm-Gothenburg / Malmö mainlines. TSK NS will be used for ballasted track system for speeds up to and including 250 km / h, and ballast-free track system for speeds up to and including 320 km / h (Trafikverket, 2021). Trafikverket’s infrastructure regulations “Track components” (Cagatay, 2023) will also be used for requirements about specific track components.

### Develop IT concepts to link the regulations to BIM objects.

During this activity, an IT concept will be developed to establish a connection between regulations and the BIM model. The regulations and requirements from activity 2.1 will be transformed into a digital format and stored in a database, enabling future linking to objects within the BIM model. The objective is to easily reference various requirements that govern the design of systems or components in the track structure.

The primary digitalization method utilized for this task will be the "Semantic web" technology. (W3C, 2015). Three concepts from W3C will be utilized in the development:

1. Ontology (Vocabulary) that defines the concepts and relationships (also referred to as “terms”) used to describe and represent an area of concern (in this case the railway domain). Ontologies are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms. In practice, Ontology (Vocabulary) can be very complex (with several thousands of terms) or very simple (describing one or two concepts only). There is no clear division between what is referred to as “vocabularies” and “ontologies”. The trend is to use the word “ontology” for more complex, and possibly quite formal collection of terms, whereas “vocabulary” is used when such strict formalism is not necessarily used or only in a very loose sense. Vocabularies are the basic building blocks for inference techniques on the Semantic Web. (W3C, 2015)

In this task the “terms” and the relationship that need to be defined are the “Requirement”, the type of “Asset” that makes-up a railway facility, and the “BIM-Model” used to represent the assets in a railway facility.

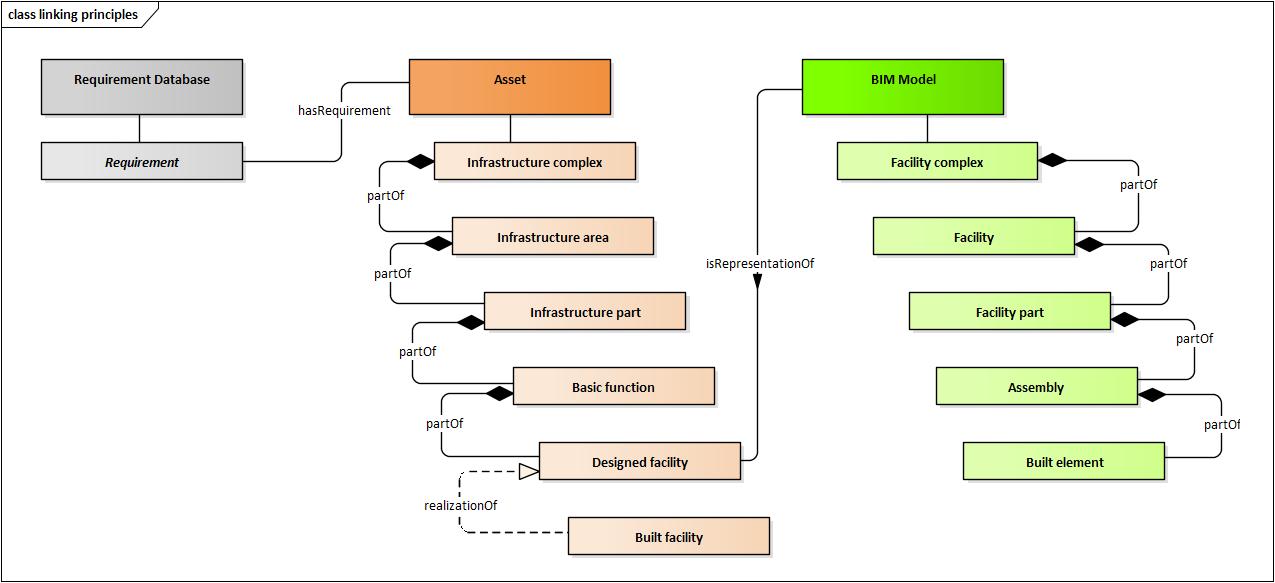


Figure Principle for linking regulations to different objects in the BIM model

When creating an ontology for the IT concept described earlier, it is important to adhere to several principles. These principles serve as guidelines to ensure the ontology's effectiveness and usability. First and foremost, clarity and consistency are paramount. The ontology should offer clear and unambiguous definitions, avoiding any confusion or inconsistency in how information is represented. Precise and coherent terminology is essential in maintaining a well-defined ontology. (Gruninger, 1996)

Another crucial principle is reusability, the ontology should be designed to be adaptable and applicable across various domains and applications. It should have the flexibility to accommodate future changes and expansions while retaining its core structure and principles. To accurately represent the domain and capture relevant knowledge. Involving domain experts is essential, their expertise contributes to the ontology's accuracy and ensures that it reflects the specific requirements and the workings of the IT concept. Modularity is also crucial, breaking down the ontology into modular components or sub-ontologies enhances its manageability and scalability. This modular approach facilitates easier maintenance, updates, and integration with other systems or ontologies. (Gruninger, 1996)

Extensibility is an important principle to consider, the ontology should be designed with the ability to incorporate new concepts, relationships, and properties as the IT concept evolves or as new requirements emerge. Interoperability is another key principle, the ontology should adhere to open standards and best practices, enabling seamless integration and data exchange with other systems and ontologies. This promotes interoperability across different IT systems or domains. Proper documentation is crucial throughout the ontology development process. Comprehensive documentation facilitates understanding, adoption, and maintenance of the ontology. Clear explanations of concepts, relationships, and constraints within the ontology enable users to effectively utilize and extend its capabilities. (Gruninger, 1996)

By following these principles, the creation of an ontology for the IT concept described in Figure 3 can result in a robust and effective representation of knowledge. This ontology will support the linking of regulations to the BIM model and enhance the overall capabilities of the system.

1. Linked Data (semantic modeling and linking) for large scale integration of, and reasoning on, data on the Web. Almost all applications are essentially based on the accessibility of, and integration of Linked Data at various level of complexities. (W3C, 2015). Linked Data is a set of principles that guides the publication, connection, and interlinking of structured data on the web. It was introduced by Tim Berners-Lee, the creator of the World Wide Web, with the aim of making data more discoverable, interconnected, and reusable. The fundamental principles of Linked Data involve the use of Uniform Resource Identifiers (URIs) to uniquely identify resources on the web. URIs provide unique identifiers for entities or concepts, ensuring their uniqueness across the Linked Data space. (Berners-Lee, 2006)

Another key principle is the utilization of the HTTP protocol for data retrieval. By using standard web protocols like HTTP, Linked Data enables data to be accessed and retrieved using web browsers and tools, making it easily accessible to users. Linked Data also promotes the use of the Resource Description Framework (RDF) as a common format for representing data. RDF provides a structured and machine-readable format, facilitating data integration and exchange across different sources. The principle of interlinking plays a crucial role in Linked Data. It emphasizes the creation of explicit links between resources using RDF. These links establish connections and relationships between different data elements, forming a web of interconnected data that can be traversed and explored. (W3C, 2016)

Additionally, Linked Data encourages the inclusion of links to other related data sources. By incorporating links to external datasets, Linked Data enables users to navigate and discover additional information beyond the initial dataset, fostering data integration and expanding knowledge domains. Metadata plays a vital role in Linked Data as well. It involves providing additional information about the data, such as descriptions, provenance, and licensing details. Metadata enhances the understanding and trustworthiness of the published data, enabling users to interpret and utilize it effectively. (W3C, 2016)

By adhering to these principles, Linked Data facilitates the discovery, integration, and reuse of data on the web. It establishes a network of interconnected data, allowing diverse datasets to be seamlessly combined and queried. This opens up opportunities for data-driven applications, knowledge discovery, and innovative uses of interconnected data resources.

1. Query [*in Semantic Web context*] means technologies and protocols that can programmatically retrieve information from the Web of Data. Linked Data, represented using RDF as a data format, needs its own RDF-specific query language and facilities. SPARQL (SPARQL Protocol and RDF Query Language) is a query language specifically designed for working with RDF data. (W3C, 2015)

SPARQL provides a versatile syntax that allows users to define patterns of triples and apply filters and conditions to extract relevant information. Technically, SPARQL queries are based on *(triple) patterns*. RDF can be seen as a set of relationships among resources (i.e., RDF triples); SPARQL queries provide one or more patterns against such relationships. These triple patterns are similar to RDF triples, except that one or more of the constituent resource references are variables. A SPARQL engine would returns the resources for all triples that match these patterns (W3C, 2015). This makes SPARQL a powerful tool for working with interconnected and diverse datasets.

A SPARQL query consists of different components, it begins with a clause, such as SELECT, CONSTRUCT, DESCRIBE, or ASK, which determines the type of query being executed. Patterns of triples are then specified to indicate the desired data to be retrieved. Triple patterns play a crucial role in SPARQL queries, they consist of a subject, predicate, and object, and are used to match specific relationships within the RDF graph. Variables can be introduced using a question mark followed by a name, allowing for flexible querying and capturing of values during execution.

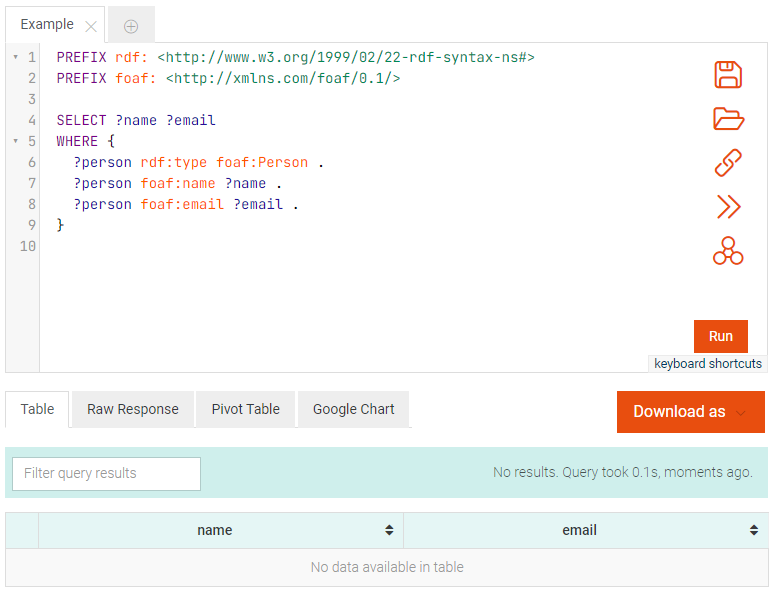


Figure Basic example of a SPARQL query run on Ontotext GraphDB

Query results in SPARQL can be obtained in different formats, such as XML, JSON, or tabular forms. These results provide the values of the queried variables based on the matching patterns and specified conditions. SPARQL offers the ability to apply filters and conditions to further refine query results. Filters allow for comparisons, regular expressions, arithmetic operations, and logical conditions to be applied to the data.

## Phase 3: The development of a virtual environment as a digital information carrier

### Combine data and 3D objects to create BIM model

The final task in this project involves consolidating all the data and linking it with a corresponding 3D object to create a comprehensive BIM model following the structure defined in Phase 1. In conjunction with this activity, various types of "Model Viewer" will be tested to determine the most suitable option based on the following criteria:

1. Visualization of information and 3D elements.
2. Exporting information in alternative formats for further analysis using different tools.
3. User-friendliness.

The objective is to develop a virtual dashboard that effectively visualizes the Track Model and its interconnected data, which can originate from multiple sources if necessary.

# Appendix 8.1.1-2: GAP analysis

The term “gap” refers to the space between “where we are” (the present state) and where “we want to be” (the target state). Gap analysis assesses the differences between the actual and expected performance in an organization or a business. It can also be called a need analysis, need assessment, or need-gap analysis. (Bhat, 2023). A gap analysis is a valuable tool used in IT system- and process development to identify gaps between the current state of an organization's process/IT system and its desired future state.

This analysis aims to bridge the gap by outlining the necessary steps and improvements required to achieve the desired outcomes for this task.

## Method

The method used for conducting the gap analysis in this task will be based on Trafikverkets own internal rutins and the princeples discriped by Abi Bhat in his article “What is Gap Analysis: Definition, Method, and Template” (Bhat, 2023). The method involves the following steps:

1. First, clearly define the objectives of the proposed IT concept for the development project (se [Appendix 8.1.1-1](#_Appendix_A-_task)). These objectives should align with the Trafikverket's strategic goals and may include improving efficiency, enhancing functionality, or addressing specific business needs.
2. Next, thoroughly assess the current state of the IT concept/- system. This involves examining the existing system architecture, software, hardware, databases, and relevant processes and workflows. Understanding the current system's strengths and weaknesses provides a foundation for identifying areas of improvement. For the porposes of this task, the scope will be only the evaluation of the design concept (se [1.6](#_Evaluation_of_the))
3. Once the current state has been assessed, establish the desired future state of the design concept. This ideal state should align with the previously defined objectives and consider factors such as scalability, security, usability, and performance. The desired future state serves as a benchmark for identifying gaps.
4. Comparing the current state with the desired future state reveals the gaps or discrepancies that need to be addressed. These gaps can encompass missing features, performance issues, usability shortcomings, security vulnerabilities, or any other areas where the existing system falls short of expectations.
5. After identifying the gaps, prioritize them based on their impact on the objectives, business needs, and available resources. Prioritization allows for a focused approach in tackling the most critical areas for improvement.
6. To determine the root causes of each identified gap, conduct a thorough analysis. This involves examining factors such as outdated technology, insufficient resources, inadequate processes, or lack of skills that contribute to the gaps.
7. Develop action plans to address each identified gap. These plans should outline specific steps, allocate resources, define timelines, and assign responsibilities. It is crucial to ensure that the action plans are feasible and achievable within the project's constraints.
8. Implement the action plans by executing the necessary changes or improvements to close the identified gaps. This may involve software development, infrastructure upgrades, process reengineering, training, or other activities required to bridge the gaps.
9. Regularly monitor and evaluate the progress of the development project. This helps ensure that the gaps are being effectively closed and the desired future state is being achieved. Monitoring also provides opportunities to identify any new gaps that may arise during the implementation phase.
10. Based on the monitoring and evaluation, make any necessary adjustments to the action plans or project approach. Iteratively refine the development process to continuously improve and address emerging gaps.

By following these steps, Trafikverket can effectively conduct a gap analysis for the development of a new IT-concept/system, enabling Trafikverket to bridge the gaps and align the result with the desired future state.

## Analysis Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Area | Section | Discription (What must be achieved) | Current state | Desired future state | Gap root causes |
| Model structure (information model) for systems and components in a railway track environment | [1.1](#_Define_model_structure) | Creating an information model for the artificial environment, specifically the BIM model, of a Track superstructure. | Trafikverket lacks a unified or standardized information model that can be implemented and supported by various BIM design tools. | The implementation of an ISO-approved schema for the information model in BIM is a key aspect of Trafikverket's projects and virtual environment. Specifically, one of the IFC schemas developed by buildingSmart is primarily utilized. | * 1. Trafikverket's heavy reliance on proprietary tools for CAD and BIM design poses a significant challenge for the organization and the Swedish AEC industry as they transition to an open format like IFC.   2. As of May 2022, the IFC 4x3 standard for infrastructure and railway facilities has not yet received ISO approval.   3. Due to the use of multiple "legacy" IT systems, Trafikverket's information model for managing facilities and assets is currently fragmented. As of May 2022, efforts are underway to develop a centralized model. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Area | Section | Discription (What must be achieved) | Current state | Desired future state | Gap root causes |
| Data structure | [1.2](#_Define_a_data) | The goal is to establish a standardized data model principle for defining the properties associated with various types of facilities, systems, or components. | Trafikverket currently lacks a unified or standardized data model that can be implemented and supported by various BIM design tools. However, Trafikverket does have attribute requirements for facility assets, primarily components, which are intended for facility management and maintenance purposes. | * 1. The implementation of an ISO-approved schema for the data model to be utilized in BIM and other IT systems within Trafikverket.   2. The establishment of a data dictionary comprising object classifications, their properties, allowed values, units, and definitions. This information should be cataloged and managed within an "Object Type Library." | * 1. Due to the use of multiple "legacy" IT systems, Trafikverket's own data model for managing facility assets-related data has become fragmented. As of May 2022, a centralized model is still under development.   2. Trafikverket employs various classification systems for different types of facilities, making it challenging to implement an integrated data dictionary that can be universally utilized across all of Trafikverket's systems. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Area | Section | Discription (What must be achieved) | Current state | Desired future state | Gap root causes |
| Object Type Library (OTL) | [1.3](#_Identify,_collect_(or) | A principle must be established for a standardized Object Type Library to be utilized in the design process and the creation of the virtual environment. | * 1. Trafikverket lacks a centralized Object Type Library (OTL). Different OTLs were created for various projects, each with its own structure (data model) for the information and attributes linked to different object types.   2. Trafikverket lacks an IT system that automates the creation, updating, and distribution of an OTL. Currently, these tasks are manually performed using Excel spreadsheets and distributed via email or SharePoint databases. | * 1. Establishing a centralized Operational Track Layout (OTL) system within Trafikverket that can be accessed by multiple users across multiple projects.   2. Implementing an IT system that supports and automates the creation, updating, and distribution of the centralized OTL. Utilizing the "Semantic Web" technology as the framework for this system would be beneficial.   3. Creating a centralized organization within Trafikverket that is responsible for managing and maintaining the OTL content. This organization would oversee the accuracy and relevance of the information provided.     By implementing these measures, Trafikverket can achieve greater consistency, accessibility, and effectiveness in handling the OTL, leading to improved project outcomes and resource utilization. | * 1. Due to the use of multiple "legacy" IT systems, Trafikverket's own information model for managing facilities and assets information is fragmented. As of May 2022, a centralized model is still under development.   2. Trafikverket currently lacks an established organization or process for creating, updating, or distributing an OTL content. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Area | Section | Discription (What must be achieved) | Current state | Desired future state | Gap root causes |
| Design method an tools | [1.5](#_Define_appropriate_tools) | The establishment of a principle for the design process and the selection of an appropriate tool for creating a virtual environment. | * 1. Trafikverket utilizes traditional CAD tools and methods for creating railway track models.   2. The design process lacks automation and is too complex to allow for more than one iteration.   3. As of May 2022, Trafikverket does not exchange data in an openBIM format such as IFC or XML. | * 1. To enhance the efficiency of the design process, one approach is to establish a parametric design process utilizing modern graphical programming tools. This method allows for the optimization of the design process by leveraging the power of these tools.   2. In order to foster an iterative and agile design workflow, it is beneficial to utilize user-friendly design tools. These tools enable designers to easily make changes, iterate on designs, and adapt to evolving requirements throughout the design process.   3. Adopt the Industry Foundation Classes (IFC) as the primary data exchange format. Standardizing on IFC facilitates seamless interoperability and effective data exchange between different software applications and stakeholders involved in the project. | * 1. Trafikverket's heavy reliance on proprietary tools for CAD and BIM design poses a significant challenge for the organization and the Swedish AEC industry as a whole when transitioning to an open format like IFC.   2. The use of cloud-based design software is prohibited by Trafikverket.   3. Trafikverket lacks in-house design resources and outsources all design-related activities to consultants.   4. Trafikverket does not provide consultants with significant economic incentives to encourage the adoption of a more automated design process. |

# Appendix 8.1.1-3: Information model for The Built Environment

This appendix will include a description of the information model that were used or developed in task 8.1.1 for describing the built environment.

## Framework

The information model serves as the foundation for describing the built environment within the virtual environment model. The entities in the model are categorized into distinct groups based on classes from the ISO standard **12006-2**: “Built space”, “Construction result”, “Construction complex”, “Construction entity”, “Construction element”, and “Construction information”.

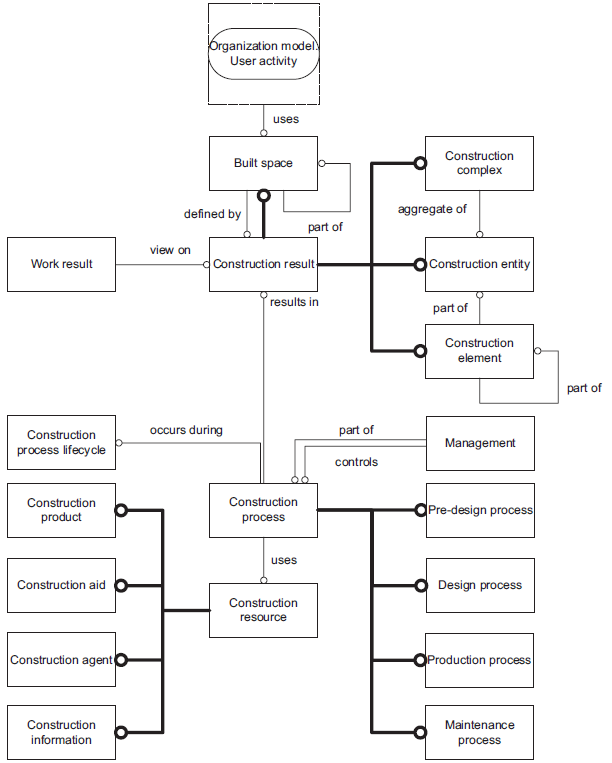


Figure Classes and the general relationship between them. Source SS-EN ISO 12006-2:2020

These groups are:

1. **Spatial entities**, Based on IFC 4x3 and CoClass, three levels of spatial structures were used to breakdown the built environment using BIM. The spatial breakdown of a facility or a building refers to the detailed representation and categorization of different areas and components within the structure. This breakdown is essential for maintenance or facility management and for several other reasons. These levels are categorized as following:

* **Infrastructural complex**: an aggregation of several facilities that function together to serve a certain mode of transportation (e.g., Railway or Road).
* **Facility**: an independent unit in the built environment with a characteristic spatial structure, intended to serve at least one function or user activity that provides a specific function (e.g., Railway Track, Railway bridge or Railway tunnel).
* **Facility part**: a spatial breakdown of a built facility (e.g., Track structure, Superstructure or substructure)

1. **Structural entities**, physical “Built element” which can be subdivided into components or assemblies, which realizes the designed facility part. Several built element units (assemblies) and/or several individual components can be aggregated into Construction System (e.g., Foundation, Signaling or Energy).
2. **Property**, for every entity descried previously there are many properties connected to them. The properties used to describe these entities in different ways and perspectives. They are the actual information holder and can be used to convey information to both humans and other IT-system
3. **Property values**, all the properties have predefined data type (refer to [Appendix 8.1.1-6](#_Framework_2)), and some of the properties have predefined values based on Trafikverket asset management system.
4. **Property sets**, properties that describes similar type of entities or have a similar use purpose were grouped in a “property sets”.

*NOTE: In the initial project iteration, there was a conflation between the "Facility part" and "Construction System" due to limitations in the design tools. These limitations prevented the attachment of "Property sets" to the "Construction System" when exporting the BIM model to the IFC 4 schema.*

The Construction information that describes characteristic properties of every “Asset” of the built environment are structured based on the European standard **EN 17412-1:2020** Building Information Modelling – Level of Information Need. To support information exchange, the level of information need (LOIN) should be used. LOIN describes the extent of information exchanged in terms of:

* Geometric information,
* Alphanumeric information,
* Documentation.

Different purposes have their own requirements for geometric information, alphanumeric information, and documentation. LOIN should be used to discuss and agree on the information delivery between two or more actors. LOIN describes information requirements that can be humanly and machine interpretable.

To specify LOIN (Level of Information Need) and how information should be delivered, the following conditions should be defined (Swedish Institute for Standards, 2020):

1. **Purpose for the use of the information to be delivered**. When LOIN is specified, the purpose of the information delivery should be defined to clarify why the information is needed. LOIN should be used for the purposes it has been required for. During a milestone for information delivery, the same LOIN required for an asset can be used for different purposes. In some cases, the purpose should not be clear to all actors (e.g., for security reasons). In these cases, the purpose should be considered "classified" and only authorized actors should be informed. The following are the main categories of purposes related to Trafikverket's different activities:
   * Analysis: e.g., LCC, LCA
   * Design activities: simulation, verification, monitoring, and approval
   * Production activities: planning, simulation, verification, monitoring, and approval
   * Management and maintenance
   * Visualization: e.g., visualization of information in a digital twin or a VR/AR application.
2. **Milestones for information delivery**. When LOIN is specified, milestones for information delivery should be defined. Within the same milestone for an information delivery, the geometric information, alphanumeric information, and documentation may vary for different assets. Milestones for information delivery and frequency are defined (unless otherwise specified) by the main processes of Trafikverket:
   * Collect and process information about roads and railways.
   * Plan actions on roads and railways.
   * Implement actions on roads and railways.

Milestones for information delivery may occur in conjunction with other processes within Trafikverket.

1. **Actors requesting and actors delivering the information**. When LOIN is specified, actors requiring and providing the information should be identified. LOIN does not specify the actor. The same information needs may be required by different actors at the same milestone to fulfill different purposes, but it is also possible that different LOINs may be required by different actors at the same milestone to fulfill the same purpose. Trafikverket may request a specific LOIN for an object at an agreed-upon milestone for information delivery, without specifying who needs to deliver the level. In this case, the responsible project manager, contractor, or supplier has the freedom to assign responsibility as they deem appropriate. For purposes related to design activities, different actors are often dependent on each other to deliver the specified LOIN for an object to Trafikverket. The project should identify these dependencies at an early stage and regulate the exchange of information between the different actors.
2. **Assets organized in one or more breakdown structures**. When LOIN is specified, the assets within a breakdown structure (hierarchy) should be defined for information delivery. The semantic, functional, and/or spatial breakdown of the facility must be defined for all assets within a facility in order to specify LOIN for these assets (e.g., construction elements and spaces). Based on the purpose, LOIN can be related to:
   1. Construction result (spaces, construction complexes, construction contracts, and construction elements)
   2. Construction information (building information model, specification, documentation, and diagrams).

Breakdown structures can follow a classification system, system engineering principles, or a federation strategy (e.g., Function ID or CoClass). Different purposes may require different breakdown structures.

1. **Management of different versions of LOIN is done through an object type library (OTL)**. A definition of LOIN for a specific asset should be documented and version-controlled in an Object Type Library (OTL). An object in the library should carry/communicate all LOIN definitions. The information delivery should be reviewed and approved according to the latest applicable version of LOIN, which was defined at the project's inception or before a new milestone.

To facilitate the registration and management of these entities, a relational database was created using the BIMQ information management tool ([se Implementation](#_Implementation)).

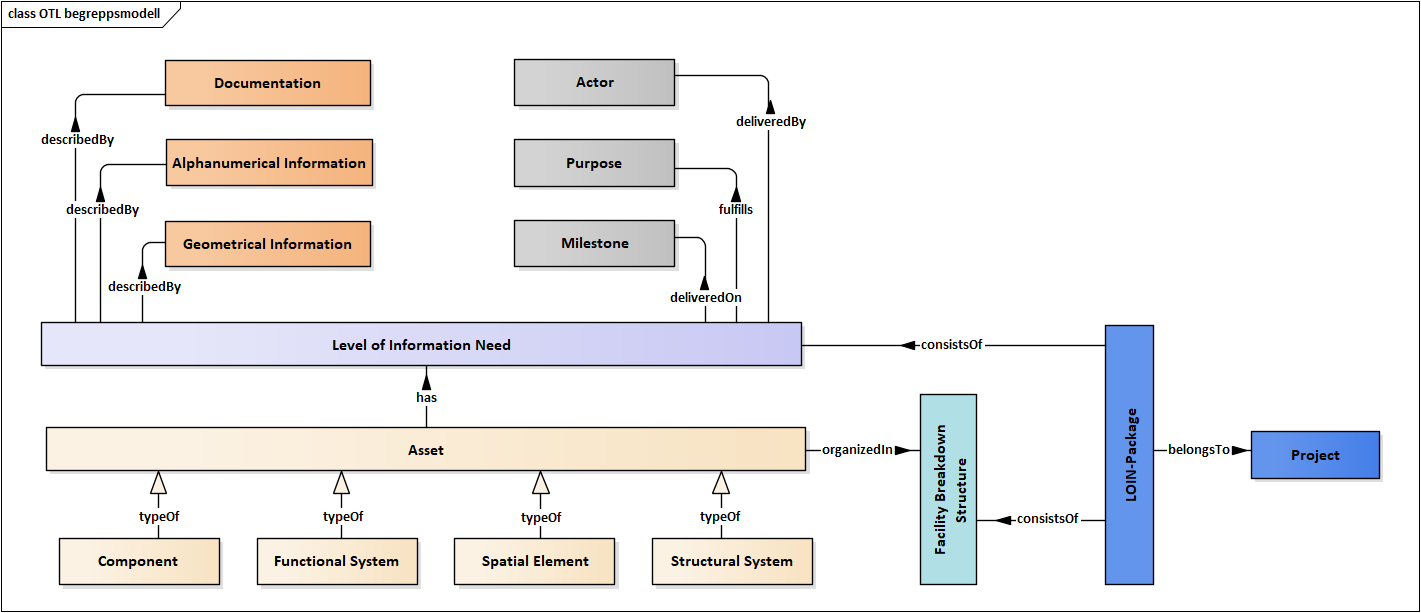


Figure conceptual information model of the built environment within the virtual environment model

## Implementation

Within the spatial breakdown of the model, three levels of spatial structures were utilized based on IFC 4x3 and CoClass (a classification system described farther on in this section). These levels include **Infrastructural Complexes**, **Facilities**, and **Facility Parts**. Trafikverket also implemented an additional breakdown system focusing on **Functional Systems** and **Intended functions** and **Structural systems**. Structural entities, representing building components and assemblies, were identified using Trafikverket's predefined list of standard railway components, TDOK 2022:0006 (Davidson, 2022) and registered in BIMQ.

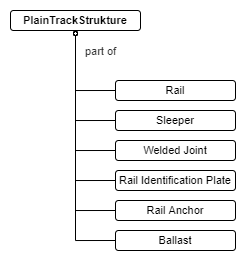


Figure Structural breakdown of Plain Track Structure used in the modern Swedish railway network

*NOTE: Trafikverket's additional breakdown system, which focuses on Functional Systems, Intended Functions, and Structural Systems, is beyond the scope of this task and will not be defined or described in great detail. These concepts will be included in future research.*

The properties associated with the entities play a crucial role in describing them from various perspectives. Every entity (asset) mentioned earlier is associated with several properties, these properties act as the actual information holders and serve to convey information (data) to both humans and IT systems.

To enhance organization and manageability, properties that describe similar types of entities or serve a common purpose are grouped into "property sets." Each property has a predefined data type, and some of them even come with predefined values, which are based on Trafikverket’s asset management system.

Based on Trafikverket’s asset management system “BIS” as a starting point, a comprehensive list of properties intended for asset management purposes has been identified. These properties were subsequently organized into distinct property sets, each corresponding to a specific entity. Finally, to streamline the process and ensure effective management, all the property sets were registered in BIMQ.

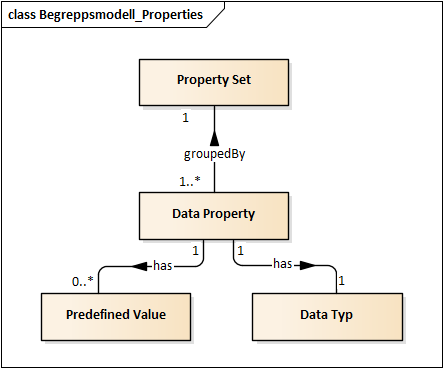


Figure conceptual information model of data properties.

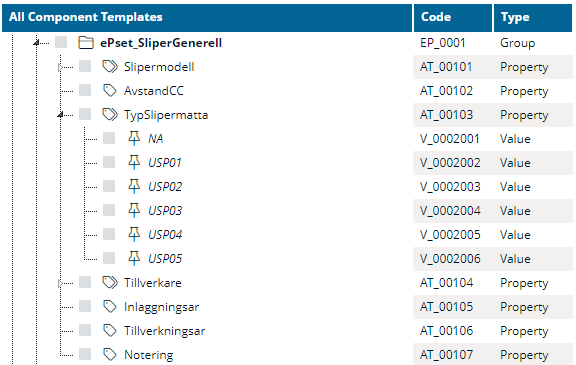


Figure BIMQ table for a property set “ePset\_SliperGenerall”, associated to the asset sleeper and contains multiple properties and their predefined values, for example model, manufacturer, installation year etc.

To classify and organize the entities, a CoClass system was implemented. Each entity class, subclass, and type have their own classification code, and are registered in BIMQ. The classification codes serve as "Object Keys" for linking information stored in a mockup database to the BIM model.

Using the assets previous type definitions (see Figure 6), the assets were classified and organized under the CoClass system as:

* Facility Complex classified as Construction Complex (CoClass BX)
* Facility classified as Construction Entity (CoClass BV)
* Facility Part classified as Constructive System (CoClass KS)
* Assembly/ Component classified as Component (CoClass KO)

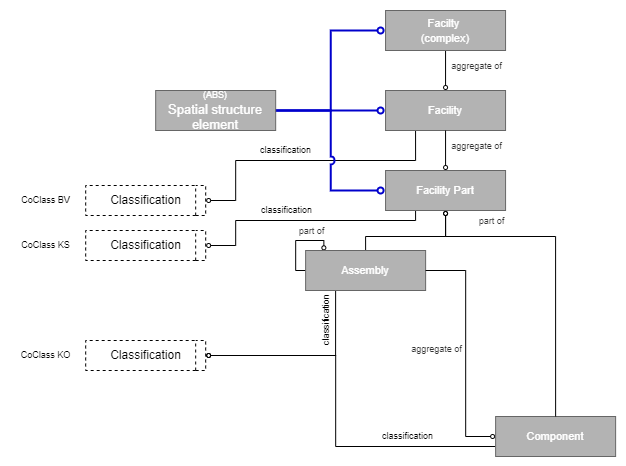


Figure Spatial and structural breakdown of the BIM model classified by CoClass

CoClass is based on the following international standards:

1. SS-ISO 12006-2:2020 - Building construction – Organization of information about construction works – Part 2: Framework for classification. This standard provides the basic structure and division of CoClass into different tables.
2. SS-EN IEC 81346-1:2010 - Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules. This standard contains the rules for reference designations used to describe digital objects and give them a unique identity.
3. SS-EN IEC 81346-2:2019 - Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 2: Classification of objects and codes for classes. This standard includes classes for building components and spaces.
4. ISO 81346-12:2019 - Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 12: Construction works and building services. This standard includes classes for building systems.

The information model was constructed using the IFC file format. Although the current official version of IFC (IFC 4) is primarily used for building-related structures and systems, a new rail/road IFC standard (IFC 4x3) as for May 2022 is still under development by BuildingSmart. To accommodate railway-related entities and properties, the information model was developed in both IFC 4 and IFC 4x3. Spatial entities like “Building” and “Building Story” were used instead of “Railway” and “Railway Part” (se the example in Figure 11). User-defined entities and properties were used to complete IFC 4 with railway-related information.

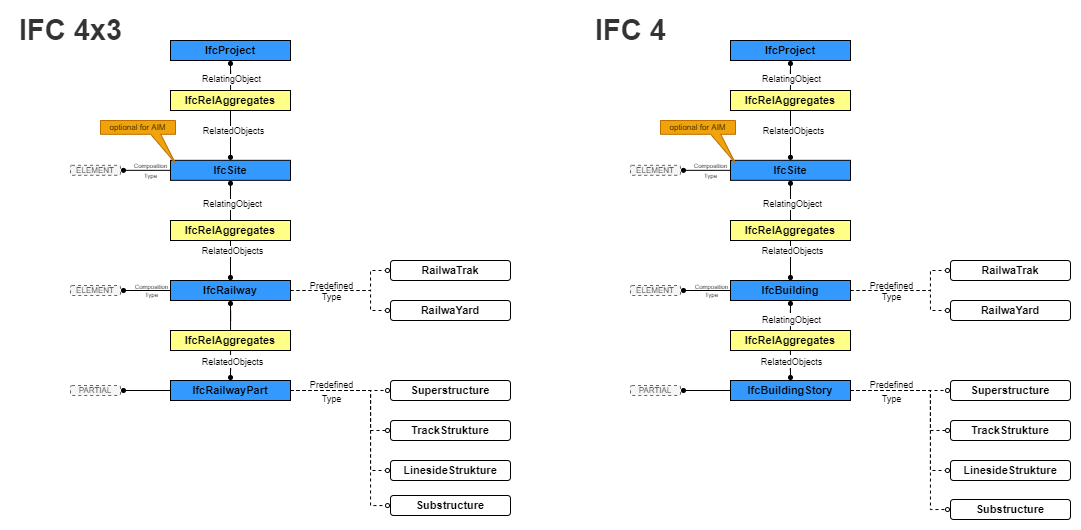


Figure Spatial and structural breakdown of the BIM model in IFC 4 and 4.3 schema

In this task the concept of “LOIN-packages” were implemented. A LOIN-Package is a group of different requirements, properties etc. that are connected to a building element (asset) based on the information model described earlier in [Appendix 8.1.1-3](#_Framework). The different building elements included in the model will functions as a placeholder for all property sets (information) that may be needed for different purposes (requirements, management, simulation, design), required by a different actor and delivered in different millstone.

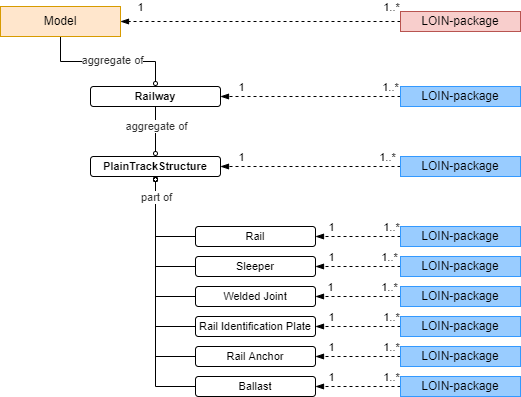


Figure LOIN-Packages for the different Buildings element (Assets) in the model

# Appendix 8.1.1-4: Design process

This appendix will include a description of the design process used or developed in task 8.1.1 to create a BIM model of a railway track environment (Track Information Model).

## Framework

During activity [1.5](#_Define_appropriate_tools), several methods and tools capable of performing parametric design, graphical programming, and exporting designs to the IFC file format were identified. The assets comprising a railway track and considered in the design process can be categorized into the following groups:

* Liner objects, elements swept continuously according to the 3D alignment created in the design phase (e.g., Rails, Ballast)
* Track objects, discontinuous elements arranged at regular intervals according to the object-creation perspective of 3D alignment (e.g., Sleepers) (Kwon, Park, Jang, & Lee, 2020)

This limited the choice to the flowing tools:

* Autodesk Civil (combined with Dynamo)
* Autodesk Revit (combined with Dynamo)
* Bentley OpenRail (combined with Generative components)
* Quadri Novapoint

*Note: Quadri Novapoint has no graphical programming application but is a powerful tool for track design.*

*Note: In the initial project iteration, both Autodesk Revit and Civil 3D were used.*

The design process offers several possible entry points, but a common milestone is the inclusion of a Civil 3D model with at least a corridor or an alignment accompanied by vertical profiles. The primary objective is to establish a mechanism that enables near real-time updates to the model through Civil 3D. This is accomplished by configuring data shortcuts in Civil 3D and utilizing dynamic input updates, such as LandXML for alignments, surfaces, and linear targets. (Autodesk, 2019)

The next step involves defining the dynamic relationships within the linear coordinate systems that describe the asset. This essentially means defining the volumes along the asset. In Civil 3D, this is accomplished through the use of assemblies, regions, and targets. Assemblies (and subassemblies) refer to the parametric components that define the cross-section of a linear asset (See Figure 13). Regions represent the segments along the linear asset where a specific cross-section is applied. Targets, on the other hand, consist of entities and constraints that enable control over the overall geometry of the linear asset. These targets include surfaces, alignments, vertical profiles, feature lines, and polylines. (Autodesk, 2019)

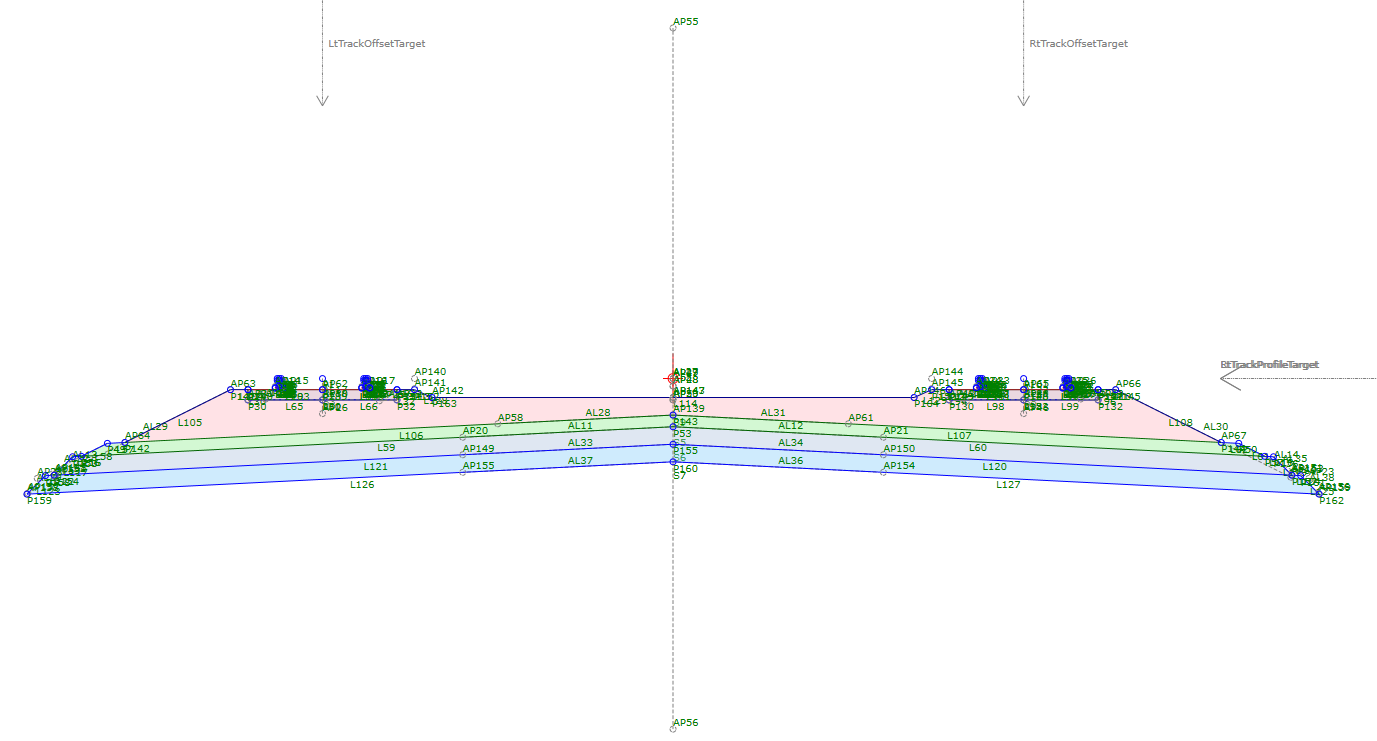


Figure 13 Parametric Assembly of a double track railway

*Note: the concept of Civil 3D Assemblies is implemented in the other design tools under different names.*

Assigning each assembly to a specific region within the asset is crucial. The determination of division points for these assignments forms an integral part of the modeling strategy, which should be based on the design process, work breakdown, and facilitating downstream BIM applications. Civil 3D is responsible for positioning and maintaining assemblies, following the frequency settings established in a corridor. (Autodesk, 2019)

The next step involves defining feature lines, which are linear coordinate systems used to establish dynamic relationships with other objects managed through Dynamo. These relationships can be stored in Revit using shared parameters, on Civil 3D objects, or serialized in formats such as spreadsheets, XML, or similar. When changes are made to the configuration of the linear coordinate systems, the design will automatically update by recalculating the values of the dynamic relationships. This update will be reflected in the model in Civil 3D and Revit through the use of Dynamo. (Autodesk, 2019)

Final step involves adding properties to different objects (assets) in the model using Civil 3D or Revit native tools (or alternatively using Dynamo) before generating an IFC model according to the schema described in [Appendix 8.1.1-3](#_Implementation). (See Figure 11)

*Note: some of the attribute and the model schema can be added/adjusted after the initial export of the IFC-file from the othering tools (Revit or Civil 3D) by using different othering tools like SimpleBIM.*

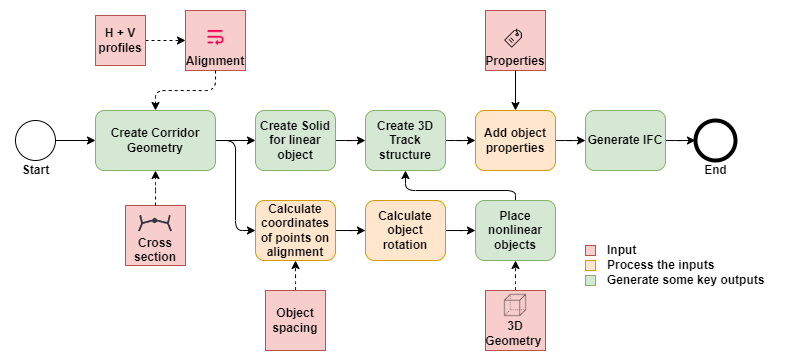


Figure 14 The design process used to create a BIM model of a railway track environment (Track Information Model)

## Implementation

The design process was implemented in two iterations through two pilot projects. In the first pilot, only Autodesk products were utilized, as described in the earlier [framework](#_Framework_1). The second pilot took place within the context of a TRL 6 trial (Technology demonstrated in relevant environment) and involved Trafikverket's internal development program "Technical Platform" in collaboration with the investment project "East Link." For the second pilot, the consulting firm "Norconsult" conducted the design process using a different set of tools to verify the process's validity in a real investment project.

The result of the first pilot included the development of a Dynamo script based on the technical methods and tool described in the rapport “Generating, Transforming, and Analyzing Railway Design Data in Civil 3D and Dynamo” by Wouter Bulens, and the article “Computational Design for Civil Engineers” by Paolo Emilio Serra and Safi Hage.

The tool "CivilConnection" facilitates the exchange of information among Civil 3D, Dynamo, and Revit. It extracts information from rich linear objects, such as Civil 3D alignments, corridors, or feature lines, and generates proxy elements within Dynamo. These proxy elements can then establish dynamic relationships to drive the creation of various discrete Revit elements, including single point family instances, line-based objects like structural framing or MEP segments, and complex objects like adaptive components, floors, walls, and Revit link instances.

Furthermore, the tool offers functionalities to update the location, orientation, and metadata of Revit elements based on input from Civil 3D. It reads the shapes and connections of Civil 3D corridors, and without requiring tessellation, it generates and updates editable Revit families. This allows for further detailing in Revit, utilization of parts and rebar, assignment of custom materials to objects, and preparation for the construction phase. Additionally, it can retrieve or modify the parameters of subassemblies in a corridor and trigger the corridor to rebuild (Serra & Hage, 2019).

In the first pilot, CivilConnection were used to place concrete sleepers along 2 km railway track at 0,6 m distance so that the location and orientation of the sleepers is connected to the feature lines in the corridor using the work flow described in Figure 15.



Figure 15 Process for adding concrete sleepers (Ties) to the BIM model, sorce (Serra & Hage, 2019)

Using the Dynamo script, the placement and the rotation of every sleeper element is calculated to make these elements orthogonal to the alignment, considering the superelevation of the track at each position (see Figure 16)

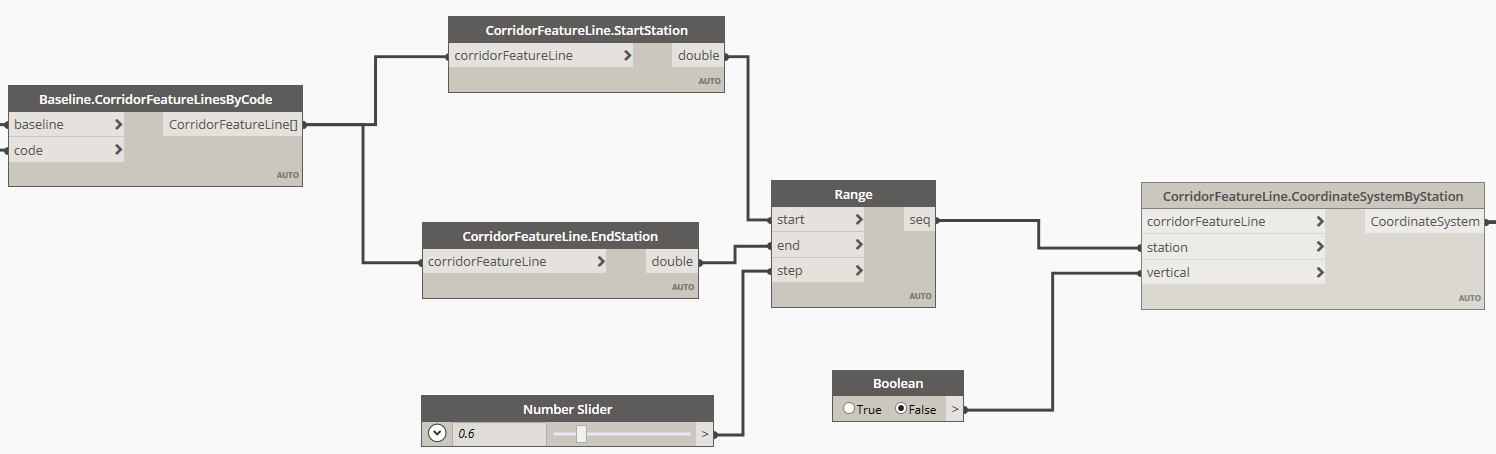


Figure 16 Dynamo script for the calculation of the placement and the rotation of every sleeper element

The next step in the process is the placement of the sleeper elements at each position using prepared CAD model that represents a single concert sleeper, see Figure 17.

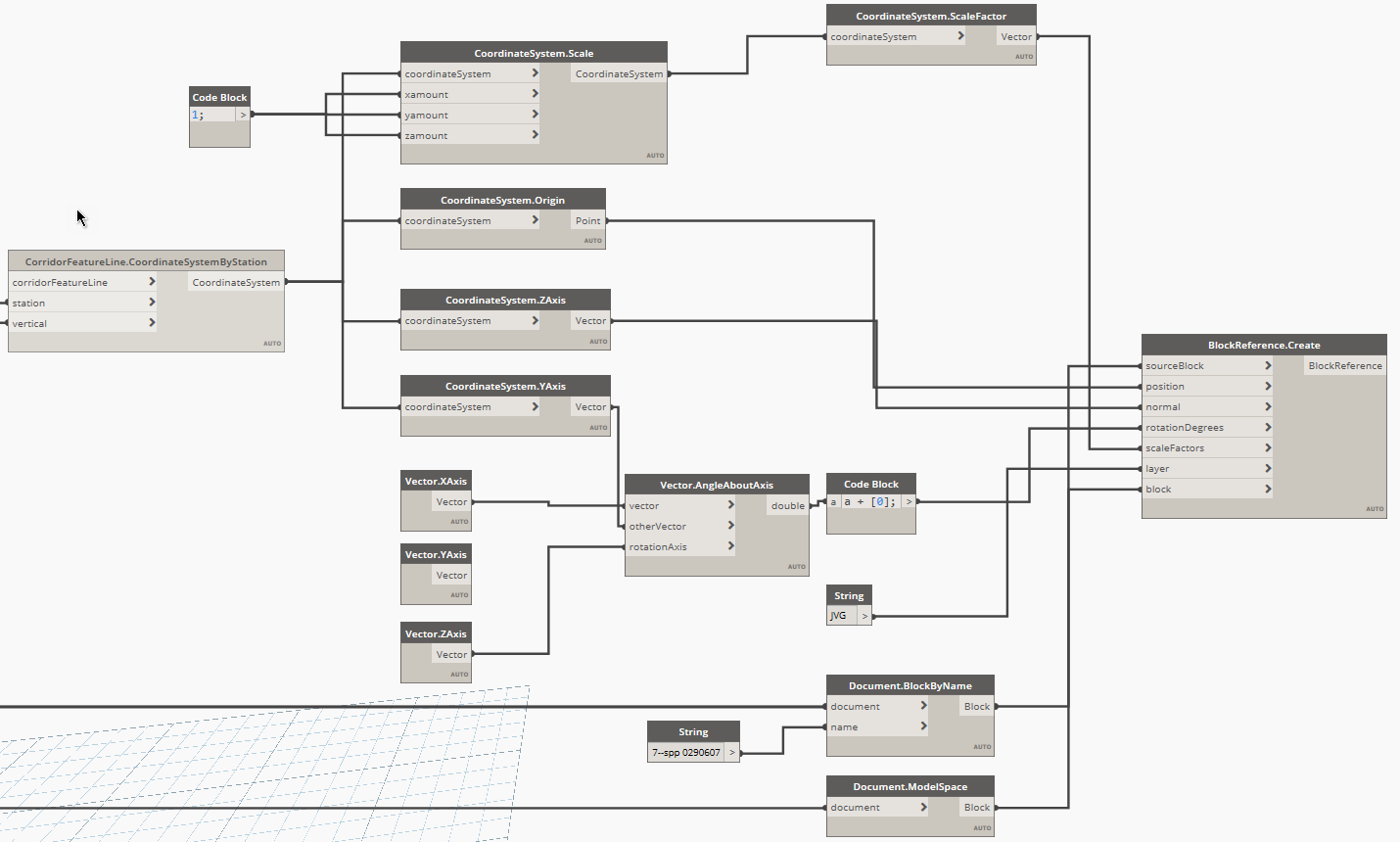


Figure 17 placement of the sleeper elements at each position

The same process can be repeated for each track and different types of elements. In Figure 14, the last step of the design process involves adding attributes to each element in the BIM model, which conveys the information requirements as described in [Appendix 8.1.1-4](#_Appendix_D-_Information). This step of the process could be performed using different methods. For the first pilot the information was manually added to the objects using Civil 3D “Property Set Data”.

*NOTE: Manually adding attributes to objects is not recommended for large-scale models in real projects. This step should be automated and can indeed be automated.*

Norconsult used different tools in the secund pilot, the design tool Microstation, the Quadri database, and Naviate Simple BIM have been used in the design process. In the secund pilot the project objective was to design the multiple facility parts, track structure, the substructure and the catenary system. Deferent BIM models were created for each facility part.

Two parallel workflows in different software have been employed for the development of Trak structure models. The foundation for the track geometry was imported into Microstation, where the rails and sleepers were modeled for the track. The files were segmented and exported to Dgn format. The basis for the track geometry (a LanXML file) was imported into Quadri, where the ballast was modeled as part of the Trak structure model. The Trak structure models were segmented by the different types of the cross-section of the Substructure.

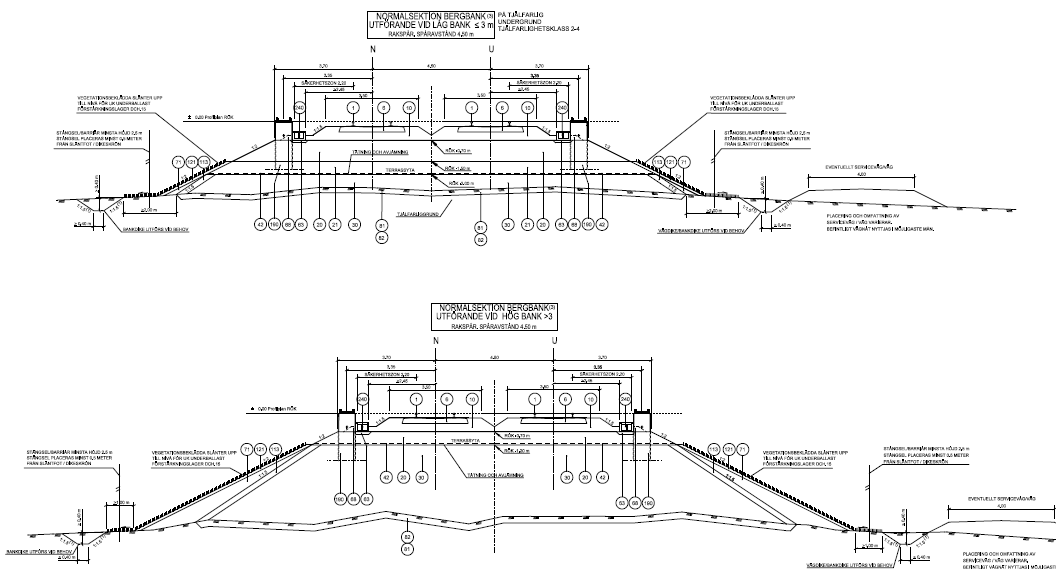


Figure 18 example of two types of cross-sections for railway track substructure, used in the secund pilot project.

The Dgn files were imported into the Quadri database. Ballast, along with sleepers and rails from the respective Trak structure models, were placed in specific selection activities (Quadri packages). This was done to facilitate management and save time during modifications.



Figure 19 Track structure models organized in multiple Quadri packages

Subsequently, a property set was created. A property set is an activity in Quadri used to add user-defined properties to objects, allowing attribute data to be added at the component level. This should not be confused with our ePset, which is defined differently in the database. Finally, all geometric volume objects were enriched with component-level attributes through the property set editor. Then, the selection packages were exported to IFC using the export rule (Rail 2 IFC).

Exported IFC files from Quadri were opened in Naviate Simple BIM. The models were then enriched with ePset at the levels of building complex (Facility complex in IFC 4x3), Building (Facility in IFC 4x3), and structural system (Facility part IFC 4x3). Unwanted property sets were removed to refine the models. The models were structured according to [Appendix 8.1.1-3](#_Appendix_C-_Information) Figure 10.

## Material

The resulting material produced during the pilot projects were uploaded to a GitHub repository.

**Pilot project 1**

The IFC model:

<https://github.com/karrarhm/S2R_TIM/blob/main/Model/Pilot_1/RFA_NS_250_1_TypSek1_Uppdelat%20(EDITED).ifc>

The Dynamo script:

<https://github.com/karrarhm/S2R_TIM/blob/main/Model/Pilot_1/PlaceraBlock_2.dyn>

**Pilot project 2**

GitHub's file size limitation for repository uploads posed difficulties when attempting to upload the BIM models for this project. To address this issue, the IFC models were converted into a single NavisWorks model (NWD format). This solution effectively reduces the size of the models while preserving both the geometry and the information contained within them.

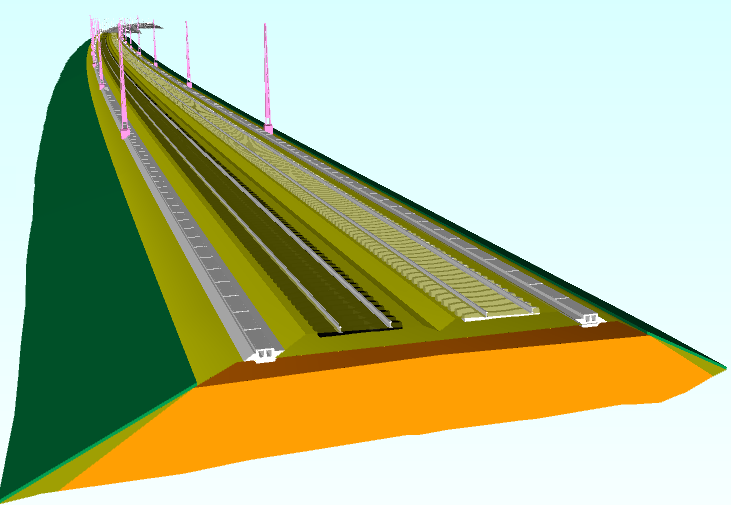


Figure 20 the resulting NavisWorks model from the second pilot project

<https://github.com/karrarhm/S2R_TIM/blob/main/Model/Samordningsmodell%20(Referenssektion).nwd>

# Appendix 8.1.1-5: Information model for the governing requirements

This appendix will include a description of the information model used or developed in task 8.1.1 to describe the structure of a governing requirement and the asset it pertains to.

## Framework

The information model used in this task, which describes the structure of railway facility assets requirements, is based on the "Swedish requirement terminology" created and published by Patrik Sternudd (Sternudd, 2020). This terminology incorporates concepts from multiple ISO standards:

* ISO/IEC/IEEE 15288:2015 Systems and software engineering – System life cycle processes.
* ISO/IEC/IEEE 29148:2011 Systems and software engineering – Life cycle processes – Requirements engineering.
* ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description.

The starting point for Swedish requirement terminology is that the work on requirements is a part of the effort to establish or manage a system of any kind. ISO 15288:2015 defines a system as follows: a combination of interacting elements that are organized to achieve one or more stated purposes.

There are three important aspects in the definition. The first is that there are one or several underlying purposes with a system. The second is that a system consists of a number of underlying elements, and the third is that the definition does not say anything about what these elements actually are. The definition is intentionally broad to be applicable across industries, and there is nothing in it that states that a system necessarily has to be technical. For example, a system can be administrative and consist of various processes. However, Swedish terminology for requirements, is adapted for socio-technical systems: a system where both people and technical artifacts are included as system elements. A trainset can be regarded as a technical system, while the trainset together with the driver can be seen as elements in a sociotechnical system whose purpose is to transport passengers or goods between desired locations. Additional elements in such a system could be the routines and regulations that ensure the driver has the appropriate training and is in good physical health (Sternudd, 2020).

The conceptual model in this section illustrates the relationship between requirements and a selection of other central concepts in Swedish requirement terminology as defined below (Sternudd, 2020):

* **Requirement**: An expression that conveys a need along with its limitations and conditions.
* **Architecture**: fundamental concepts and characteristics of a system within its environment that permeate the components, their relationships, and the principles of system design and evolution.
* **Architectural description** (work product): a description of a system's architecture.
* **System**: a combination of interacting elements organized to achieve one or more stated purposes.
* **Stakeholder**: an individual or organization that is affected by or has influence over the design or existence of a system.
* **Set of requirements**: two or more requirements that can conceptually be grouped together.
* **Requirement set property**: a characteristic that a set of requirements has.
* **Requirements hierarchy**: a hierarchical structure consisting of a goal, as well as requirements broken down and refined from the goal.
* **Goal**: is a requirement at the highest level of abstraction within a given context.
* **Requirement type**: a collective term or categorization for requirements that conceptually belong together.
* **Requirement property**: a property that a requirement has.
* **Requirement attribute**: an attribute that is linked to a requirement.
* **Abstraction level**: how concrete or abstract something is.
* **Requirement notation**: a notation intended for documenting requirements.
* **Notation**: a description technique intended for a specific purpose.
* **Degree of formality**: the level of mathematical formalism that a given notation has.

*Note: The formality level in a notation has nothing to do with the style level in a text paragraph. A formally written document is not the same as one or more requirements expressed in formal notation. There is a significant risk of confusion and misunderstanding in this regard. An example of when one must be particularly cautious is when authors use the term "formal specifications" because it can be used in both cases.* (Sternudd, 2020)

* **Presentation format**: the expression form that a given notation has.

The model utilizes the notation of a UML v2.5 class diagram (originally in Swedish). An important concept in UML is multiplicity, which represents a range with a lower and upper limit. Relationships with arrows should only be read in the direction indicated by the arrows. When the multiplicity for an outgoing relationship is one, it should be read as "each [concept]" or alternatively "a [concept]". When the multiplicity for a relationship includes an asterisk (\*), it refers to a finite non-negative integer. A single asterisk is thus an abbreviation for "0..\*".

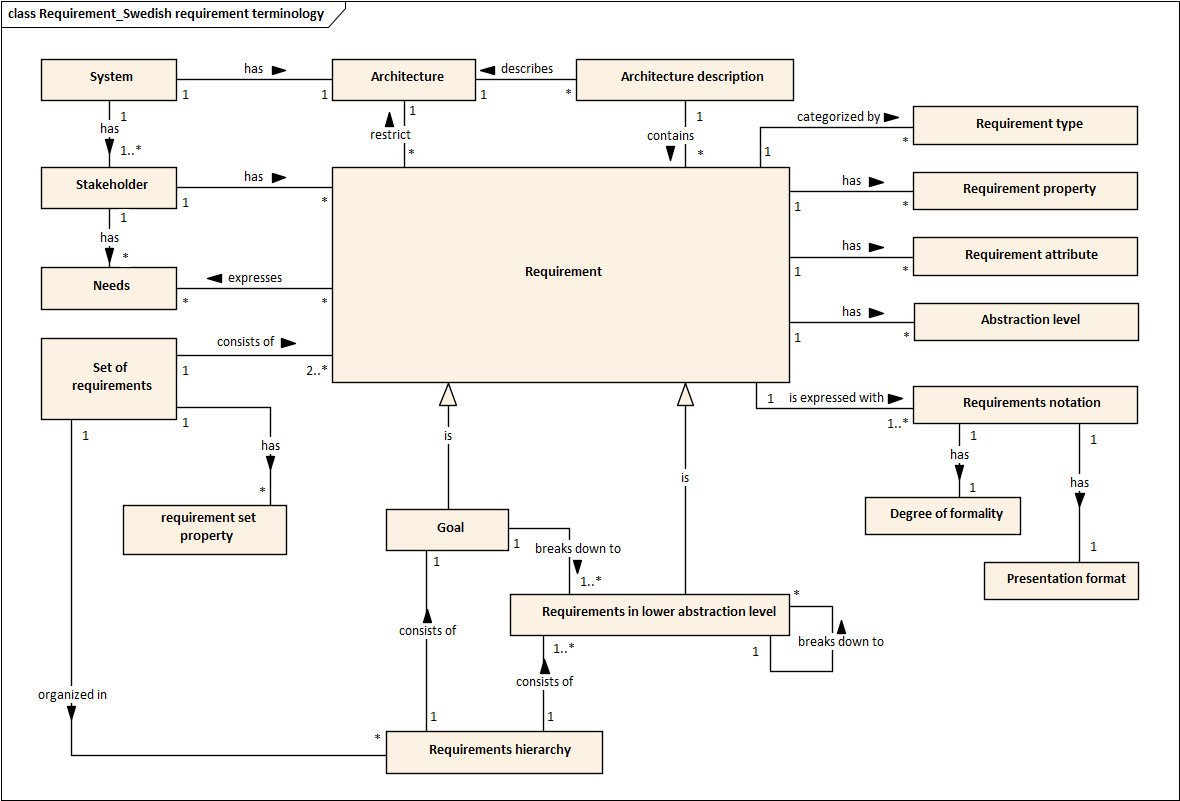


Figure Conceptual model illustrating the relationship between requirements and a selection of other central concepts. Source (Sternudd, 2020)

However, as of May 2022, Trafikverket has not implemented all the concepts described in the above model. Trafikverket's requirements are currently organized in documents without a clear definition of the requirements hierarchy, goals, needs, or stakeholders.

## Implementation

To achieve the objective of this task, a simpler model was created to describe the structure of governing requirements. The new model incorporates concepts from Swedish requirement terminology as well as concepts from the European standard **EN 17632-1:2022** "Building information modelling (BIM) – Semantic modelling and linking (SML) – Part 1: Generic modelling patterns." Within the SML framework, a semantic relationship called "hasRequirement" is defined, which connects different types of entities (in this case, assets) to multiple requirements (0..\* in some cases).

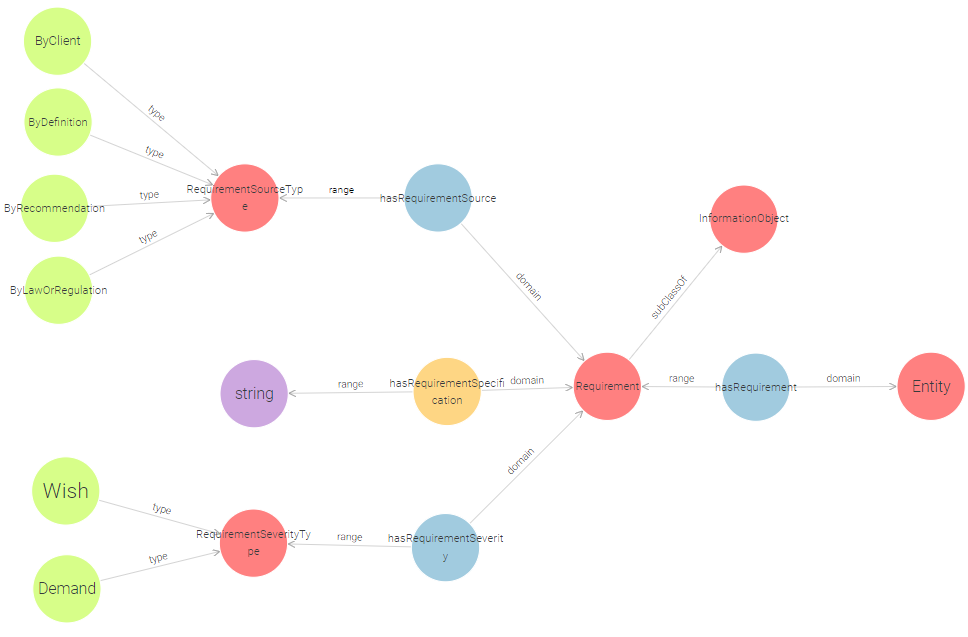


Figure Conceptual model illustrating the relationship between requirements and entities based on Semantic modelling and linking (EN 17632-1:2022)

The new model divides the requirements into different categories (Requirement types) based on Trafikverket's existing set of requirements. The asset is defined and subdivided according to the concepts described in [Appendix 8.1.1-3](#_Appendix_C-_Information).

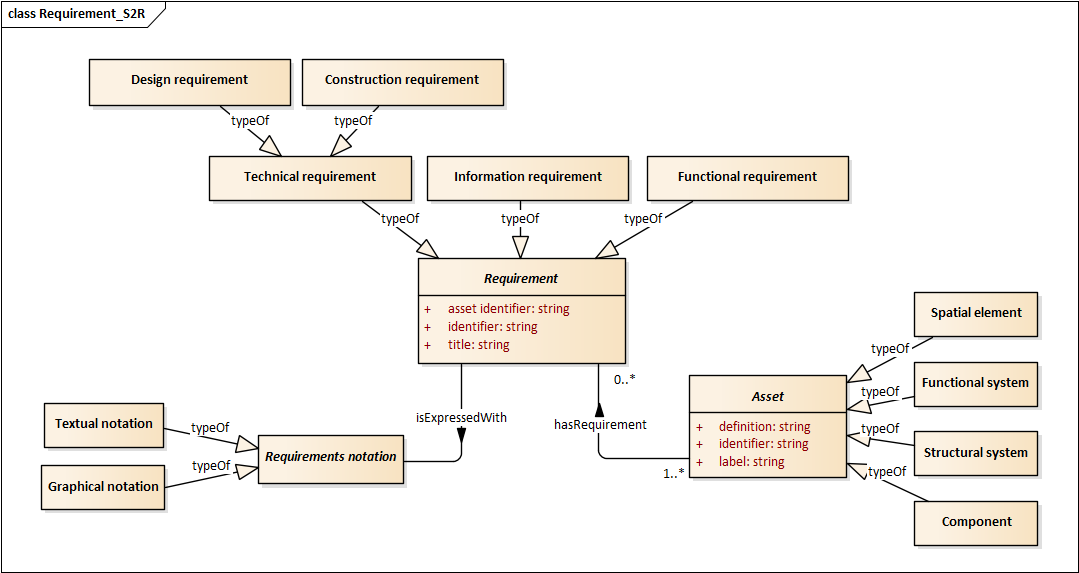


Figure Information model used to describe the structure of a governing requirement and the asset it pertains to.

# Appendix 8.1.1-6: Semantic information modeling and linking

This appendix will include a description of the sematic information model (Ontology) used or developed in task 8.1.1 to describe the structure of a governing requirement and the asset it pertains to. This work incorporated part from Appendix 8.1.1-3: Information model for The Built Environment and Appendix 8.1.1-5: Information model for the governing requirements

## Framework

During activity [2.2](#_Develop_IT_concepts) , we explored various frameworks and ontologies related to defining the building environment. The goal is to reference established components (Entities) from existing ontologies instead of creating entirely new ones. An ontology comprises key elements such as **classes**, **object properties**, **data properties**, **annotation properties**, **datatypes**, and **individuals**.

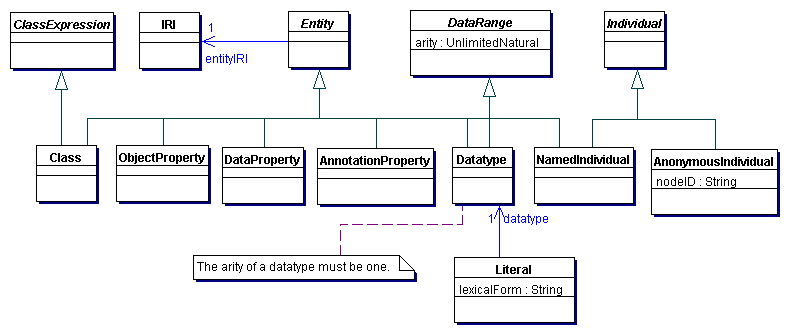


Figure diagram for the different type of entities (Literals, and Anonymous Individuals) based in OWL 2. Source (Motik, Patel-Schneider, & Parsia, 2009)

1. Classes: Classes represent concepts or categories in an ontology. They define the types of objects or entities that can exist in a domain. Classes establish the structure and hierarchy of the ontology, with subclasses and superclasses forming a taxonomy of related concepts.
2. Object Properties: Object properties describe relationships between individuals or instances of classes. They represent binary relationships where the subject and object are both individuals. Object properties define how instances of classes are connected or related to each other.
3. Data Properties: Data properties are used to describe attributes or characteristics of individuals. They associate individuals with specific data values or literals. Data properties are typically used to capture information such as names, ages, dates, or other measurable or textual attributes.
4. Annotation Properties: Annotation properties are used to add metadata or additional descriptive information to ontology elements. They are used to annotate classes, object properties, data properties, or individuals with comments, labels, documentation, or other informative details that aid in understanding and interpretation.
5. Datatypes: Datatypes define the range of possible values for data properties. They specify the type or format of the data that can be associated with a particular property. Common datatypes include strings, integers, booleans, dates, and floats.
6. Individuals: Individuals represent specific instances or members of classes in an ontology. They are the actual entities or objects that exist in a domain. Individuals have properties and can be related to other individuals through object properties.

Together, these key parts of an ontology work in concert to define the structure, relationships, and characteristics of the domain being modelled. Classes establish the conceptual framework, object and data properties capture relationships and attributes, annotation properties provide additional metadata, datatypes define the range of values, and individuals represent specific instances within the ontology. Collectively, they enable the representation and reasoning about knowledge in a structured and formalized manner (Motik, Patel-Schneider, & Parsia, 2009).

The objective of this task is to model the concept introduced in Appendix 8.1.1-3 and -5 flowing the information model in Figure 23. The resulting ontology should define the classes “Assets”, “Requirements” and “Requirement notation”.

## Implementation

The first ontology to be developed (TrvClass) is one that can store information about various types of assets. For this task, the focus is solely on Components. This ontology is based on a previous work done by the consulting firm Triona in collaboration with Swedish Building Centre, that converted the classification system CoClass into an RDF-format.

The ontology “TrvClass”, will contain component types based on CoClass and ISO 81346-2, build on the flowing entities in the flowing table:

Table 2 basic entities used in the ontology TrvClass

|  |  |  |
| --- | --- | --- |
| Entity | Entity type | URI |
| Creator | Annotation properties | <http://purl.org/dc/elements/1.1/creator> |
| Title | Annotation properties | <http://purl.org/dc/elements/1.1/title> |
| Abstract | Annotation properties | <http://purl.org/dc/terms/abstract> |
| Created | Annotation properties | <http://purl.org/dc/terms/created> |
| Preferred Namespace Prefix | Annotation properties | <http://purl.org/vocab/vann/preferredNamespacePrefix> |
| Preferred Namespace Uri | Annotation properties | <http://purl.org/vocab/vann/preferredNamespaceUri> |
| Definition | Annotation properties | <http://www.w3.org/2004/02/skos/core#definition> |
| Category Identifier | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/categoryIdentifier> |
| Example | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/example> |
| Identifier | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/identifier> |
| Is Category | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/isCategory> |
| Note | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/note> |

*NOTE: the URI variable “trvclass” (*[*https://www.trafikverket.se/ontologies/classification\_81346\_2/*](https://www.trafikverket.se/ontologies/classification_81346_2/)*) is inactive URI, used only as placeholder for the purpose of this project.*

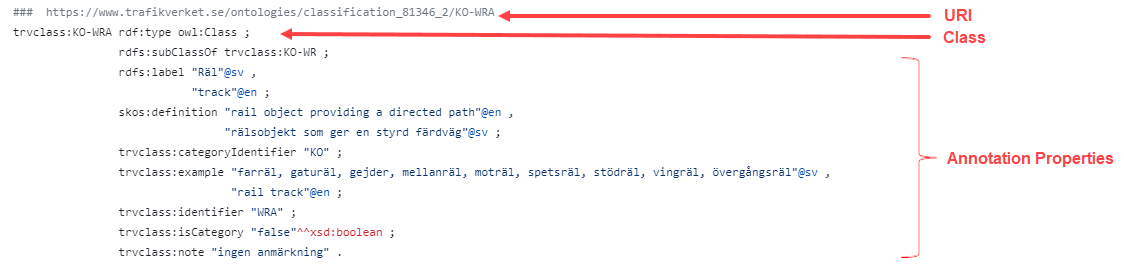


Figure Part of TrvClass ontology expressed in Turtle syntax

The second ontology (TrvKrav) was developed to express the information model described in Figure 23, which links the assets to the requirements. It utilizes concepts and entities implemented in the ontology described in the European standard **EN 17632-1:2022**, titled "Building information modelling (BIM) – Semantic modelling and linking (SML) – Part 1: Generic modelling patterns," as a starting point.

The ontology “TrvKrav”, will contain requirements based on Trafikverket internal document **TRVINFRA-00018- Track components**, published 2020-04-01. The ontology is built on the flowing entities in the flowing table:

Table 3 basic entities used in the ontology TrvKlass

|  |  |  |
| --- | --- | --- |
| Entity | Entity type | URI |
| Creator | Annotation properties | <http://purl.org/dc/elements/1.1/creator> |
| Title | Annotation properties | <http://purl.org/dc/elements/1.1/title> |
| Abstract | Annotation properties | <http://purl.org/dc/terms/abstract> |
| Created | Annotation properties | <http://purl.org/dc/terms/created> |
| Preferred Namespace Prefix | Annotation properties | <http://purl.org/vocab/vann/preferredNamespacePrefix> |
| Preferred Namespace Uri | Annotation properties | <http://purl.org/vocab/vann/preferredNamespaceUri> |
| Definition | Annotation properties | <http://www.w3.org/2004/02/skos/core#definition> |
| Category Identifier | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/categoryIdentifier> |
| Example | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/example> |
| Identifier | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/identifier> |
| Is Category | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/isCategory> |
| Note | Annotation properties | <https://www.trafikverket.se/ontologies/classification_81346_2/note> |
| Has Requirement | Object Properties | <https://www.trafikverket.se/ontologies/trvkrav#hasRequirement> |
| Is Expressed With | Object Properties | <https://www.trafikverket.se/ontologies/trvkrav#isExpressedWith> |
| Graphical | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#Graphical> |
| Identifier | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#Identifier> |
| Motive Text | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#MotiveText> |
| Requirement Notation | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#RequirementNotation> |
| Requirement Text | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#RequirementText> |
| Requirements Properties | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#RequirementsProperties> |
| Textual | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#Textual> |
| Title | Data properties | <https://www.trafikverket.se/ontologies/trvkrav#Title> |
| Asset | Classes | <https://www.trafikverket.se/ontologies/trvkrav#Asset> |
| Requirements | Classes | <https://www.trafikverket.se/ontologies/trvkrav#Requirements> |

*NOTE: the URI variable “trvclass” (*[https://www.trafikverket.se/ontologies/trvkrav#](https://www.trafikverket.se/ontologies/trvkrav)*) is inactive URI, used only as placeholder for the purpose of this project.*



Figure Part of TrvKrav ontology expressed in Turtle syntax

*"NOTE: In the TrvKrav ontology, the class “Asset” serves as the superclass of “Component”, which is equivalent to the class “KO” (component) in the TrvClass ontology.*

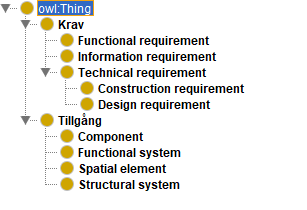


Figure Subclasses of Requirements (Krav) and Assets (Tillgång)

## Material

The resulting material produced during the pilot projects were uploaded to a GitHub repository.

**TrvKrav**

<https://github.com/karrarhm/S2R_TIM/blob/main/ontologies/Requirements/TrvKrav.ttl>

**TrvClass Components**

<https://github.com/karrarhm/S2R_TIM/blob/main/ontologies/classification/81346-KO.ttl>

# Appendix 8.1.1-7: The virtual environment

This appendix will contain a detailed description of the Virtual Environment that was developed in task 8.1.1. Its purpose is to display the Track Information Model along with its governing requirements.

## Framework

The flowing key principles were used in the development of the Virtual Environment:

1. The information has to be easily accessible to different stockholders with different needs and perspectives.
2. The Virtual Environment has to be able to connect the objects inside the Track Information Modell to multiple data sources.

The objective is to develop a virtual dashboard that effectively visualizes the Track Model and its interconnected data, which can originate from multiple sources if necessary.

To achieve the objective of this task and based on approach method described in Appendix 8.1.1-1, ‎3.1, the flowing IT-tools were utilized to create the Virtual Environment:

1. **Speckle**, an IT platform, offers an open-source solution for collaborative design and construction data exchange in the architecture, engineering, and construction (AEC) industry. Its primary goal is to facilitate interoperability and data exchange among various software applications used in the AEC field. The platform enables secure real-time sharing, streaming, and management of 3D models, drawings, and other design-related data. It supports popular file formats like Revit, Rhino, Grasshopper, and more. Users can upload their design data to Speckle and easily collaborate with project stakeholders. Furthermore, Speckle offers APIs and developer tools, empowering users to extend its functionality and integrate it into their workflows and software applications. This flexibility caters to diverse project requirements, enabling customization based on specific needs. (Aec Systems Ltd, 2022)

The platform provides **The Speckle Server**, a central repository for design and construction data in the Speckle platform. It securely stores and manages 3D models, drawings, and other design information. Speckle altos provides **The Speckle Viewer**, a web-based app that lets users visualize and explore 3D models and design data shared through Speckle. It offers a user-friendly interface to view, inspect, and interact with 3D models in browsers. Users can navigate, measure, and analyze models efficiently.

1. **Power BI**, a powerful business intelligence tool developed by Microsoft. It empowers users to analyze and visualize data from multiple sources, offering a user-friendly interface for creating interactive dashboards, reports, and data visualizations. This enables organizations to gain insights and make informed decisions. Power BI's robust data modeling capabilities allow it to connect to various data sources, such as databases, spreadsheets, and cloud services. With intuitive drag-and-drop functionality, users can explore and manipulate data effortlessly. They can also create visually appealing charts, graphs, and maps that are fully customizable. Additionally, Power BI provides advanced features for data querying, transformation, and the creation of calculated measures and columns. Moreover, it facilitates collaboration and sharing, enabling secure publication and distribution of reports and dashboards within an organization or with external stakeholders. ( Wikimedia Foundation Inc, 2022)

## Implementation

The workflow of transforming data of the Track Information Model (IFC-models) to Power BI utilizes Autodesk Navisworks and Speckle Connectors involves several steps. Firstly, the IFC models, which contains detailed information about the Track, is prepared in Autodesk Navisworks, a widely used software for coordination and visualization in the construction industry.

Once the IFC model is ready, the data transformation process begins. Speckle Connectors come into play as they provide a bridge between Navisworks and Power BI, facilitating the transfer of data. Speckle Connectors are plugins or extensions that enable seamless data exchange between different software applications. With the Speckle Connector installed and configured in both Autodesk Navisworks and Power BI, the data export process starts.

The relevant data from the BIM model, such as Track elements, properties, quantities, and metadata, is extracted and converted into a compatible format for Power BI. This ensures that the data retains its structure and integrity during the transfer. Once the data is exported from Autodesk Navisworks, it is imported into Power BI using the Speckle Connector for Power BI. The connector acts as a mediator, facilitating the integration of the IFC data into Power BI's data model.

In Power BI, the imported IFC data can be further refined and transformed to meet specific reporting and visualization requirements. Users can apply data cleansing, filtering, and aggregation techniques to ensure the data is accurate and relevant for analysis and liking. After the data transformation is complete, Power BI's set of tools and features are used to create 3D visualizations, reports, and dashboards.

In Power BI, the query result extracting all requirements from the "TrvKrav" ontology is utilized to generate a table called "Regelverk." This table consists of the following columns:

1. **KravID**: Stores the identifier for the requirements.
2. **KravText**: Stores the textual notation of the requirements.
3. **AssetTypID**: Stores the identifier for associated assets (primary key).
4. Finally, **Grafisk**: Stores the graphical notation of the requirements.

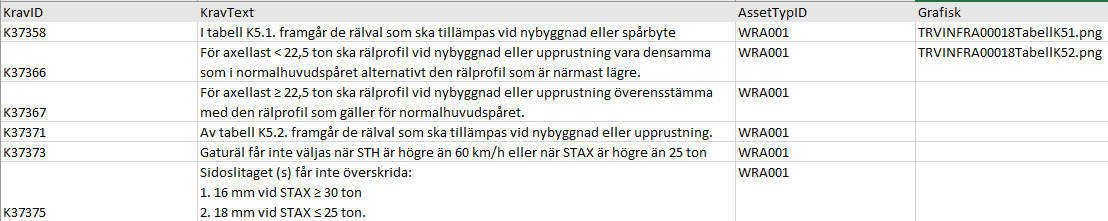


Figure Sample of the query result of the ontology “TrvKrav”

The data from the Speckle server is stored in a "Master" data table, structured in hierarchical levels to correspond with the IFC spatial structure. Subsequently, the data of the IFC models is filtered and grouped based on the asset identifier (TypID), allowing for the linkage of data from the requirements table (Regelverk).

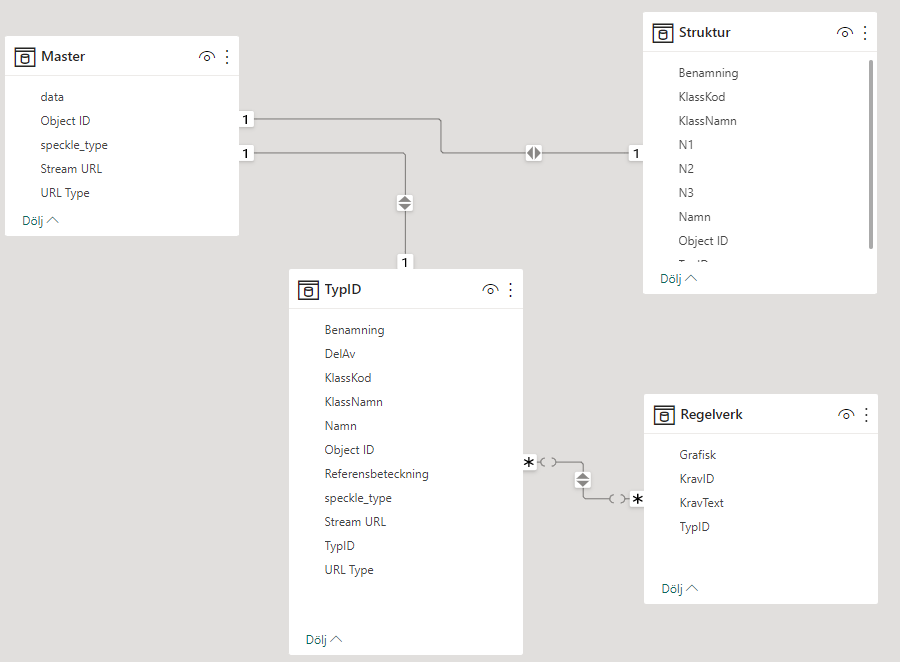


Figure Data model for the tables used in the Virtual Environment using Power BI

The data is streamed directly from the Speckle server to Power BI, allowing continuous updates. This data is used to create an interactive 3D scene in Power BI (refer to number 2 in Figure 30). The scene enables the selection of an asset or group of assets, revealing all the requirements that govern the chosen asset (refer to number 3 and 4 in Figure 30).

In addition, the assets are organized in a hierarchical selection tree that mirrors the structure of an IFC model. This tree can be utilized as an alternative to the 3D scene if required (refer to number 1 in Figure 30).

## Material

The resulting Power BI model is uploaded to a GitHub repository:

<https://github.com/karrarhm/S2R_TIM/blob/main/Virtual%20Environment/S2R_3D_2.pbix>

*NOTE: for the purposes of this task and to achieve an RTL 6 level (Technology demonstrated in relevant environment) the models from the Pilot project East Link were used. The scope was limited to requirement pertains the Rail objects.*

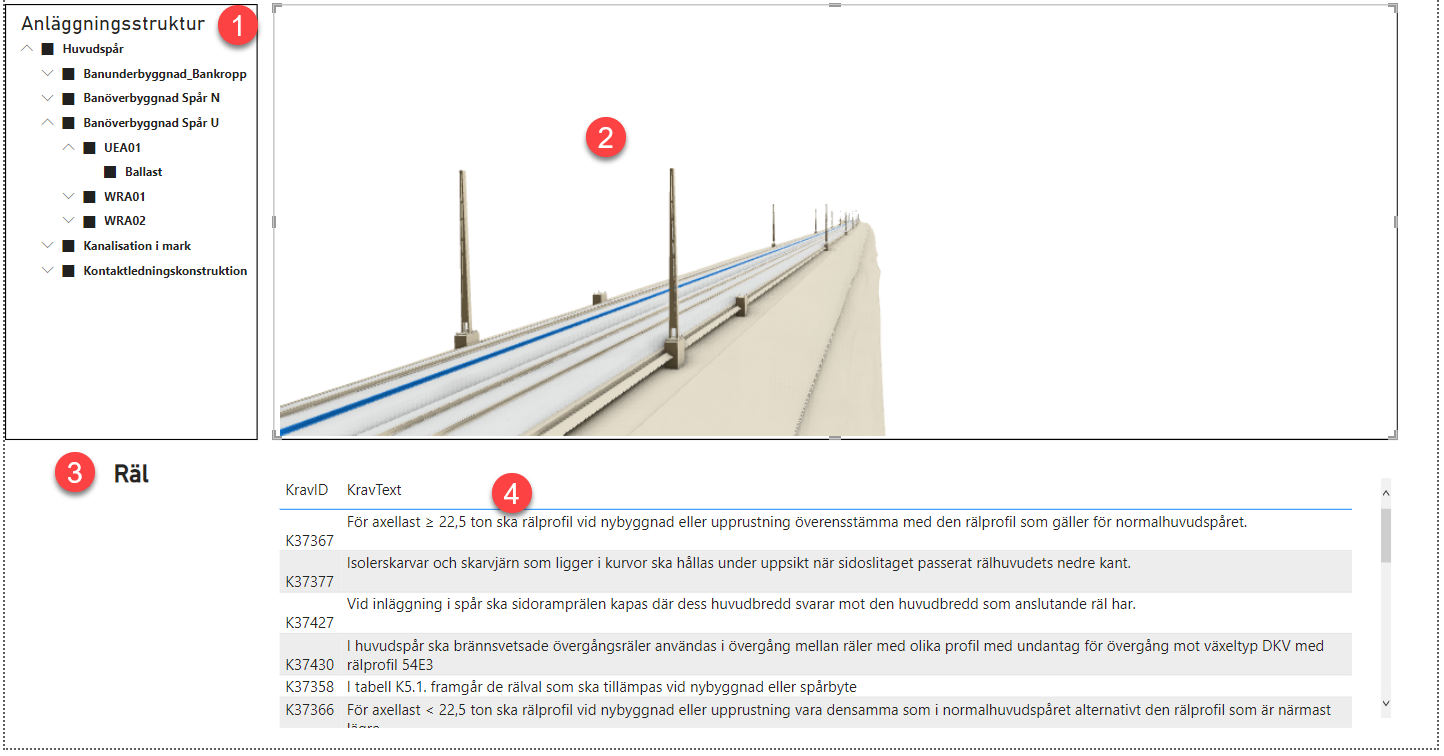


Figure The Virtual Environment Power BI application

# References

Wikimedia Foundation Inc. (2022). *Microsoft Power BI: Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Microsoft\_Power\_BI

Aec Systems Ltd. (2022). *FAQs: Speckle guide*. Retrieved from Speckle guide: https://speckle.guide/user/FAQs.html

Autodesk. (2019). *Linear Structures Workflow Guide.* Autodesk.

Berners-Lee, T. (2006). *W3: Design Issues, Linked Data*. Retrieved from W3: https://www.w3.org/DesignIssues/LinkedData.html

Bhat, A. (2023). *What is Gap Analysis: Definition, Method, and Template: Questionpro, Blog*. Retrieved from Questionpro: https://www.questionpro.com/blog/gap-analysis/

Cagatay, D. (2023). *TRVINFRA-00018 Version 3.0: Puben, Trafikverket*. Retrieved from Trafikverket: https://puben.trafikverket.se/dpub/visa-dokument/6db4169f-b044-4298-adc2-2ebd09e2ff2d

Davidson, M. (2022). *TDOK 2022 0006: Dokumentcenter.* Retrieved from Trafikverket: https://trvdokument.trafikverket.se/Versioner.aspx?spid=6640&dokumentId=TDOK%202022%3a0006

Gruninger, M. U. (1996). *The Knowledge Engineering Review*.

IFC Rail Team. (2022). *IFC Rail Phase 2 Final Report, WP4 – Conceptual Model Report.* buildingSMART International.

Kwon, T. H., Park, S. I., Jang, Y.-H., & Lee, S.-H. (2020). Design of Railway Track Model with Three-Dimensional Alignment Based on Extended Industry Foundation Classes.

Limited, b. I. (2022). Retrieved from IFC4.3.1.0 Documentation : https://ifc43-docs.standards.buildingsmart.org/

Motik, B., Patel-Schneider, P. F., & Parsia, B. (2009, May 31). *OWL 2 Web Ontology Language, Structural Specification and Functional-Style Syntax: W3C*. Retrieved from W3C: https://www.w3.org/2007/OWL/draft/ED-owl2-syntax-20090531/

Serra, P. E., & Hage, S. (2019). *Computational Design for Civil Engineers.* Retrieved from Autodesk.com/autodesk-university: https://www.autodesk.com/autodesk-university/class/Computational-Design-Civil-Engineers-2019#handout

Sternudd, P. (2020, 03 23). *Svensk kravterminologi - Gemensamt språk ger bättre krav!* Retrieved from https://www.kravterminologi.se/: https://www.kravterminologi.se/pdf-version/Kravterminologi.pdf

Svärdby-Bergman, A. &. (n.d.). Virtuell Masteranläggning : ökad förmåga att arbeta modellbaserat i Trafikverke. Trafikverket. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:trafikverket:diva-4502

Swedish Institute for Standards. (2020, 11 23). Building Information Modelling – Level of Information Need – Part 1: Concepts and principles. Swedish Institute for Standards.

Trafikverket. (2021). *Tekniska systemkrav Nya stambanor: Trafikverket.* Retrieved from Trafikverket: https://bransch.trafikverket.se/contentassets/f20ace30749b411392d8393c3e332ef1/tsk-ns-1.0.pdf

W3C. (2015). *Query: W3C, Semantic Web*. Retrieved from W3C, Semantic Web: https://www.w3.org/standards/semanticweb/query.html

W3C. (2015). *Semantic Web: W3C*. Retrieved from The World Wide Web Consortium (W3C): https://www.w3.org/standards/semanticweb/

W3C. (2015). *W3C: Data*. Retrieved from W3C: https://www.w3.org/standards/semanticweb/data.html

W3C. (2016). *W3C wiki: LinkedData*. Retrieved from W3C: https://www.w3.org/wiki/LinkedData