Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für Umwelt, Verkehr, Energie und Kommunikation UVEK
Département fédéral de l'environnement, des transport, de l'énergie et de la communication DETEC
Dipartimento federale dell'ambiente, dei trasporti, dell'energia e delle communication DATEC

Bundesamt für Strassen
Office fédéral des routes
Ufficio federale delle Strade

TUML - Using UML and BPMN to model the Structure, Operation and Maintenance of Road Tunnels. Comprehensive Digitalization of a Tunnel

TUML - Verwendung von UML und BPMN zur Modellierung von Infrastruktur, Betrieb und Instandhaltung von Strassentunneln.

TUML - Utilisation d'UML et de BPMN pour modéliser la structure, l'exploitation et la maintenance des tunnels routiers.

Lombardi SA
Gian Paolo Nodiroli, MSc. Eng. ETH
Melissa Artiglia, MSc. Eng. ETH
Fabio Socchi, B. Eng. SUPSI

Università della Svizzera italiana
Prof. Michele Lanza, REVEAL @ Software Institute, USI Lugano
Davide Paolo Tua, CodeLounge @ Software Institute, USI Lugano

Area dell'esercizio e della manutenzione Cantone Ticino
Cleto Pedrinis, B. Eng.

Research project BGT_20_11A_01 at the request of Bundesamt für Strassen ASTRA
July 2025 | 1798

TUML - Using UML and BPMN to model the Structure, Operation and Maintenance of Road Tunnels. Comprehensive Digitalization of a Tunnel

TUML - Verwendung von UML und BPMN zur Modellierung von Infrastruktur, Betrieb und Instandhaltung von Strassentunneln.

TUML - Utilisation d'UML et de BPMN pour modéliser la structure, l'exploitation et la maintenance des tunnels routiers.

Lombardi SA
Gian Paolo Nodiroli, MSc. Eng. ETH
Melissa Artiglia, MSc. Eng. ETH
Fabio Socchi, B. Eng. SUPSI

Università della Svizzera italiana
Prof. Michele Lanza, REVEAL @ Software Institute, USI Lugano
Davide Paolo Tua, CodeLounge @ Software Institute, USI Lugano

Area dell'esercizio e della manutenzione Cantone Ticino
Cleto Pedrinis, B. Eng.

Research project BGT_20_11A_01 at the request of Bundesamt für Strassen ASTRA
July 2025 | 1798

Imprint

Research centre and project team

Project management
Gian Paolo Nodiroli

Members
Michele Lanza
Cleto Pedrinis
Melissa Artiglia
Davide Paolo Tua
Fabio Socchi

Monitoring committee

President
Roberto Avanti

Members
Matteo Morisoli
Bernard Crausaz
Oliver Demont
Odilo Schoch

Applicant

Bundesamt für Strassen ASTRA

Source

The document can be downloaded free of charge from
https://www.mobilityplatform.ch/

July 2025

Table of contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
AKS-CH	Anlagekennzeichnungssystem Schweiz, Swiss Equipment Coding System
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
AOF	Append Only File
ARM	Advanced RISC Machines
Art.	Article
ASCII	American Standard Code for Information Interchange
ASFINAG	Autobahnen- und Schnellstraßen-Finanzierungs-Aktiengesellschaft
BK	Begleitkommission, monitoring committee
BPD	Business Process Diagram
BPEL	Business Process Execution Language
BPMN	Business Process Model and Notation
BSD	Berkeley Software Distribution
BSAS	Betriebs- und Sicherheitsausrüstungen Sofortlösung
CDR	Reference Code
CBM	Condition-Based Maintenance
CME	Minimum Operational Conditions
CPU	Central Processing Unit
CSV	Comma-Separated Values
E.g.	For Example
Ex.	Example
FA BSA	Fachapplikation Betriebs- und Sicherheitsausrüstungen
FEDRO	Swiss Federal Roads Office
FIWARE	Open-source IoT and smart city platform
GIS	Geographic Information System
GUID	Globally Unique Identifier
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input/Output
IFC	Industry Foundation Classes
IoT	Internet of Things
ISO	International Organization for Standardization
KUBA	Kunstbauten-Datenbank, Structure Database for Road Infrastructure
LRU	Least Recently Used
MDA	Model-Driven Architecture
MDE	Model-Driven Engineering

MIMOSA	Machinery Information Management Open System Alliance
MOF	Meta-Object Facility
MS WORD	Microsoft Word
MQTT	Message Queuing Telemetry Transport
NGSI	Next Generation Service Interface
NGSI-LD	Next Generation Service Interface Linked Data
OCL	Object Constraint Language
OCR	Optical Character Recognition
OMG	Object Management Group
OPC-UA	Open Platform Communications Unified Architecture
PDF	Portable Document Format
PLC	Programmable Logic Controller
PUBACK	Publish Acknowledgment
PUBCOMP	Publish Complete
PUBREC	Publish Received
PUBREL	Publish Release
QoS	Quality of Service
RBBS	FEDRO’s spatial reference system
RCM	Reliability-Centered Maintenance
Re.	Regarding
REST/HTTP	Representational State Transfer Hypertext Transfer Protocol
SCADA	Supervisory Control And Data Acquisition
SCADA/PLC	Integration of SCADA with Programmable Logic Controllers
SISTO	Sicherheitsstollen, Safety Tunnel
SQL	Structured Query Language
SSL	Secure Sockets Layer
STOMP	Simple Text Oriented Messaging Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TRA	Trassee
TUML	Tunnel UML
UML	Unified Modelling Language
UUID	Universally Unique Identifier
Vol.	Volume
WP	Work Package
XML	eXtensible Markup Language

Summary

Introduction and Context

The management of road tunnels is increasingly complex due to factors such as rising traffic volumes, aging infrastructure, and the growing need for enhanced safety and efficiency. As a result, tunnel operators require advanced solutions to ensure uninterrupted functionality. Digitalization presents a transformative approach, leveraging state-of-the-art technologies to optimize tunnel management, improve real-time monitoring, facilitate predictive maintenance, and enhance overall operational control.

Project Objectives

This research investigates how Unified Modelling Language (UML) and Business Process Model and Notation (BPMN) can be applied to improve tunnel management and maintenance. The main objectives include:

- Developing a static digital model of tunnel infrastructure for enhanced visualization and documentation.
- Refining dynamic modelling to optimize workflow efficiency and response time to emergencies.
- Ensuring seamless integration between various tunnel subsystems to facilitate data exchange.
- Automating UML diagram generation to improve documentation and process efficiency.
- Exploring IoT and AI technologies to enhance real-time monitoring and predictive maintenance.

Methodology and Project Phases

The study was conducted in four key phases:

- WP1 - Project Setup and Objective Definition: Identification of appropriate modelling tools and selection of a representative tunnel.
- WP2 - Static Modelling of Tunnel Structure: Development of UML-based structural diagrams to document the tunnel’s physical components and operational subsystems.
- WP3 - Dynamic Modelling with BPMN: Application of BPMN for simulating various operational and maintenance processes.
- WP4 - Future Perspectives and Technological Advancements: Exploration of emerging technologies, including Digital Twins, IoT integration, and AI-based automation.

WP1: Project Setup and Objective Definition

The first work package (WP1) of the research focuses on establishing the foundations for the digitalization of Swiss road tunnels. This phase involves selecting the appropriate modelling tools, defining project objectives, and choosing a representative tunnel to serve as a case study. The primary goal is to ensure that subsequent phases of the project can be executed efficiently with clearly established methodologies and frameworks.

Selection of Modelling Tools
To model tunnel infrastructure and processes effectively, different tools were evaluated for their suitability in static and dynamic modelling. The selection criteria included:
- Scalability: Ability to handle large-scale infrastructure models.
- Interoperability: Compatibility with existing databases and other industry-standard modelling tools.
- User Accessibility: Ease of use for engineers, operators, and other stakeholders.

The study concluded that Unified Modelling Language (UML) and Business Process Model and Notation (BPMN) are the most suitable methodologies for representing tunnel structures and operational workflows, respectively.

Selection of a Representative Tunnel
A crucial aspect of WP1 was identifying a tunnel that meets the necessary complexity and data availability requirements. The Mappo-Morettina Tunnel was selected based on:
- Length and Structural Complexity: A 5.5 km-long tunnel with multiple cross passages.
- Traffic Load and Operational Importance: High daily traffic (26,000 vehicles) requiring continuous monitoring and efficient maintenance strategies.
- Availability of Documentation: Existing data from safety reports, intervention plans, and operational manuals.
- Integration Potential with Digitalization Initiatives: Feasibility of incorporating IoT, Digital Twins, and AI-driven automation.

Establishing the Workflow for Digitalization
With the tunnel selected, a structured workflow was defined for implementing digital modelling techniques. The workflow consists of:
- Data Collection and Structuring: Gathering existing tunnel documentation and organizing it for digital processing.
- Static Modelling: Developing UML diagrams to represent the tunnel's physical and structural components.
- Dynamic Workflow Design: Creating BPMN models to illustrate maintenance, emergency responses, and operational routines.
- Integration with IoT and Middleware: Exploring connectivity solutions, including OPC-UA, MQTT, and real-time sensor data integration.

Challenges Identified in WP1
During the project setup phase, several challenges were identified:
- Heterogeneous Data Formats: Existing tunnel documentation varies in structure and lacks uniformity, requiring significant preprocessing.

- System Compatibility Issues: Legacy infrastructure presents integration difficulties with modern digital tools.
- Stakeholder Coordination: Ensuring alignment between different agencies and technical teams involved in tunnel management.

Conclusion of WP1 and Next Steps

WP1 successfully laid the groundwork for the following project phases by defining objectives, selecting appropriate modelling tools, and identifying the Mappo-Morettina tunnel as a case study. Future steps involve the full-scale implementation of UML-based structural modelling (WP2) and BPMN-driven dynamic process simulations (WP3), with a long-term focus on incorporating AI and IoT technologies for real-time monitoring and predictive maintenance.

WP2: Static Modelling of Tunnel Structure

The second work package (WP2) of the research focuses on the static modelling of tunnel structures using Unified Modelling Language (UML). The goal is to develop a structured digital representation of the tunnel infrastructure to facilitate data integration, visualization, and efficient management of maintenance operations.

UML as a Modelling Tool

UML was chosen as the primary modelling methodology due to its ability to represent complex infrastructure in a clear and standardized manner. The key advantages of UML include:

- Hierarchical Representation: Ability to model tunnels at different levels of granularity.
- Interconnectivity: Facilitates integration with existing databases and maintenance tools.
- Reusability: Enables the use of predefined structures that can be adapted to other tunnels.

Structure and Data Collection

To create an accurate UML model, extensive data collection and structuring were performed. The key sources of data included:

- Geometric Layout and Architectural Data: Tunnel dimensions, cross-sectional structure, and material specifications.
- Electromechanical Systems: Ventilation, lighting, signages, and safety systems.
- Integration with Existing Databases: Utilization of platforms like KUBA and FA BSA to store and retrieve infrastructure data.

UML Diagram Development

The UML modelling process involved the creation of several types of diagrams:

- Class Diagrams: Representing tunnel components, including roadway sections, ventilation shafts, and safety equipment.
- Object Diagrams: Detailing real-world instances of tunnel elements and their relationships.

- Component Diagrams: Illustrating the interaction between different subsystems, such as power supply and traffic monitoring.
- Deployment Diagrams: Mapping out the physical distribution of hardware and sensors within the tunnel.

Each diagram was developed to provide a comprehensive view of the tunnel's structural and functional aspects, ensuring clarity for engineers, maintenance teams, and stakeholders.

Challenges in Static Modelling

Several challenges were encountered during the static modelling process:
- Heterogeneous Data Sources: Data inconsistencies due to different documentation situations across tunnel systems.
- Scalability Issues: The need to create a model that could be expanded or adapted for different tunnel infrastructures.
- Integration with Dynamic Modelling: Ensuring that UML models could be effectively linked to BPMN-based process flows in later phases.

Conclusion of WP2 and Next Steps

WP2 successfully established a structured framework for tunnel digitalization, providing a scalable and reusable UML-based model. The next steps involve transitioning to WP3 - Dynamic Modelling, which will focus on BPMN-based workflows for tunnel operations, maintenance, and emergency response.

WP3: Dynamic Modelling of Operational and Maintenance Processes

The third work package (WP3) focuses on the dynamic modelling of tunnel operations and maintenance workflows using Business Process Model and Notation (BPMN). The goal is to develop a structured representation of processes that ensure operational efficiency, maintenance optimization, and emergency response effectiveness.

BPMN as a Modelling Tool
BPMN was selected as the preferred methodology for modelling dynamic processes due to its advantages:
- Process Standardization: Ensures a unified approach to defining workflows across tunnels.
- Clarity and Simplicity: Provides a visual representation of processes, making them accessible to all stakeholders.
- Automation Readiness: Facilitates integration with IoT-based monitoring and control systems.

Key Process Areas Modelled
WP3 focused on creating BPMN diagrams to model critical tunnel operations, including:
1. Routine Maintenance Procedures: Defining scheduled inspections, equipment servicing, and system diagnostics.

2. Emergency Response Protocols: Mapping workflows for incidents such as fires, vehicle breakdowns, and hazardous material spills.
3. Traffic and Ventilation Management: Optimizing air quality, traffic flow, and energy consumption in real time.
4. Incident Reporting and Resolution: Establishing clear communication channels for reporting and resolving faults.

BPMN Diagram Development

To capture these workflows, various BPMN diagrams were developed, including:

- Process Flow Diagrams: Illustrating step-by-step execution of tunnel operations.
- Decision Gateways: Defining response actions based on different operational conditions.
- Swimlane Diagrams: Representing responsibilities among different entities (e.g., tunnel operators, maintenance teams, emergency services).

Challenges in Dynamic Modelling

Several challenges emerged during WP3:

- Process Complexity: Many tunnel operations involve interdependencies that require careful modelling.
- Integration with Static Models: Ensuring BPMN diagrams align with the UML-based static tunnel models from WP2.
- Data Synchronization: Maintaining real-time accuracy when integrating BPMN workflows with sensor data and IoT systems.

Conclusion of WP3 and Next Steps

WP3 successfully developed a dynamic process framework for tunnel operations, facilitating better decision-making and automation potential. The next phase, WP4 - Future Perspectives, will explore emerging technologies such as AI, Digital Twins, and advanced automation to further enhance tunnel management and efficiency.

WP4: Future Perspectives and Technological Advancements

The fourth work package (WP4) explores future technological developments and innovations that can further enhance tunnel management, safety, and efficiency. This phase focuses on integrating Artificial Intelligence (AI), Digital Twin technology, IoT advancements, and next-generation communication networks to establish a more autonomous and intelligent tunnel infrastructure.

Digital Twin Technology for Tunnel Management

A **Digital Twin** is a virtual model of a physical tunnel that reflects real-time conditions and can be used for simulation, monitoring, and predictive maintenance. Key advantages include:

- **Real-time Data Synchronization**: Ensures that digital representations stay updated with live sensor data.
- **Predictive Maintenance**: Uses AI-driven analytics to anticipate equipment failures before they occur.
- **Simulation and Scenario Planning**: Enables testing of emergency response plans in a virtual environment.

Integration of Artificial Intelligence (AI)

AI-driven tools can revolutionize tunnel operations by automating key tasks and improving decision-making processes. Applications of AI in tunnel management include:

- **Anomaly Detection**: AI-powered algorithms can identify deviations in tunnel conditions, such as temperature spikes or unusual traffic flow patterns.
- **Automated Traffic and Ventilation Control**: AI can dynamically adjust tunnel ventilation and lane usage based on traffic density and air quality.
- **Smart Maintenance Scheduling**: AI optimizes maintenance timelines by analysing historical data and current equipment performance.

IoT and Smart Infrastructure

The Internet of Things (IoT) continues to play a critical role in digital tunnel management. WP4 explores:

- **Sensor Networks**: Integration of smart sensors to monitor air quality, lighting, structural integrity, and traffic conditions in real-time.
- **Edge Computing**: Processing data locally within tunnel control systems to reduce latency and improve response times.
- **Cybersecurity Enhancements**: Protecting tunnel communication systems from cyber threats through encrypted data transmission and AI-driven security measures.

Next-Generation Communication Networks

The emergence of **6G and quantum computing** presents new opportunities for tunnel digitalization. Key benefits include:

- **Ultra-low Latency**: Faster communication between IoT devices and control systems.
- **Quantum-Enhanced Data Processing**: More efficient computation for real-time analytics and decision-making.
- **Improved Connectivity**: Reliable high-speed networks for better integration of tunnel infrastructure with city-wide smart transportation systems.

Challenges and Considerations

Despite the benefits of emerging technologies, several challenges must be addressed:

- **Data Standardization**: Ensuring compatibility between different digital models and existing infrastructure.
- **Cost and Scalability**: Implementing AI, IoT, and Digital Twins requires substantial investment and careful planning for scalability.
- **Regulatory Compliance**: Adapting to evolving safety and cybersecurity regulations to maintain tunnel operation integrity.

Conclusion of WP4 and Future Steps

WP4 highlights the potential of **AI, Digital Twins, IoT, and next-gen communication networks** in revolutionizing tunnel management. Future steps include:

- **Pilot implementation of Digital Twins** for selected tunnels to evaluate real-world benefits.
- **Further development of AI-driven automation tools** for predictive maintenance and traffic optimization.

- **Collaboration with policymakers and industry leaders** to establish standardized frameworks for digital tunnel infrastructure.

By embracing these technological advancements, Swiss tunnels can transition into a fully connected, **intelligent transportation network**, ensuring enhanced safety, operational efficiency, and sustainability for the future.

Zusammenfassung

Einführung und Kontext

Das Management von Strassentunneln wird aufgrund von Faktoren wie dem steigenden Verkehrsaufkommen, der alternden Infrastruktur und dem wachsenden Bedarf an mehr Sicherheit und Effizienz immer komplexer. Infolgedessen benötigen Tunnelbetreiber fortschrittliche Lösungen, um eine ununterbrochene Funktionalität zu gewährleisten. Die Digitalisierung stellt einen transformativen Ansatz dar, bei dem modernste Technologien eingesetzt werden, um das Tunnelmanagement zu optimieren, die Echtzeitüberwachung zu verbessern, die vorausschauende Wartung zu erleichtern und die allgemeine Betriebskontrolle zu verbessern.

Ziele des Projekts

In dieser Forschungsarbeit wird untersucht, wie die Unified Modelling Language (UML) und die Business Process Model and Notation (BPMN) zur Verbesserung des Tunnelmanagements und der Wartung eingesetzt werden können. Zu den Hauptzielen gehören:

- Entwicklung eines statischen digitalen Modells der Tunnelinfrastruktur zur besseren Visualisierung und Dokumentation.
- Verfeinerung der dynamischen Modellierung zur Optimierung der Effizienz der Arbeitsabläufe und der Reaktionszeit bei Notfällen.
- Gewährleistung einer nahtlosen Integration zwischen den verschiedenen Teilsystemen des Tunnels, um den Datenaustausch zu erleichtern.
- Automatisierung der UML-Diagrammerstellung zur Verbesserung der Dokumentation und der Prozesseffizienz.
- Erforschung von IoT- und KI-Technologien zur Verbesserung der Echtzeitüberwachung und vorausschauenden Wartung.

Methodik und Projektphasen

Die Studie wurde in vier Hauptphasen durchgeführt:

- WP1 - Projektaufbau und Zieldefinition: Identifizierung geeigneter Modellierungswerkzeuge und Auswahl eines repräsentativen Tunnels.
- WP2 - Statische Modellierung der Tunnelstruktur: Entwicklung von UML-basierten Strukturdiagrammen zur Dokumentation der physikalischen Komponenten und betrieblichen Subsysteme des Tunnels.
- WP3 - Dynamische Modellierung mit BPMN: Anwendung von BPMN zur Simulation verschiedener Betriebs- und Wartungsprozesse.
- WP4 - Zukunftsperspektiven und technologischer Fortschritt: Erforschung neuer Technologien, einschliesslich digitaler Zwillinge, IoT-Integration und KI-basierter Automatisierung.

WP1: Projektaufbau und Zieldefinition

Das erste Arbeitspaket (WP1) der Forschung fokussiert sich auf die Entwicklung grundlegender Konzepte und Rahmenbedingungen konzentriert sich auf die Schaffung der Grundlagen für die Digitalisierung von Schweizer Strassentunneln. In dieser Phase geht es um die Auswahl der geeigneten Modellierungswerkzeuge, die Definition der Projektziele und die Auswahl eines repräsentativen Tunnels, der als Fallstudie dienen soll. Damit soll sichergestellt werden, dass die nachfolgenden Phasen des Projekts mit klar definierten Methoden und Rahmenbedingungen effizient durchgeführt werden können.

Auswahl der Modellierungswerkzeuge

Zur effektiven Modellierung von Tunnelinfrastruktur und -prozessen wurden verschiedene Tools auf ihre Eignung für die statische und dynamische Modellierung geprüft. Zu den Auswahlkriterien gehörten:

- Skalierbarkeit: Fähigkeit zur Handhabung ok umfangreicher Infrastrukturmodelle.
- Interoperabilität: Kompatibilität mit bestehenden Datenbanken und anderen branchenüblichen Modellierungswerkzeugen.
- Benutzerfreundlichkeit: Benutzerfreundlichkeit für Ingenieure, Bediener und andere Beteiligte.

Die Studie kam zu dem Schluss, dass die Unified Modelling Language (UML) und die Business Process Model and Notation (BPMN) die am besten geeigneten Methoden zur Darstellung von Tunnelstrukturen bzw. betrieblichen Abläufen sind.

Auswahl eines repräsentativen Tunnels

Ein entscheidender Aspekt von WP1 war die Auswahl eines Tunnels, der den Anforderungen an Komplexität und Datenverfügbarkeit entspricht. Der Mappo-Morettina-Tunnel wurde aus folgenden Gründen ausgewählt:

- Länge und strukturelle Komplexität: Ein 5,5 km langer Tunnel mit mehreren Fluchtwegen innerhalb der Tunnel.
- Verkehrsbelastung und betriebliche Bedeutung: Hohes tägliches Verkehrsaufkommen (26.000 Fahrzeuge), das eine kontinuierliche Überwachung und effiziente Wartungsstrategien erfordert.
- Verfügbarkeit von Dokumentation: Vorhandene Daten aus Sicherheitsberichten, Einsatzplänen und Anlagendokumentationen.
- Integrationspotenzial mit Digitalisierungsinitiativen: Durchführbarkeit der Integration von IoT, Digital Twins und KI-gesteuerter Automatisierung.

Aufbau des Workflows für die Digitalisierung

Anhand des ausgewählten Tunnels wurde ein strukturierter Arbeitsablauf für die Umsetzung der digitalen Modellierungstechniken festgelegt. Der Arbeitsablauf besteht aus:

- Datenerfassung und -strukturierung: Erfassung der vorhandenen Tunneldokumentation und deren Organisation für die digitale Verarbeitung.
- Statische Modellierung: Entwicklung von UML-Diagrammen zur Darstellung der physikalischen und strukturellen Komponenten des Tunnels.
- Dynamischer Workflow-Entwurf: Erstellung von BPMN-Modellen zur Veranschaulichung von Wartungs-, Notfall- und Betriebsroutinen.

- Integration mit IoT und Middleware: Erforschung von Konnektivitätslösungen, einschliesslich OPC-UA, MQTT und Echtzeit-Sensordatenintegration.

In WP1 identifizierte Herausforderungen
Während der Projektaufbauphase wurden mehrere Herausforderungen festgestellt:

- Heterogene Datenformate: Die vorhandene Tunneldokumentation ist unterschiedlich strukturiert und uneinheitlich, was eine umfangreiche Vorverarbeitung erfordert.
- Probleme mit der Systemkompatibilität: Veraltete Infrastrukturen bereiten Schwierigkeiten bei der Integration mit modernen digitalen Werkzeugen.
- Koordinierung der Interessengruppen: Sicherstellung der Abstimmung zwischen verschiedenen Behörden und technischen Teams, die am Tunnelmanagement beteiligt sind.

Abschluss von WP1 und nächste Schritte
WP1 legte erfolgreich den Grundstein für die folgenden Projektphasen, indem Ziele definiert, geeignete Modellierungswerkzeuge ausgewählt und der Mappo-Morettina-Tunnel als Fallstudie identifiziert wurde. Künftige Schritte umfassen die vollständige Implementierung von UML-basierter Strukturmodellierung (WP2) und BPMN-gesteuerten dynamischen Prozesssimulationen (WP3), wobei der langfristige Fokus auf der Einbeziehung von KI- und IoT-Technologien für Echtzeitüberwachung und prädiktive Wartung liegt.

WP2: Statische Modellierung der Tunnelstruktur

Das zweite Arbeitspaket (WP2) der Forschung konzentriert sich auf die statische Modellierung von Tunnelbauwerken mit Hilfe der Unified Modeling Language (UML). Ziel ist es, eine strukturierte digitale Darstellung der Tunnelinfrastruktur zu entwickeln, um die Datenintegration, die Visualisierung und die effiziente Verwaltung von Wartungsarbeiten zu erleichtern.

UML als Modellierungswerkzeug
UML wurde als primäre Modellierungsmethodik gewählt, da sie komplexe Infrastrukturen auf klare und standardisierte Weise darstellen kann. Zu den wichtigsten Vorteilen von UML gehören:

- Hierarchische Darstellung: Möglichkeit der Modellierung von Tunneln auf verschiedenen Ebenen der Granularität.
- Interkonnektivität: Erleichtert die Integration mit bestehenden Datenbanken und Wartungstools.
- Wiederverwendbarkeit: Ermöglicht die Verwendung von vordefinierten Strukturen, die an andere Tunnel angepasst werden können.

Struktur und Datenerhebung
Um ein genaues UML-Modell zu erstellen, wurden umfangreiche Datenerhebungen und -strukturierungen durchgeführt. Zu den wichtigsten Datenquellen gehörten:

- Geometrisches Layout und architektonische Daten: Tunnelabmessungen, Querschnittsstruktur und Materialspezifikationen.
- Elektromechanische Systeme: Lüftung, Beleuchtung, Signalisation und Sicherheitssysteme.

- Integration mit bestehenden Datenbanken: Nutzung von Plattformen wie KUBA und FA BSA zum Speichern und Abrufen von Infrastrukturdaten.

UML-Diagramm-Entwicklung
Der UML-Modellierungsprozess umfasst die Erstellung verschiedener Diagrammtypen:
- Klassendiagramme: Darstellung von Tunnelkomponenten, einschliesslich Fahrbahnabschnitten, Lüftungsschächten und Sicherheitseinrichtungen.
- Objektdiagramme: Detaillierte Darstellung realer Instanzen von Tunnelelementen und ihrer Beziehungen.
- Komponentendiagramme: Veranschaulichung der Interaktion zwischen verschiedenen Teilsystemen, z. B. Stromversorgung und Verkehrsüberwachung.
- Verteilungsdiagramme: Darstellung der physischen Verteilung von Hardware und Sensoren innerhalb des Tunnels.

Jedes Diagramm wurde entwickelt, um einen umfassenden Überblick über die strukturellen und funktionalen Aspekte des Tunnels zu geben und so die Klarheit für Ingenieure, Wartungsteams und Beteiligte zu gewährleisten.

Herausforderungen bei der statischen Modellierung
Bei der statischen Modellierung traten mehrere Herausforderungen auf:
- Heterogene Datenquellen: Dateninkonsistenzen aufgrund unterschiedlicher Dokumentationszustände in verschiedenen Tunnelsystemen.
- Fragen der Skalierbarkeit: Die Notwendigkeit, ein Modell zu schaffen, das für verschiedene Tunnelinfrastrukturen erweitert oder angepasst werden kann.
- Integration mit dynamischer Modellierung: Sicherstellung, dass UML-Modelle in späteren Phasen effektiv mit BPMN-basierten Prozessabläufen verknüpft werden können.

Abschluss von WP2 und nächste Schritte
WP2 hat erfolgreich einen strukturierten Rahmen für die Digitalisierung von Tunneln geschaffen, der ein skalierbares und wiederverwendbares UML-basiertes Modell liefert. Die nächsten Schritte umfassen den Übergang zu WP3 - Dynamische Modellierung, das sich auf BPMN-basierte Arbeitsabläufe für den Tunnelbetrieb, die Wartung und die Notfallmassnahmen konzentrieren wird.

WP3: Dynamische Modellierung von Betriebs- und Instandhaltungsabläufen

Das dritte Arbeitspaket (WP3) konzentriert sich auf die dynamische Modellierung von Tunnelbetriebs- und Wartungsabläufen unter Verwendung der Business Process Model and Notation (BPMN). Ziel ist es, eine strukturierte Darstellung von Prozessen zu entwickeln, die die betriebliche Effizienz, die Optimierung der Instandhaltung und die Wirksamkeit der Notfallmassnahmen gewährleisten.

BPMN als Modellierungswerkzeug
Die BPMN wurde aufgrund ihrer Vorteile als bevorzugte Methode für die Modellierung dynamischer Prozesse ausgewählt:
- Standardisierung von Prozessen: Sorgt für einen einheitlichen Ansatz bei der Definition von Arbeitsabläufen in allen Tunneln.

- Klarheit und Einfachheit: Bietet eine visuelle Darstellung von Prozessen und macht sie für alle Beteiligten zugänglich.
- Bereitschaft zur Automatisierung: Erleichtert die Integration mit IoT-basierten Überwachungs- und Steuerungssystemen.

Modellierte Hauptprozessbereiche

WP3 konzentrierte sich auf die Erstellung von BPMN-Diagrammen zur Modellierung kritischer Tunnelbetriebe, unter anderem:

5. Routinemässige Wartungsverfahren: Festlegung geplanter Inspektionen, Wartung von Geräten und Systemdiagnose.
6. Protokolle für Notfallmassnahmen: Kartierung von Arbeitsabläufen für Vorfälle wie Brände, Fahrzeugpannen und Gefahrgutunfälle.
7. Verkehrs- und Lüftungsmanagement: Optimierung der Luftqualität, des Verkehrsflusses und des Energieverbrauchs in Echtzeit.
8. Meldung und Lösung von Vorfällen: Einrichtung klarer Kommunikationskanäle für die Meldung und Behebung von Fehlern.

Entwicklung von BPMN-Diagrammen

Um diese Arbeitsabläufe zu erfassen, wurden verschiedene BPMN-Diagramme entwickelt, darunter:

- Prozessablaufdiagramme: Veranschaulichung der schrittweisen Ausführung von Tunneloperationen.
- Entscheidungs-Gateways: Festlegung von Reaktionsmassnahmen auf der Grundlage verschiedener Betriebsbedingungen.
- Swimlane-Diagramme: Darstellung der Zuständigkeiten zwischen verschiedenen Stellen (z. B. Tunnelbetreiber, Instandhaltungsteams, Rettungsdienste).

Herausforderungen bei der dynamischen Modellierung

In WP3 wurden mehrere Herausforderungen deutlich:

- Komplexität der Prozesse: Viele Tunnelbetriebe sind mit Abhängigkeiten verbunden, die eine sorgfältige Modellierung erfordern.
- Integration mit statischen Modellen: Sicherstellen, dass BPMN-Diagramme mit den UML-basierten statischen Tunnelmodellen aus WP2 übereinstimmen.
- Daten-Synchronisierung: Aufrechterhaltung der Echtzeitgenauigkeit bei der Integration von BPMN-Workflows mit Sensordaten und IoT-Systemen.

Abschluss von WP3 und nächste Schritte

WP3 entwickelte erfolgreich einen dynamischen Prozessrahmen für den Tunnelbetrieb, der eine bessere Entscheidungsfindung und ein höheres Automatisierungspotenzial ermöglicht. In der nächsten Phase, WP4 - Zukunftsperspektiven, werden neue Technologien wie KI, digitale Zwillinge und fortschrittliche Automatisierung erforscht, um das Tunnelmanagement und die Effizienz weiter zu verbessern.

WP4: Zukunftsperspektiven und technologischer Fortschritt

Das vierte Arbeitspaket (WP4) erforscht zukünftige technologische Entwicklungen und Innovationen, die das Tunnelmanagement, die Sicherheit und die Effizienz weiter verbessern können. Diese Phase konzentriert sich auf die Integration von künstlicher

Intelligenz (KI), digitaler Zwillingstechnologie, IoT-Fortschritten und Kommunikationsnetzen der nächsten Generation, um eine autonomere und intelligentere Tunnelinfrastruktur zu schaffen.

Digitale Zwillingstechnologie für das Tunnelmanagement
Ein **Digitaler Zwilling** ist ein virtuelles Modell eines physischen Tunnels, das Echtzeitbedingungen widerspiegelt und für Simulation, Überwachung und prädiktive Wartung verwendet werden kann. Die wichtigsten Vorteile sind:

- **Datensynchronisierung in Echtzeit**: Stellt sicher, dass die digitalen Darstellungen mit den aktuellen Sensordaten aktualisiert werden.
- **Vorausschauende Wartung**: Nutzt KI-gesteuerte Analysen, um Geräteausfälle vorherzusehen, bevor sie auftreten.
- **Simulation und Szenarienplanung**: Ermöglicht das Testen von Notfalleinsatzplänen in einer virtuellen Umgebung.

Integration von Künstlicher Intelligenz (KI)
KI-gesteuerte Tools können den Tunnelbetrieb revolutionieren, indem sie wichtige Aufgaben automatisieren und Entscheidungsprozesse verbessern. Zu den Anwendungen von KI im Tunnelmanagement gehören:

- **Erkennung von Anomalien**: KI-gestützte Algorithmen können Abweichungen von den Tunnelbedingungen erkennen, z. B. Temperaturspitzen oder ungewöhnliche Verkehrsflussmuster.
- **Automatisierte Verkehrs- und Lüftungssteuerung**: Die künstliche Intelligenz kann die Tunnelbelüftung und die Nutzung der Fahrspuren je nach Verkehrsdichte und Luftqualität dynamisch anpassen.
- **Intelligente Planung der Wartung**: KI optimiert die Wartungszeitpläne durch die Analyse historischer Daten und der aktuellen Geräteleistung.

IoT und intelligente Infrastruktur
Das Internet der Dinge (IoT) spielt weiterhin eine entscheidende Rolle beim digitalen Tunnelmanagement. WP4 erforscht:

- **Sensornetzwerke**: Integration intelligenter Sensoren zur Überwachung von Luftqualität, Beleuchtung, struktureller Integrität und Verkehrsbedingungen in Echtzeit.
- **Edge Computing**: Lokale Datenverarbeitung innerhalb von Tunnelsteuerungssystemen, um Latenzzeiten zu verringern und die Reaktionszeiten zu verbessern.
- **Verbesserungen bei der Cybersicherheit**: Schutz der Tunnelkommunikationssysteme vor Cyberbedrohungen durch verschlüsselte Datenübertragung und KI-gesteuerte Sicherheitsmassnahmen.

Kommunikationsnetze der nächsten Generation
Das Aufkommen von **6G und Quantencomputing** eröffnet neue Möglichkeiten für die Digitalisierung von Tunneln. Die wichtigsten Vorteile sind:

- **Ultra-niedrige Latenzzeit**: Schnellere Kommunikation zwischen IoT-Geräten und Steuerungssystemen.
- **Quantengestützte Datenverarbeitung**: Effizientere Berechnungen für Echtzeit-Analysen und Entscheidungsfindung.

- **Verbesserte Konnektivität**: Zuverlässige Hochgeschwindigkeitsnetze für eine bessere Integration der Tunnelinfrastruktur in stadtweite intelligente Verkehrssysteme.

Herausforderungen und Überlegungen

Trotz der Vorteile der neuen Technologien müssen mehrere Herausforderungen bewältigt werden:

- **Standardisierung von Daten**: Gewährleistung der Kompatibilität zwischen verschiedenen digitalen Modellen und der bestehenden Infrastruktur.
- **Kosten und Skalierbarkeit**: Die Implementierung von KI, IoT und digitalen Zwillingen erfordert erhebliche Investitionen und eine sorgfältige Planung der Skalierbarkeit.
- **Einhaltung gesetzlicher Vorschriften**: Anpassung an sich entwickelnde Sicherheits- und Cybersicherheitsvorschriften zur Aufrechterhaltung der Integrität des Tunnelbetriebs.

Abschluss von WP4 und zukünftige Schritte

WP4 widmet sich eingehend dem Potential dem beleuchtet das Potenzial von **KI, digitalen Zwillingen, IoT und Kommunikationsnetzen der nächsten Generation** für die Revolutionierung des Tunnelmanagements. Künftige Schritte umfassen:

- **Pilotimplementierung von Digitalen Zwillingen** für ausgewählte Tunnel, um die Vorteile in der Praxis zu bewerten.
- **Weiterentwicklung von KI-gesteuerten Automatisierungswerkzeugen** für vorausschauende Wartung und Verkehrsoptimierung.
- **Zusammenarbeit mit politischen Entscheidungsträgern und Branchenführern**, um standardisierte Rahmenbedingungen für die digitale Tunnelinfrastruktur zu schaffen.

Durch die Nutzung dieser technologischen Fortschritte können die Schweizer Tunnel zu einem vollständig vernetzten, **intelligenten Verkehrsnetz** werden, das mehr Sicherheit, betriebliche Effizienz und Nachhaltigkeit für die Zukunft gewährleistet.

Résumé

Introduction et contexte

La gestion des tunnels routiers est de plus en plus complexe en raison de facteurs tels que l'augmentation des volumes de trafic, le vieillissement de l'infrastructure et le besoin croissant d'une sécurité et d'une efficacité accrues. Par conséquent, les exploitants de tunnels ont besoin de solutions avancées pour garantir une fonctionnalité ininterrompue. La digitalisation présente une approche transformatrice, tirant parti des technologies de pointe pour optimiser la gestion des tunnels, améliorer la surveillance en temps réel, faciliter la maintenance prédictive et renforcer le contrôle global de l'exploitation.

Objectifs du projet

Cette recherche étudie comment le langage de modélisation unifié (UML) et le modèle et la notation des processus d'entreprise (BPMN) peuvent être appliqués pour améliorer la gestion et l'entretien des tunnels. Les principaux objectifs sont les suivants :

- Développement d'un modèle numérique statique de l'infrastructure des tunnels pour améliorer la visualisation et la documentation.
- Affiner la modélisation dynamique pour optimiser l'efficacité du flux de travail et le temps de réponse aux urgences.
- Assurer une intégration transparente entre les différents sous-systèmes du tunnel afin de faciliter l'échange de données.
- Automatiser la génération de diagrammes UML pour améliorer la documentation et l'efficacité des processus.
- Explorer les technologies de l'IoT et de l'IA pour améliorer la surveillance en temps réel et la maintenance prédictive.

Méthodologie et phases du projet

L'étude a été réalisée en quatre phases principales :

- WP1 - Mise en place du projet et définition des objectifs : Identification des outils de modélisation appropriés et sélection d'un tunnel représentatif.
- WP2 - Modélisation statique de la structure du tunnel : Développement de diagrammes structurels basés sur UML pour documenter les composants physiques et les sous-systèmes opérationnels du tunnel.
- WP3 - Modélisation dynamique avec BPMN : Application de BPMN pour la simulation de divers processus opérationnels et de maintenance.
- WP4 - Perspectives d'avenir et avancées technologiques : Exploration des technologies émergentes, y compris les jumeaux numériques, l'intégration de l'IoT et l'automatisation basée sur l'IA.

WP1 : Mise en place du projet et définition des objectifs

Le premier work package (WP1) de la recherche se concentre sur l'établissement des bases de la digitalisation des tunnels routiers suisses. Cette phase implique la sélection des outils de modélisation appropriés, la définition des objectifs du projet et le choix d'un tunnel représentatif qui servira comme model. L'objectif principal est de s'assurer que les phases suivantes du projet peuvent être exécutées efficacement avec des méthodologies et des cadres clairement établis.

Sélection des outils de modélisation
Pour modéliser efficacement l'infrastructure et les processus des tunnels, différents outils ont été évalués en fonction de leur aptitude à la modélisation statique et dynamique. Les critères de sélection comprenaient

- Évolutivité : Capacité à gérer des modèles d'infrastructure à grande échelle.
- Interopérabilité : Compatibilité avec les bases de données existantes et les autres outils de modélisation standard de l'industrie.
- Accessibilité pour l'utilisateur : Facilité d'utilisation pour les ingénieurs, les opérateurs et les autres parties prenantes.

L'étude a conclu que le langage de modélisation unifié (UML) et le Business Process Model and Notation (BPMN) sont les méthodologies les plus appropriées pour représenter les structures des tunnels et les flux de travail opérationnels, respectivement.

Sélection d'un tunnel représentatif
Un aspect crucial du WP1 a été l'identification d'un tunnel répondant aux exigences de complexité et de disponibilité des données. Le tunnel de Mappo-Morettina a été sélectionné pour les raisons suivantes :

- Longueur et complexité de la structure : Tunnel de 5,5 km de long avec passages transversaux (voies de fuite).
- Charge de trafic et importance opérationnelle : Trafic journalier élevé (26 000 véhicules) nécessitant une surveillance continue et des stratégies d'entretien efficaces.
- Disponibilité de la documentation : Données existantes provenant de rapports de sécurité, de plans d'intervention et de manuels opérationnels.
- Potentiel d'intégration avec les initiatives de digitalisation : Faisabilité de l'intégration de l'IoT, des jumeaux numériques et de l'automatisation pilotée par l'IA.

Établir le flux de travail pour la digitalisation
Une fois le tunnel sélectionné, un processus structuré a été défini pour mettre en œuvre les techniques de modélisation numérique. Le flux de travail consiste en

- Collecte et structuration des données : Rassembler la documentation existante sur les tunnels et l'organiser en vue d'un traitement numérique.
- Modélisation statique : Développement de diagrammes UML pour représenter les composants physiques et structurels du tunnel.
- Conception de flux de travail dynamiques : Création de modèles BPMN pour illustrer la maintenance, les réponses d'urgence et les routines opérationnelles.
- Intégration avec l'IoT et les intergiciels : Exploration des solutions de connectivité, y compris OPC-UA, MQTT et l'intégration de données de capteurs en temps réel.

Défis identifiés dans le WP1
Au cours de la phase de mise en place du projet, plusieurs défis ont été identifiés :
- Formats de données hétérogènes : La documentation existante sur les tunnels varie dans sa structure et manque d'uniformité, ce qui nécessite un prétraitement important.
- Problèmes de compatibilité des systèmes : L'infrastructure existante présente des difficultés d'intégration avec les outils numériques modernes.
- Coordination des parties prenantes : Assurer l'alignement entre les différentes agences et équipes techniques impliquées dans la gestion du tunnel.

Conclusion du WP1 et prochaines étapes
Le WP1 a posé avec succès les bases des phases suivantes du projet en définissant les objectifs, en sélectionnant les outils de modélisation appropriés et en identifiant le tunnel de Mappo-Morettina comme étude de cas. Les étapes futures impliquent la mise en œuvre à grande échelle de la modélisation structurelle basée sur UML (WP2) et des simulations de processus dynamiques pilotées par BPMN (WP3), avec un objectif à long terme sur l'incorporation de l'IA et des technologies IoT pour la surveillance en temps réel et la maintenance prédictive.

WP2 : Modélisation statique de la structure du tunnel

Le deuxième work package (WP2) de la recherche se concentre sur la modélisation statique des structures des tunnels à l'aide du langage de modélisation unifié (UML). L'objectif est de développer une représentation numérique structurée de l'infrastructure du tunnel afin de faciliter l'intégration des données, la visualisation et la gestion efficace des opérations d'entretien.

UML en tant qu'outil de modélisation

UML a été choisi comme principale méthode de modélisation en raison de sa capacité à représenter une infrastructure complexe de manière claire et normalisée. Les principaux avantages d'UML sont les suivants
- Représentation hiérarchique : Possibilité de modéliser les tunnels à différents niveaux de granularité.
- Interconnectivité : Facilite l'intégration avec les bases de données et les outils de maintenance existants.
- Réutilisation : Permet l'utilisation de structures prédéfinies qui peuvent être adaptées à d'autres tunnels.

Structure et collecte de données

Pour créer un modèle UML précis, une collecte et une structuration approfondies des données ont été effectuées. Les principales sources de données sont les suivantes
- Disposition géométrique et données structurelles : Dimensions du tunnel, structure de la section transversale et spécifications des matériaux.
- Systèmes électromécaniques : Systèmes de ventilation, d'éclairage, de signalisation et de sécurité.
- Intégration avec les bases de données existantes : Utilisation de plateformes telles que KUBA et FA BSA pour stocker et récupérer les données relatives à l'infrastructure.

Développement de diagrammes UML

Le processus de modélisation UML implique la création de plusieurs types de diagrammes :

- Diagrammes de classe : Représentation des composants du tunnel, y compris les tronçons de route, les puits de ventilation et les équipements de sécurité.
- Diagrammes d'objets : Détailler les instances réelles des éléments du tunnel et leurs relations.
- Diagrammes des composants : Illustration de l'interaction entre différents sous-systèmes, tels que l'alimentation électrique et la surveillance du trafic.
- Diagrammes de déploiement : La cartographie de la distribution physique du matériel et des capteurs à l'intérieur du tunnel.

Chaque diagramme a été conçu pour fournir une vue d'ensemble des aspects structurels et fonctionnels du tunnel, afin d'assurer la clarté pour les ingénieurs, les équipes d'entretien et les parties prenantes.

Les défis de la modélisation statique

Plusieurs difficultés ont été rencontrées au cours du processus de modélisation statique :

- Sources de données hétérogènes : Incohérences des données dues à des états de documentation différentes entre les systèmes de tunnels.
- Problèmes d'évolutivité : Nécessité de créer un modèle qui puisse être étendu ou adapté à différentes infrastructures de tunnel.
- Intégration avec la modélisation dynamique : Veiller à ce que les modèles UML puissent être reliés efficacement aux flux de processus basés sur BPMN dans les phases ultérieures.

Conclusion du WP2 et prochaines étapes

Le WP2 a réussi à établir un cadre structuré pour la digitalisation des tunnels, en fournissant un modèle évolutif et réutilisable basé sur UML. Les prochaines étapes comprennent la transition vers le WP3 - Modélisation dynamique, qui se concentrera sur les flux de travail basés sur BPMN pour l'exploitation des tunnels, la maintenance et les interventions d'urgence.

WP3 : Modélisation dynamique des processus opérationnels et de maintenance

Le troisième work package (WP3) se concentre sur la modélisation dynamique de l'exploitation des tunnels et des flux de travaux d'entretien à l'aide du Business Process Model and Notation (BPMN). L'objectif est de développer une représentation structurée des processus qui assurent l'efficacité de l'exploitation, l'optimisation de la maintenance et l'efficacité des interventions d'urgence.

BPMN en tant qu'outil de modélisation

BPMN a été choisi comme méthodologie privilégiée pour modéliser les processus dynamiques en raison de ses avantages :

- Normalisation des processus : Assure une approche unifiée de la définition des flux de travail dans les tunnels.
- Clarté et simplicité : Fournit une représentation visuelle des processus, les rendant accessibles à toutes les parties prenantes.
- Préparation à l'automatisation : Facilite l'intégration avec les systèmes de surveillance et de contrôle basés sur l'IoT.

Principaux domaines de processus modélisés

Le WP3 s'est concentré sur la création de diagrammes BPMN pour modéliser les opérations critiques des tunnels, notamment

9. Procédures d'entretien de routine : Définition des inspections programmées, de l'entretien des équipements et du diagnostic des systèmes.
10. Protocoles d'intervention en cas d'urgence : Cartographie des flux de travail pour les incidents tels que les incendies, les pannes de véhicules et les déversements de matières dangereuses.
11. Gestion du trafic et de la ventilation : Optimisation de la qualité de l'air, de la circulation et de la consommation d'énergie en temps réel.
12. Indication et résolution des incidents : Mise en place de canaux de communication clairs pour la signalisation et la résolution des incidents.

Développement de diagrammes BPMN

Pour saisir ces flux de travail, divers diagrammes BPMN ont été élaborés :

- Diagrammes de flux de processus : Illustration de l'exécution étape par étape des opérations dans le tunnel.
- Passerelles de décision : Définition d'actions de réponse basées sur différentes conditions opérationnelles.
- Diagrammes Swim-lane : Représentation des responsabilités entre différentes entités (par exemple, exploitants de tunnels, équipes d'entretien, services d'urgence).

Les défis de la modélisation dynamique

Plusieurs défis sont apparus au cours du WP3 :

- Complexité du processus : De nombreuses opérations dans les tunnels impliquent des interdépendances qui nécessitent une modélisation minutieuse.
- Intégration avec les modèles statiques : Assurer l'alignement des diagrammes BPMN avec les modèles statiques de tunnel basés sur UML du WP2.
- Synchronisation des données : Maintenir la précision en temps réel lors de l'intégration de flux de travail BPMN avec des données de capteurs et des systèmes IoT.

Conclusion du WP3 et prochaines étapes

Le WP3 a développé avec succès un cadre de processus dynamique pour l'exploitation des tunnels, facilitant une meilleure prise de décision et un meilleur potentiel d'automatisation. La prochaine phase, WP4 - Perspectives d'avenir, explorera les technologies émergentes telles que l'IA, les jumeaux numériques et l'automatisation avancée afin d'améliorer encore la gestion et l'efficacité des tunnels.

WP4 : Perspectives d'avenir et avancées technologiques

Le quatrième work package (WP4) explore les futurs développements et innovations technologiques susceptibles d'améliorer encore la gestion, la sécurité et l'efficacité des tunnels. Cette phase se concentre sur l'intégration de l'intelligence artificielle (IA), de la technologie Digital Twin, des progrès de l'IoT et des réseaux de communication de nouvelle génération pour établir une infrastructure de tunnel plus autonome et plus intelligente.

La technologie Digital Twin pour la gestion des tunnels
Un **jumeau numérique** est un modèle virtuel d'un tunnel physique qui reflète les conditions en temps réel et peut être utilisé pour la simulation, la surveillance et la maintenance prédictive. Ses principaux avantages sont les suivants :

- **Synchronisation des données en temps réel** : Garantit que les représentations numériques restent à jour avec les données des capteurs.
- **Maintenance prédictive** : Utilise des analyses pilotées par l'IA pour anticiper les pannes d'équipement avant qu'elles ne se produisent.
- **Simulation et planification de scénarios** : Permet de tester les plans d'intervention d'urgence dans un environnement virtuel.

Intégration de l'intelligence artificielle (IA)
Les outils pilotés par l'IA peuvent révolutionner l'exploitation des tunnels en automatisant les tâches clés et en améliorant les processus de prise de décision. Les applications de l'IA dans la gestion des tunnels comprennent

- **Détection des anomalies** : Les algorithmes alimentés par l'IA peuvent identifier les écarts dans les conditions du tunnel, tels que les monts de température ou les schémas d’action inhabituels.
- **Contrôle automatisé du trafic et de la ventilation** : L'IA peut ajuster dynamiquement la ventilation du tunnel et l'utilisation des voies en fonction de la densité du trafic et de la qualité de l'air.
- **Programmation intelligente de la maintenance** : L'IA optimise les délais de maintenance en analysant les données historiques et les performances actuelles de l'équipement.

IoT et infrastructure intelligente
L'internet des objets (IoT) continue de jouer un rôle essentiel dans la gestion des tunnels numériques. Le WP4 explore :

- **Réseaux de capteurs** : Intégration de capteurs intelligents pour surveiller la qualité de l'air, l'éclairage, l'intégrité structurelle et les conditions de circulation en temps réel.
- **Informatique de périphérie** : Traitement local des données dans les systèmes de contrôle des tunnels afin de réduire la latence et d'améliorer les temps de réponse.
- **Amélioration de la cybersécurité** : Protéger les systèmes de communication des tunnels contre les cybermenaces grâce à la transmission cryptée des données et à des mesures de sécurité basées sur l'IA.

Réseaux de communication de nouvelle génération

L'émergence de la **6G et de l'informatique quantique** offre de nouvelles opportunités pour la digitalisation des tunnels. Les principaux avantages sont les suivants :

- **Latence ultra-faible** : Communication plus rapide entre les appareils IoT et les systèmes de contrôle.
- **Traitement quantique des données** : Un calcul plus efficace pour l'analyse et la prise de décision en temps réel.
- **Amélioration de la connectivité** : Des réseaux fiables à haut débit pour une meilleure intégration de l'infrastructure des tunnels avec les systèmes de transport intelligents à l'échelle de la ville.

Défis et considérations

Malgré les avantages des technologies émergentes, plusieurs défis doivent être relevés:

- **Normalisation des données** : Assurer la compatibilité entre les différents modèles numériques et l'infrastructure existante.
- **Coût et évolutivité** : La mise en œuvre de l'IA, de l'IoT et des jumeaux numériques nécessite des investissements substantiels et une planification minutieuse de l'évolutivité.
- **Conformité réglementaire** : S'adapter à l'évolution des réglementations en matière de sécurité et de cybersécurité pour maintenir l'intégrité de l'exploitation des tunnels.

Conclusion du WP4 et étapes futures

Le WP4 met en évidence le potentiel de l'**IA, des jumeaux numériques, de l'IoT et des réseaux de communication de nouvelle génération** pour révolutionner la gestion des tunnels. Les étapes futures comprennent :

- **Mise en œuvre pilote de jumeaux numériques** pour certains tunnels afin d'évaluer les avantages réels.
- **Poursuite du développement d'outils d'automatisation pilotés par l'IA** pour la maintenance prédictive et l'optimisation du trafic.
- **Collaboration avec les décideurs politiques et les chefs d'entreprise** afin d'établir des cadres normalisés pour l'infrastructure des tunnels numériques.

En adoptant ces avancées technologiques, les tunnels suisses peuvent se transformer en un **réseau de transport intelligent** et entièrement connecté, garantissant une sécurité, une efficacité opérationnelle et une durabilité accrue pour l'avenir.

1 Initial Situation and Principles

1.1 Introduction

Managing tunnels in the recent years has become more demanding and challenging mainly due to the increase of traffic volume as well as the aging of the existing structures that must comply to more stringent standards than those applicable at the time when they were built. Tunnel managers must deal with systems reaching their life span, plan the refurbishment of a part of the equipment and eventually totally replace it. The tunnel equipment is becoming more complex and interconnected, the information technology domain and data communication is getting more and more relevant, the minimal operation conditions are more intricated and the interfaces between systems more complex. All activities must be carried out without causing any interruption of the communication channels, preventing traffic disruptions and the formation of intolerable queues.

Tunnels in Switzerland are owned by the Confederation and their operation is delegated to the Territorial units. This situation introduces an important interface between the two entities that is also becoming more complex; there are many different actors involved in the tunnel operation, starting from the administrative owner, the tunnel operating centre (Police), the tunnel maintenance division, the Safety Officer (SiBe) and the intervention teams. Another aspect is determined by the fluctuation of staff in the different linguistic areas and the respective need for a transfer of skills and knowledge.

Tunnel documentation is available in different formats and languages, making the exchange of information between the actors involved in the increasingly complex management of tunnels difficult.

This research stems from the need to find a standardized and unambiguous exchange data format, independent of linguist constraints and able to simplify and optimize management processes. Unified Modelling Language (UML) and Business Process Model and Notation (BPMN) have been identified as potentially suitable modelling standards and are therefore the focus of this research.

UML is a standardized language generally used in informatic project development to model software prior to its implementation. The modelling phase is not necessarily mandatory for the implementation of a given computer program; programmers often build software directly by programming it in the language they prefer.

UML provides developers with a language that, through a semi-graphical and semi-formal notation, allows system architects, without necessarily having a specific basis in computer design, to model a system in its various aspects and thus provide a complete model that can be used for the actual programming.

UML also allows to define models (patterns) that can be reused for different but still similar basic problems; these patterns can be optimized and adapted as needed, giving the designer a solid model to apply in his own project. Similar to what a programmer can do throwing himself without modelling in the implementation of his own software, a service engineer/technician can throw himself into the execution of his own activities basing himself on his own experience and abilities.

BPMN is a graphical modelling standard designed to specifically represent complex business processes and whose structure is like UML activity diagrams. BPMN can provide easily understandable graphical notation for a wide range of processes, starting from design processes to implementation processes, representing therefore a powerful tool for many stakeholders.

As with software projects, UML and BPMN can be used to model the operational processes of a tunnel with the following objectives:

- Reducing the complexity of a process by means of optimization
- Automate and optimize the execution of processes, particularly those related to the maintenance and monitoring of objects
- Increase the quality of work while reducing costs and time for processes

UML and BPMN are therefore able to create a common language for everyone, facilitating the passage and understanding of information, making the implementation of measures uniform and unambiguous, and allowing the use of models that can be easily adopted in different geographical and linguistic regions.

1.2 Project goals

The goal of this research is to evaluate how UML and BPMN could be used to model the different processes necessary for the operational and safety management of a tunnel, starting from the management of static data, the management of minimum operating conditions, maintenance and inspection activities.

This research shall be intended as feasibility study in which the following aspects are investigated:

- Static representation of a tunnel (geometry and system architectural modelling)
- Dynamic representation of a tunnel (modelling of changing values, modifications of the static representation of the tunnel)
- Feasibility study for modelling of interactions and interfaces between systems in tunnel
- Possible automation process for of UML diagram creation
- Representation of workflows, especially regarding maintenance activities and procedures to be undertaken to guarantee minimum operational conditions

Once the feasibility is verified, it will be necessary to detail how the modelling might be applied to the management, operation and maintenance of the FEDRO highway tunnels.
The goal is to obtain implementable models for a realistic complete and comprehensive digitization of a tunnel, providing in this way a fact-based tool with a general applicability to operation safety and maintenance management of tunnels. It is noted that software application development is not in the scope of this research. The aim is to study the feasibility of applying UML models for tunnel safety and management, to give concrete examples of the application of the models and to formulate conclusions on possible applications of the models and their processes. Similarly, to what happens in the IT world, UML models form the basis for possible future implementations.

1.3 Structure and procedure of the initial project

1.3.1 Description of work packages

The project was structured into 4 distinct phases, which correspond to the work packages WP1-WP4. Each WP was consisted in the activities described as follows:

- WP1: Project Setup and Target Definition
- WP2: Static, UML Usage for Structural Tunnel Modeling
- WP3: Dynamic, BPMN for Maintenance and Operation Tunnel Modelling
- WP4: Project Outlook

The WP1 consisted mostly of organizational activities aimed to bridge the communication gap induced by the very different backgrounds of the team members and to develop an efficient working strategy. The goal was to dry-run the complete pipeline from informal domain knowledge to the creation of a valid set of UML diagrams encoding that knowledge, as well as ensuring that the thus created models can be persisted in a knowledge base that can be accessed by all team members. Moreover, in WP1 the selection criteria for the choice of a real-practice tunnel example were analysed.

The focus of WP2 is the static modelling of the tunnel, consisting in the creation of a wide set of UML diagrams that model the complete structure, in all its constituent parts, of a real-practice tunnel example. The goal was to verify whether the tunnel complexity could be reproduced with a sufficient degree of detail and to evaluate the usability of the chosen modelling tool in terms of its capability to handle the needed amount of data and diagrams. Due to the complex structure of a tunnel and consequent high modelling effort, the investigation of modelling automation techniques was set as well as goal of this work package. A relevant part of WP2 consisted moreover in the analysis of existing modelling formats (as for example IFC, OPC-UA data modelling, etc.) with the aim of finding a common structure and to find therefore a way to centralize and eventually distribute tunnel-related information. For this reason, the research on middleware was as well considered as part of this package.

WP3 focused on the modelling of the operational and maintenance parts of the tunnel. This work package aimed to evaluate the ability of BPMN in modelling tunnel management activities and to compare it to UML. For this purpose, the goal was to produce typical maintenance and process diagrams using both modelling techniques.

The content of WP4 was mainly constituted by the consolidation of the analysis carried out in the previous work packages and by the evaluation of possible future applications of the acquired knowledge. In-depth discussion and conclusions about pros, cons and effectiveness of using UML and BPMN diagrams for tunnel modelling represented a consistent part of this work package.

1.3.2 Procedure, methodology and approach

The project comprises a multidisciplinary team that brings in-depth knowledge of both the IT-world and tunnels applications. The team was formed of the following three competence domains:

- Tunnel competence, including civil and electromechanical engineering, tunnel management and maintenance, safety procedures, financial management and BIM.
- Information technology knowledge concerning modelling and programming.
- Operative in-depth knowledge for tunnels.

The iterations within the teams consisted of initial exchange meetings, where the knowledge of tunnel competence was transferred to the software engineers. After the modelling phase carried out by the software engineers, during which the tunnel was modelled and implemented through UML diagrams, the results were jointly discussed and analysed within the team.

The results of each work package were discussed with an monitoring committee, responsible for overseeing the progress of the project. The content elaborated within a work package were presented to the monitoring committee during periodic meetings.

1.4 Choice of UML Modelling Tool

One question that needed upfront clarification was the choice of the UML tool that was going to be used throughout the project.

A comparative evaluation of several tools was carried out, as shown in the table below, with the tool called "Astah UML" finally being chosen for reasons that can be deduced from Table 1.

For BPMN modelling, an online tool has been identified that allows a standard modelling level to support this project (https://bpmn.io/).

UML tool comparison

	Astah UML	Star UML	RAD Studio	CollabUML	PlantUML	JetUML
General information						
Short description	Lightweight UML tool with plugin customization.	Text	Delphi and C++ IDE with virtual visual design tools including UML.	Collaborative PlantUML editor using EtherPad.	Text	Lightweight desktop app for creating UML diagrams.
Price	53 EUR/year Single license	99 USD	3'330 EUR/year	Free	Free	Free
Extensible [1]	YES	YES	NO	YES	YES	YES
Feature requirements						
Positioning[2]	YES	YES	YES	NO	NO	YES
Interactive[3]	YES	YES	YES	NO	NO	NO
Hyperlinking[4]	YES	YES	YES	NO	YES	NO
Diagram requirements						
Class	YES	YES	YES	YES	YES	YES
Object	YES	YES	NO	YES	YES	YES
Sequence	YES	YES	NO	YES	YES	YES
Activity	YES	YES	YES	YES	YES	NO
Use Case	YES	YES	YES	YES	YES	YES
State Machine	YES	YES	YES	YES	YES	YES
Additional preferred features						
Collaborative[5]	YES, in PRO version only (119 USD per year)	NO	NO	YES, through EtherPad	NO	NO, but JSON can be exported
Scriptable[6]	YES, from source code or via custom plugins	YES	From source code	YES, with PlantUML	YES	YES, JSON or XML can be created
Useful extras	API, generate code from a given diagram.	Possibility to code your own	Synchronisation with source code, patterns	-	-	-

[1] New features can be added (open source).
[2] Position of elements can be set.
[3] Tooltips, inspection of elements etc.
[4] Links to other diagrams.
[5] Online, live, shared.
[6] Create diagrams from script descriptions.

	Astah UML	Star UML	RAD Studio	CollabUML	PlantUML	JetUML
		extensions and commands.	registry, layouting.			
Usability	Possibility to use entities from one diagram in the other one with automatic synchronisation. Creation of custom hyperlinks from one diagram to another.	Difficult navigation. Hyperlinking and interactiveness available in HTML exports. Limited customization.	UML is just one feature of a more comprehensive program. Models are related to code.	Essentially equal to PlantUML, with the difference that the script is online. Only renders as image and does not support hyperlinks.	Can render to svg, which supports hyperlinks.	Actively developed, open source. At current stage, usability on macOS difficult.

Table 1: UML tool comparison.

1.5 Graphical representation of diagrams in the present report

The graphical presentation of diagrams in this report is often a challenge due to the size and amount of content. Diagrams are often very large and the relationships between the various objects do not allow them to be broken down in a comprehensible and efficient manner. Diagrams in this report have been adapted to allow correct reading and condensed to contain the most important information.

2 Literature and Technical Documentation

2.1 Literature review: state of research

"Tunnel Management" and "Unified Modeling Language", UML are two terms from different fields. The first one falls in the area of interest and framework of the administrative authorities of tunnels (e.g., FEDRO) and engineers, the second one is mainly used by specific information technology workgroups of highly skilled software engineers.

Over the last years, the *tunnel management* has been regulated by FEDRO by a series of directives, manuals, and documentations in a dedicated "management" domain. The digitalization of the different structures (including tunnels) has been occasionally implemented on a static level for archiving and data collection databases (e.g., KUBA database, TRA database and BSAS database). In Austria, a new project called "Digitalisierungsoffensive der ASFINAG" was recently launched, requesting ideas, proposals and suggestions for a digitalization standardization of different domains, including the construction process, project variation, costing and billing, etc. This project was launched in 2020, and results are expected within the next years.

UML, on the other hand, was conceived in the field of software engineering to counter the many notational systems in use to describe the design and architecture of software systems [UML] and has become a standard tool in software engineering projects [9]. The Unified Modeling Language (UML) is a versatile and widely adopted modeling language that originates from the field of software engineering. It was developed with the goal of providing a standardized way to represent and collaborate on the design and architecture of systems. By offering a unified framework, UML allows designers, developers, and stakeholders to communicate complex ideas more clearly and effectively. Prior to UML's introduction, the software industry lacked a cohesive modeling standard, relying instead on a variety of disparate notational systems and methodologies. UML brought these various approaches together, and over time, it has become the de facto industry standard.

UML is currently overseen by the Object Management Group (OMG), an international technology standards consortium that continues to manage and evolve the language. Since 2005, UML has also held the status of an ISO-approved standard, further solidifying its role in system modeling. The UML specification is composed of several key elements, including:

- a superstructure, which defines the notations and semantics for its diagrams and model elements,
- an infrastructure that establishes the core metamodel,
- the Object Constraint Language (OCL) for setting rules, and

- an interchange format that facilitates the exchange of models between different tools and platforms.

One of the key strengths of UML is its ability to represent the "architectural blueprints" of a system. This system could be a piece of software or any general system, as UML is flexible enough to model both. The language encompasses a wide range of elements, such as components, activities, and user interfaces, to capture different aspects of system design. UML diagrams are typically categorized into two main types: structural diagrams and behavioral diagrams. Structural diagrams, such as Class diagrams, Component diagrams, and Package diagrams, focus on depicting the static aspects of a system—essentially, how the system is composed and how its various elements relate to one another. On the other hand, behavioral diagrams, such as Activity diagrams, Sequence diagrams, and State diagrams, illustrate the dynamic aspects of a system, focusing on how it operates and reacts to different inputs or changes over time.

In this way, UML serves as a powerful tool that bridges the gap between abstract system design and practical implementation, providing a visual and standardized means of understanding both the structure and behavior of systems.

Another key benefit of UML is the vast array of tools that support it, as well as the opportunities it creates for automating various aspects of software development. For instance, once a comprehensive UML model has been created, these tools can facilitate the automatic generation of source code or even the structural setup of databases. This capability can significantly accelerate development timelines while improving consistency and reducing the chance of human error during the coding or design phases.

A recent systematic literature review by Koc et al. further reinforces UML's value, showing that it continues to play an essential role in software engineering research. The review found that UML is still extensively used, especially for the purposes of software design and modeling, with class diagrams being the most frequently employed type. These diagrams provide a clear and effective way to map out the structure and relationships within a system, making them a valuable resource for developers and engineers.

Another literature review by da Silva et al. focused on domain-specific languages for self-adaptive systems and highlighted another strength of UML—its flexibility and extensibility. The review noted that UML's foundational framework has led to the development of numerous domain-specific variants, where specialized versions of UML diagrams have been crafted to meet the needs of different fields. This adaptability underscores UML's relevance across a wide range of industries, beyond traditional software engineering, allowing it to remain a crucial tool for modeling increasingly complex and dynamic systems.

2.1.1 Conclusion

In conclusion, UML's comprehensive tooling support, coupled with its adaptability to domain-specific needs, ensures its continued relevance in both general software development and specialized fields. Its ability to automate essential processes and accommodate tailored modifications makes it a versatile and powerful asset for modern system modeling.

While UML is the clear choice for modeling the structural aspects of systems—whether it be a software system or even something more physical, like a tunnel—it offers only limited support when it comes to modeling processes. Its behavioral diagrams, though useful, don't provide the depth required for complex process modeling. For this, a more suitable alternative is BPMN (Business Process Model and Notation), a graphical language specifically designed for representing business processes within a business process model. Like UML, BPMN is maintained by the Object Management Group (OMG), and it has also been ratified as an ISO standard.

BPMN's graphical notation, which results in the creation of Business Process Diagrams (BPD), is rooted in a flowcharting technique that bears similarities to UML's activity diagrams. However, BPMN goes a step further by providing a more process-centric approach, particularly with its ability to map the graphical elements directly to execution languages such as BPEL (Business Process Execution Language). This mapping allows for smoother transitions from the design of a process model to its automated execution. One of BPMN's standout features is its accessibility. It was explicitly designed with non-technical stakeholders in mind, such as business analysts and managers. The notation is intuitive and easy to understand, allowing these professionals to participate directly in process modeling and analysis without needing a deep technical background. This focus on simplicity and clarity has been one of the driving factors behind BPMN's growing popularity in recent years. BPMN's flexibility is further demonstrated by its use in a wide range of contexts, including the automated generation of web services based on BPMN specifications. This capability highlights how adaptable BPMN can be, offering solutions not only for modeling but also for automating business processes. As it continues to gain traction, BPMN is proving to be an indispensable tool for organizations looking to bridge the gap between business process design and execution.

Appling UML modelling to maintenance is not a new concept as e.g. described in the reference document [13] listed in the bibliography section, where the Reliability Centred Maintenance (RCM) and the Condition Based Maintenance (CBM) systems integration is modelled through UML. In this interesting case, the models were build on a data structure based on the MIMOSA standard that today enable Digital Twins to be defined and maintained on a supplier-neutral basis.

The subject of maintenance is very present in literature, in particular with regard to all systems directly linked to safety. As far as tunnels are concerned, it is important that maintenance can guarantee that all minimum operating conditions are always respected, in order to guarantee the designed level of safety and avoid closures to traffic that cause significant inconvenience to users.

2.2 Tunnel documentation

Automation of UML diagrams creation based on existing documentation has been part of this research and analysis of the final documentation has been carried out with the aim of finding a standardized file to use as input of the UML diagrams automation process.

In the following sections, a description of the available tunnel documentation and its formats is therefore provided.

2.2.1 Final documentation of the system for electromechanical equipment

For archival reasons, each electromechanical system in a tunnel shall have a so-called "final documentation", the purpose of which is a detailed description of the functionalities and of the architecture of the systems. The aim of the final documentation is to provide a sufficiently detailed description of the system, in order for the operational staff (Territorial Unit) to be able to correctly operate the tunnel and to carry out maintenance activities.

Final documentation is typically composed of narrative descriptions, diagrams and lists. The content and the structure of folders in the final documentation is regulated by the FEDRO's technical manual 20001-50002, which states the required submission format of specific documents (PDF only or PDF and source file). For electromechanical equipment, final documentation only sporadically comprises native files, with PDF being therefore the most used submission format.
It is noted that technical manual 20001-50002 was subject to changes over the years, resulting in electromechanical systems to show different final documentation structures. While the type of documents to be contained in the final documentation are determined by the technical manuals, there are no explicit guidelines regulating the exact content and style of each document. The documentation is prepared by the system construction company and is verified by the project engineer. The degree of detail and the exact information contained in a specific document is therefore at the construction company's and project engineer's discretion. Although the information contained in the final documentation of different systems is similar, it is rarely identical.

Moreover, final documentation language is determined by the region where the construction takes place, introducing another substantial difference between final documentations.

The main information and format files that can be expected in an electromechanical final documentation are shortly summarized and described as follows.

- Principle diagrams and construction drawings: principle diagrams typically consist of conceptual hardware and software architectural diagrams. Construction drawings show detailed dimensions and installation concept of the system. These plans are required to be submitted both as PDF and as source files (dwg, vsd/vsdx, etc.). No constraint on the program used to elaborate the diagram is given, therefore source file format is not unique.

- Material lists, electrical protection lists, wiring lists: detailed lists of the electrical connections and of the used products. These documents are submitted as PDF and typically as well as source file (xls, xlsx, doc, docx).
- Software datapoints: datapoints of information exchanged between controllers of different hierarchical levels. These lists are usually submitted as PDFs and occasionally as xls/xlsx files. Historically datapoints were coded according to the Territorial Unit's coding system. Nowadays a national coding system has been developed and is regulated by FEDRO's guideline 13013 (AKS-CH coding system), however old systems still rely on Territorial Units' coding systems.
- Products datasheets: submitted as PDF files only, content highly dependent on supply company.
- Narrative descriptions: narrative descriptions usually concern the hardware and software structure of the system, the interfaces with other systems, user manuals and maintenance manuals. These documents are submitted as PDF only.
- Flowcharts: flowcharts are used by the designer to specify how critical routines must be executed, normally limited to the ventilation control system. These flowcharts are part of the final documentation and normally embedded in text documents (doc, docx).
- Certifications and protocols: these documents comprise material/product certifications as well as test protocols of the entire system. These are usually signed documents and hence available as PDF files only.

Tunnel final documentation for electromechanical equipment is composed of various information presented in various formats and is only partially standardized. Although in the last years there have been several adjustments in FEDRO's guidelines aimed to standardize systems structure and documentation, the influence of historical and region related factors is still visible.

2.2.2 KUBA database

KUBA is an application, divided into modules, which collects the needed data for the management of national roads constructions. The goal of KUBA is to provide organizational support for planning and executing maintenance operations. One important module of the KUBA application is the KUBA database (KUBA-DB), which aims to standardize data collection of national road constructions.

The recording of road structures in the KUBA database is divided into three categories:

- Object structure: characteristics, localization and documentation
- Inspection: inspections, results, documentation, segmentation
- Maintenance: characteristics, documentation

In general, KUBA categories comprises predefined fields that shall be completed to describe the national road constructions. Whereas some fields are mandatory, other are optional. In order to ensure a standardization, the value of most fields shall be chosen from predefined options. Free fields are kept to minimum. Additional documentation as for example pictures, user manuals etc. can be loaded into the database if considered relevant.

The object structure category comprises characteristics of the construction, from the geometry of the construction itself to the subsystems that characterize the construction. Localization of the construction through the standardized FEDRO's spatial reference system (RBBS) is as well an important part of this category.

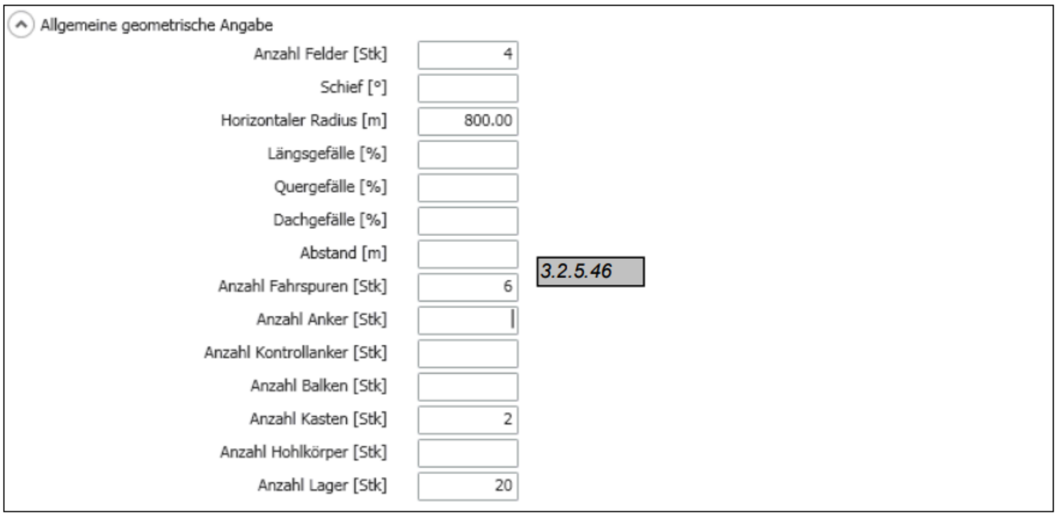

Figure 1: KUBA database extract for object structure category

The inspection category focuses on the collection of inspection dates, type of inspections and condition assessment of the construction and of its subsystems. The graphical interface of the inspection category comprises 4 tabs (inspection, documentation, results and segmentation). The inspection tab comprises general information about last inspections and recommended operations, the results tab comprises a description of inspection results, whereas the segmentation is used to further subdivide the smallest units of the building structure according to their condition development. Documentation tab is used for addition of further documentation, if considered useful.

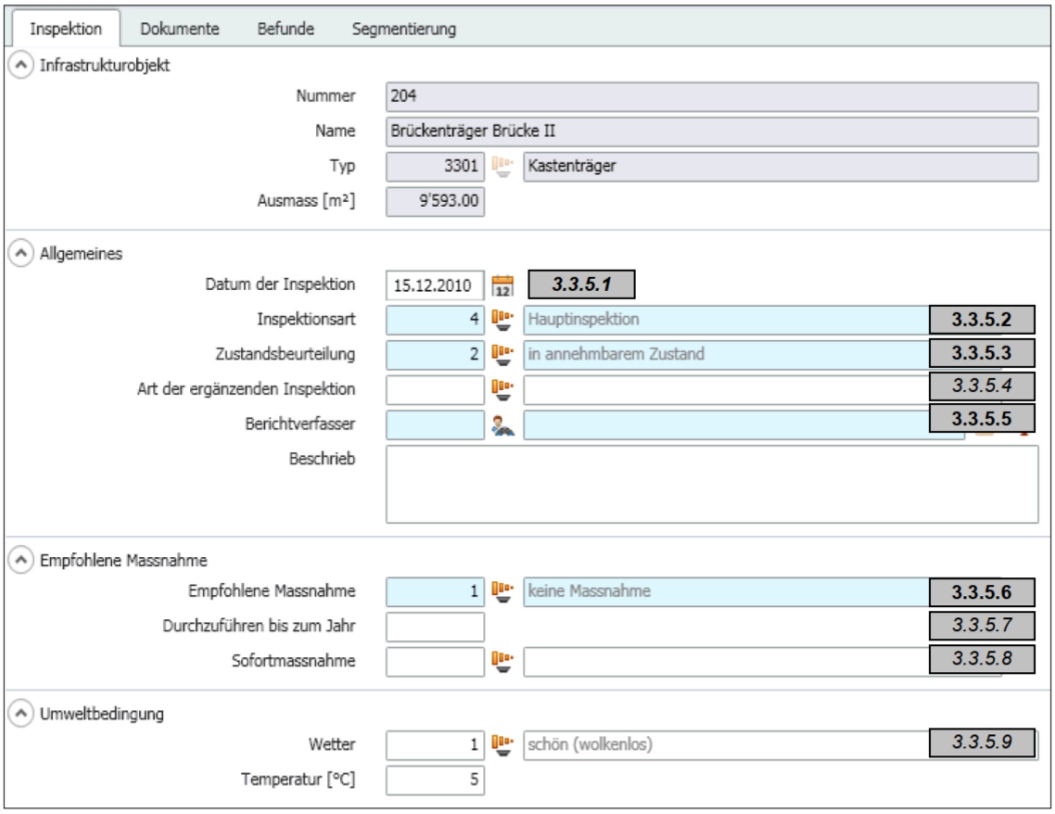

Figure 2: KUBA database extract for inspection category

The maintenance tab focuses on tracking the status and costs of maintenance activities. The documentation tab is used for addition of further documentation, if considered useful.

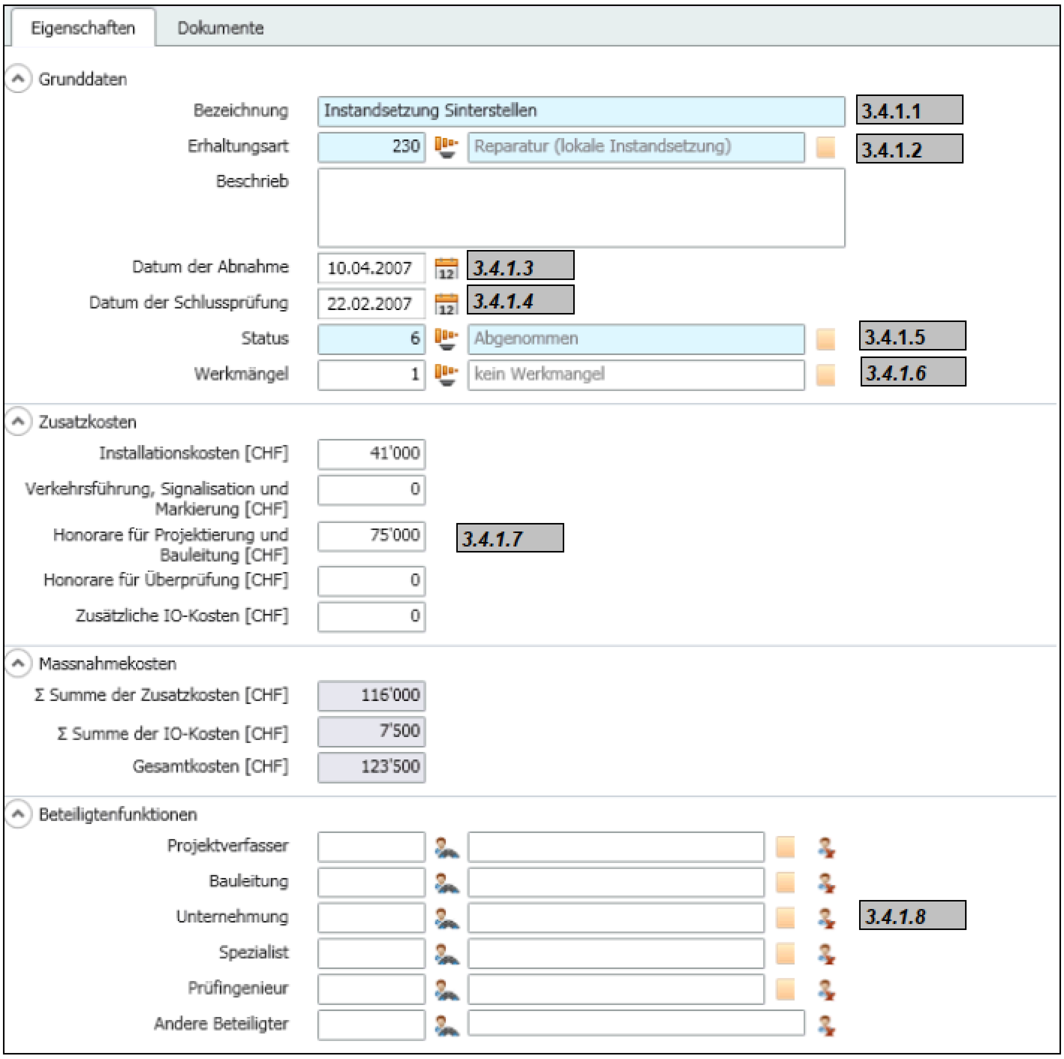

Figure 3: KUBA database extract for maintenance category

2.2.3 FA BSA database

The FA BSA database is an inventory for operational and safety equipment of national roads. The database records products installed and their related relevant information, which is used to assess the condition of equipment.
For each item the database contains following fields:

- Localization fields (building, room, installation type etc.)
- Equipment classification (system and subsystem to which the item belongs according to AKS-CH coding system)
- Additional information (installation year, guarantee period, brand, supplier etc.)

The data can be modified through a dedicated interface and exported as xls/xlsx file.
An extract of the FA BSA database is represented in the following pictures.

Tipo		Opera	Abbr.		Locale, nicchia ecc.	Abbr.		No.	Coollocazione	Abbr.		No.
Prodotto	Piumogna	centrale	Z	1/4 N-S	locale	R	L06 (Lavabo)	0006	luogo di montaggio	MO	Soffitto	02
Prodotto	Piumogna	centrale	Z	1/4 N-S	locale	R	L06 (Lavabo)	0006	luogo di montaggio	MO	Parete	01
Prodotto	Piumogna	centrale	Z	1/4 N-S	locale	R	L03	0003	luogo di montaggio	MO	Soffitto	02
Prodotto	Piumogna	centrale	Z	1/4 N-S	locale	R	L03	0003	posto armadio	SP	02 USV	02

Figure 4: FA BSA database extract, localization fields.

Impianto	Abbr.			Sottoimpianto	Abbr.			Aggregato	Abbr.		No.
impianti annessi	N	Piumogna	PIU	impianto elettrico domestico	HI	1/4 N-S	1/4 N_S	illuminazione locali	RB	L06	0006
impianti annessi	N	Piumogna	PIU	impianto elettrico domestico	HI	1/4 N-S	1/4 N_S	presa elettrica multipla	STV	L06	0006
impianti annessi	N	Piumogna	PIU	impianto elettrico domestico	HI	1/4 N-S	1/4 N_S	illuminazione locali	RB	L03	0003
energia	E	Piumogna	PIU	rete di emergenza	NST	1/4 N-S	1/4 N_S	raddrizzatore / convertitore	GWR	Impianto UPS	01

Figure 5: FA BSA database extract, equipment classification fields.

	Fabbricante dei componenti	Fornitore	Tipo di prodotti	Prezzo d'acquisto (CHF/pz.)	Messa in servizio	Inizio della garanzia	Scadenza garanzia
5	Disano	-	963 Hydro LED	110.00	2018	10.04.2018	09.04.2021
2	Feller		Presa T13	10.00	2018	10.04.2018	09.04.2021
5	Disano	-	963 Hydro LED	110.00	2018	10.04.2018	09.04.2021
1	GE Consumer & Industr.SA	GE Consumer & Industr.SA	UPS S.PRO 40kVA_S.8_ECN1304	9909.60	2013	10.07.2013	09.07.2016

Figure 6: FA BSA database extract, additional information fields.

2.2.4 Operational safety documentation

Safety documentation is prepared for each individual section or artifact. It is structured into 5 separate parts:

- Part 1: structures and objects
- Part 2: operations and safety plan
- Part 3: emergency management
- Part 4: reports and analysis
- Part 5: management

Part 1 summarizes in the different degrees of detail all the technical information that may be useful for the analysis of safety aspects. The goal is to achieve a comprehensive collection of all useful information in a unique document, in order to avoid having to analyse multiple documents in the archives. Part 1 is usually composed of a narrative text with annexes and is typically submitted in PDF format.

Part 2 of the safety documentation contains all information regarding the organization and the resources needed for the management of FEDRO's structures on the territory under the responsibility of Territorial Unit. In particular, the document aims to identifying the relevant figures involved in the supervision and management of the considered area, in relation to different operational and maintenance activities. Moreover, basic training for emergency operations is contained (including Minimal Operation of Conditions).

Part 3 is divided into modules and comprises plans and flowcharts that establish the procedures to undertake in case of emergency and all the related relevant information (intervention plans, emergency contact, emergency accesses to the site, nearest hospital etc.).

Part 4 of the safety documentation describes procedures for recording and analysing significant accidents (fires, dangerous goods transport accidents, equipment defects, etc.) and concepts suitable for planning and organizing trainings for first responders.

Part 5 covers the administrative part and defines the roles and responsibilities for tunnel maintenance, as well as the procedures for the commissioning of a tunnel.

The structure of operational safety documentation is regulated by FEDRO's guideline 16050. Although the content of operational safety documentation is standardized, the actual implementation of the documentation may vary.

2.3 UML and BPMN

UML is a standardized, general-purpose modelling language used to visualize, specify, construct, and document the components of complex systems. It is widely used in software engineering, but its versatility also makes it applicable in fields such as business process modelling and systems engineering.
The key features of UML are:

- **Standardized Notation:** UML provides a consistent and standardized way to visualize system designs, ensuring clarity and communication among stakeholders.
- **Versatility and Flexibility:** UML supports various types of systems, from software applications to business processes and hardware systems.
- **Multiple Views:** UML allows for different perspectives of a system, focusing on both static structure and dynamic behavior.
- **Interoperability:** UML can integrate with other modeling languages like BPMN and SysML, enhancing its scope for different applications.
- **Tool Support:** Many tools (such as Visio, Enterprise Architect, and Visual Paradigm) support UML, offering diagram creation, validation, and simulation.

2.3.1 Types of UML Diagrams

UML provides 14 types of diagrams divided into two main categories: structural diagrams (for static aspects) and behavioral diagrams (for dynamic aspects).

2.3.1.1 Structural Diagrams (Static View):

These diagrams describe the static structure of a system, including its components and their relationships:

- **Class Diagram:** Represents classes, their attributes, methods, and relationships (inheritance, association, etc.).
- *Use Case:* Modeling the structure of software systems.
- **Object Diagram:** Depicts instances of classes (objects) and their relationships at a specific point in time.
- *Use Case:* Illustrating examples of object structures.
- **Component Diagram:** Shows the physical components of a system and their interactions.
- *Use Case:* Describing software components and their dependencies.
- **Deployment Diagram:** Models the physical deployment of artifacts on hardware nodes.
- *Use Case:* Visualizing hardware configurations and software distribution.
- **Package Diagram:** Groups related elements into packages and shows their dependencies.
- *Use Case:* Organizing complex systems into manageable modules.

- **Composite Structure Diagram:** Describes the internal structure of a class and the collaboration of its parts.
- *Use Case:* Representing class compositions and collaborations.
- **Profile Diagram:** Extends UML to customize models for specific domains.
- *Use Case: Defining domain-specific stereotypes and constraints.*

2.3.1.2 Behavioral Diagrams (Dynamic View):

These diagrams capture the dynamic behavior of a system, including processes, interactions, and states:

- **Use Case Diagram:** Shows interactions between actors (users or systems) and the system itself.
- *Use Case:* High-level modeling of user interactions with a system.
- **Sequence Diagram:** Represents object interactions in a time sequence.
- *Use Case: Modeling the flow of messages between objects over time.*
- **Activity Diagram:** Depicts workflows, including actions and decision points.
- *Use Case:* Modeling business processes or algorithms.
- **State Machine Diagram:** Shows states and transitions of an object in response to events.
- *Use Case:* Describing the life cycle of an object.
- **Communication Diagram:** Illustrates object interactions based on messages exchanged.
- *Use Case:* Showing collaboration between objects in a system.
- **Interaction Overview Diagram:** Combines elements of activity and sequence diagrams.
- *Use Case:* Modeling complex interactions with a focus on control flow.
- **Timing Diagram:** Focuses on object states and interactions over time.
- *Use Case:* Analyzing time-dependent behaviors in real-time systems.

UML's set of diagrams allows for the modelling of both static structures and dynamic behaviours, making it an essential tool for system architects, developers, and business analysts. Its flexibility and wide range of applications ensure that systems are well-documented, clearly visualized, and effectively communicated across teams.

BPMN is a standardized graphical notation designed to model and visualize business processes. It provides a way for both technical developers and business stakeholders to understand processes clearly and collaborate effectively. Developed by the Object Management Group (OMG), BPMN is widely used in business process management (BPM) initiatives.

The key features of BPMN are:

- **Business-Oriented Notation:** BPMN is designed to bridge the gap between business process design and implementation by providing a notation that is intuitive for business users while also detailed enough for developers.
- **Standardized Notation:** As a globally recognized standard, BPMN ensures consistency in process modeling across organizations and industries.
- **Graphical Flow Representation:** BPMN uses a flowchart-style notation to represent processes, making it easy to follow the sequence of activities and decisions.

- **Executable Models:** BPMN models can be directly used to automate processes in BPM tools, turning high-level designs into actionable workflows.
- **Collaboration Focus:** BPMN supports modeling interactions between different participants, such as departments or organizations, in a process.
- **Detailed Semantics:** It includes rich semantics for defining various process elements, such as tasks, events, and gateways, allowing for complex process modeling.

2.3.1.3 BPMN Elements

BPMN focuses on a single type of diagram called the Business Process Diagram (BPD) but includes various elements to represent different aspects of a process. There are five basic categories of BPMN elements and each of them represent a unique aspect of business process:

- Flow Objects:
  - **Events:** Represent something that happens within the process. Events can be of three types: Start Event: Indicates the beginning of a process. Intermediate Event: Occurs in the middle of a process. End Event: Marks the conclusion of a process.
  - **Activities:** Represent tasks or work to be performed. Task: A single unit of work. Sub-Process: A complex task that can be broken down into smaller tasks. Call Activity: References another process.
  - **Gateways:** Represent decision points in the process, controlling the flow based on conditions. Common types include: Exclusive Gateway: Only one path is taken. Parallel Gateway: All paths are taken simultaneously. Inclusive Gateway: One or more paths can be taken.
- Connecting Objects:
  - **Sequence Flows:** Show the order in which tasks are performed.
  - **Message Flows:** Represent communication between different participants or processes.
  - **Associations:** Link artifacts or information with flow objects.
- Swimlanes: Used to group and organize tasks based on responsibility:
  - **Pools:** Represent major participants in a process (e.g., organizations)
  - **Lanes:** Subdivide pools to show roles or responsibilities within a participant.
- Artifacts: Provide additional information to enhance the model:
  - **Annotations:** Offer explanations or comments.
  - **Groups:** Organize related activities for clarity.
- **Data Objects:** Represent data required or produced by a process.

Types of BPMN Diagrams: While BPMN focuses on Business Process Diagrams (BPD), variations exist based on the modeling focus:

- **Process Diagrams:** Represent internal business processes
- **Collaboration Diagrams:** Show interactions between different entities
- **Choreography Diagrams:** Focus on the flow of messages between participants and their interactions.
- **Conversation Diagrams:** Highlight high-level communication between participants without detailing the process flow.

2.3.1.4 Applications of BPMN

- **Business Process Improvement:** Identifying inefficiencies and optimizing workflows.
- **Process Automation:** Designing executable processes for BPM tools.
- **Inter-Organizational Processes:** Modeling collaborations and interactions between different entities.
- **Training and Documentation:** Providing a clear visual representation of processes for educational purposes.

BPMN is a powerful tool for modeling and communicating business processes. Its intuitive, standardized notation makes it accessible to both business and technical users, fostering collaboration and enabling the transition from process design to automation. With its rich set of symbols and focus on clarity, BPMN is an essential tool for organizations aiming to streamline and optimize their operations.

2.4 IoT and Database

2.4.1 Introduction to IoT

The Internet of Things (IoT) is a growing field in modern technology, influencing industries, businesses, and daily activities. The concept of IoT refers to an expanding network of interconnected devices that collect, exchange, and analyse data over the Internet. These devices include sensors, home appliances, vehicles, industrial machines, wearable technology, and various other smart systems. By incorporating sensors, embedded software, and communication technologies, these objects can connect and interact with other devices, platforms, and cloud-based infrastructures, enabling automation, real-time monitoring, and informed decision-making processes.

The impact of IoT spans multiple sectors, affecting how operations are conducted in industries such as healthcare, manufacturing, transportation, agriculture, and smart cities. By providing real-time insights and predictive analytics, IoT contributes to improved efficiency, optimized resource usage, and increased system reliability. The data generated by connected devices allows businesses and organizations to refine their processes and develop more adaptive strategies.

A key aspect of IoT is its role in bridging the digital and physical worlds, allowing objects to communicate autonomously and reducing the need for human intervention. The connectivity facilitated by IoT supports the integration of artificial intelligence, machine learning, and big data analytics, leading to improvements in automation and operational efficiency. Additionally, IoT adoption can contribute to safety, security, and sustainability by enabling proactive maintenance, risk management, and optimized energy consumption.

As IoT develops further, its applications and potential uses continue to expand, playing a role in digital transformation across different fields. The increasing interconnectivity of devices and the growing volume of data being processed pave the way for more advanced and effective solutions that will continue to shape industries and everyday interactions in the future.

3 WP1: Project Setup & Target Definition

Several real-life examples of tunnels are available in our region. The definition of criteria for the selection of the tunnel was therefore a crucial task for the start of the project. Moreover, the representative tunnel had to be chosen in order to reflect a sufficiently high complexity in order to appropriately verify the effectiveness of UML modelling for the application.

Selection was performed on the following criteria:

- **Tunnel length and layout criteria.** From a structural point of view, long tunnels with a parallel safety tunnel (SISTO) are more complex and are therefore considered for the choice of a target tunnel. The presence of cross passages was also considered interesting from a structural point of view. Tunnels longer than 1.5 km are usually equipped with both longitudinal ventilation and smoke abstraction systems, which are considered indicated for the selection of a target tunnel.
- **Traffic load and flow.** Maintenance activities and interfaces between systems are typically more complex in high traffic load tunnels, therefore a tunnel located on an important communication route was set as a minimum requirement.
- **Tunnel equipment criteria.** The chosen target tunnel was required to be equipped with all the main operational and safety equipment compliant with FEDRO's standards (equipment that is part of FEDRO equipment domains D1 to D8). As already mentioned, the choice of a tunnel with a complex ventilation system (i.e. a ventilation system with both longitudinal ventilation and smoke extraction) was considered a crucial requirement.
- **Availability of base information and documentation.** A tunnel with available safety documentation according to the FEDRO standards and with relevant intervention plans and minimum operational conditions (CME) flowcharts was set as a requirement, since safety documentation is believed to be the most complete documentation for the purpose of this project.
- **Operation and maintenance organization and processes.** Since modelling of operation and maintenance processes represented an important part of this project, a tunnel which requires complex maintenance activities by the Territorial Unit was preferred. It is noted that complex operation and maintenance processes are a direct consequence of choosing a high traffic long tunnel with high availability requirements. The tunnel shall be compliant with the life-cycle procedures and be subjected to period inspection, test and refurbishment cycles so that maintenance modelling activities can be performed in the context of this project.

3.1 Concrete Choice: The Mappo-Morettina Tunnel

Based on the requirements described previously, the Mappo-Morettina tunnel was proposed and accepted by the monitoring committee. A comparison between the requirements and the characteristics of the Mappo-Morettina Tunnel is summarised in the following table.

Comparison of requirements with chosen tunnel

Requirement	Mappo-Morettina characteristics	Fulfilment of requirement
Tunnel length and layout criteria		
Minimum length > 1.5 km.	Approximate length of 5,5 km.	✓
Presence of safety gallery.	Safety tunnel connected to traffic area through cross passages.	✓
Presence of cross passages.	8 cross passages are present.	✓
Traffic load and availability requirements		
High traffic tunnel.	High traffic tunnel with an approximate daily transit of 26'000 vehicles.[7]	✓
High availability required.	Very important connection between Locarno and Bellinzona, which allows to divert commuter traffic from city road. Any maintenance activity is performed at night.	✓
Tunnel equipment criteria		
Presence of operational and safety equipment according to FEDRO's standards.	Presence of main systems for every FEDRO's domain (D1-D8).	✓
Presence of complex ventilation system.	Longitudinal and smoke extraction system are present.	✓
Availability of base information and documentation		
Availability of safety documentation.	Safety documentation, as well as final documentation for every system in the tunnel are present and documented in a comprehensive manner.	✓
Availability of intervention plans and minimum operational condition flowcharts (CME).	Intervention plans and minimum operational conditions available.	✓
Operation and maintenance organization and processes		
Complex operation and maintenance.	Activities which require closing of the tunnel are performed during the night. Every activity shall guarantee a re-opening of the tunnel the next morning. This increases the complexity of operation and maintenance activities and requires organizational effort.	✓
Tunnel compliant with life-cycle procedures.	Periodic inspections, tests and refurbishment cycles are performed by the FEDRO and the Territorial Unit.	✓

[7] Daily transit of vehicles in 2023 according to OASI (Ticino information system for monitoring of environmental evolution)

Table 2: Comparison of requirements with chosen tunnel

Compliance with the requirements is described using the UML 'package' function:

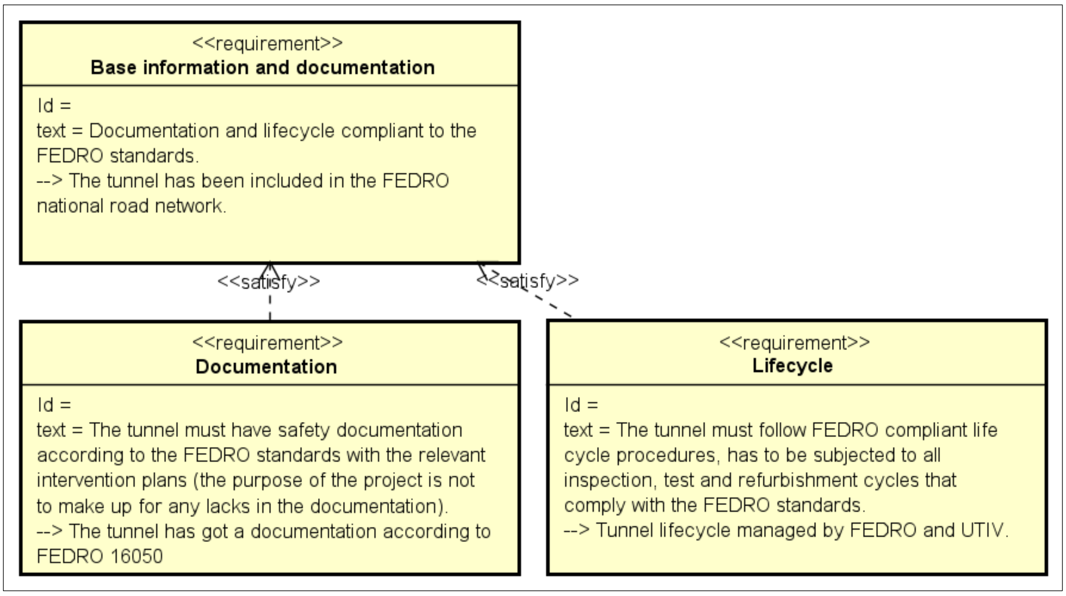

Figure 7: ULM package diagram for conformity requirement of tunnel Mappo-Morettina (documentation)

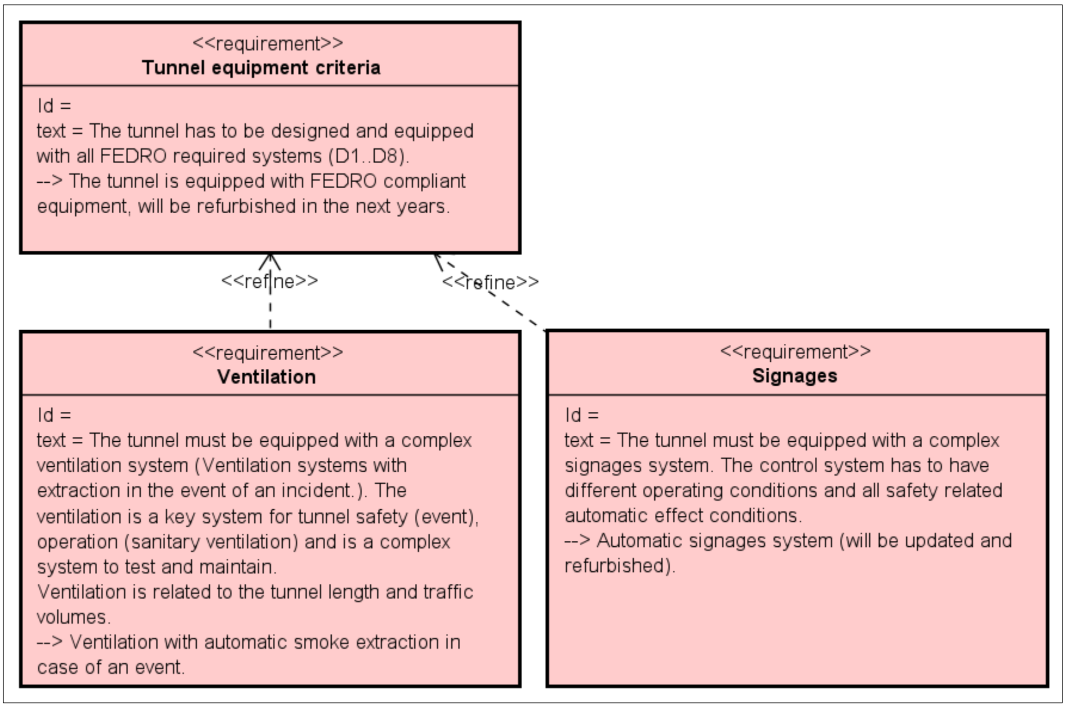

Figure 8: ULM package diagram for conformity requirement of tunnel Mappo-Morettina (equipment)

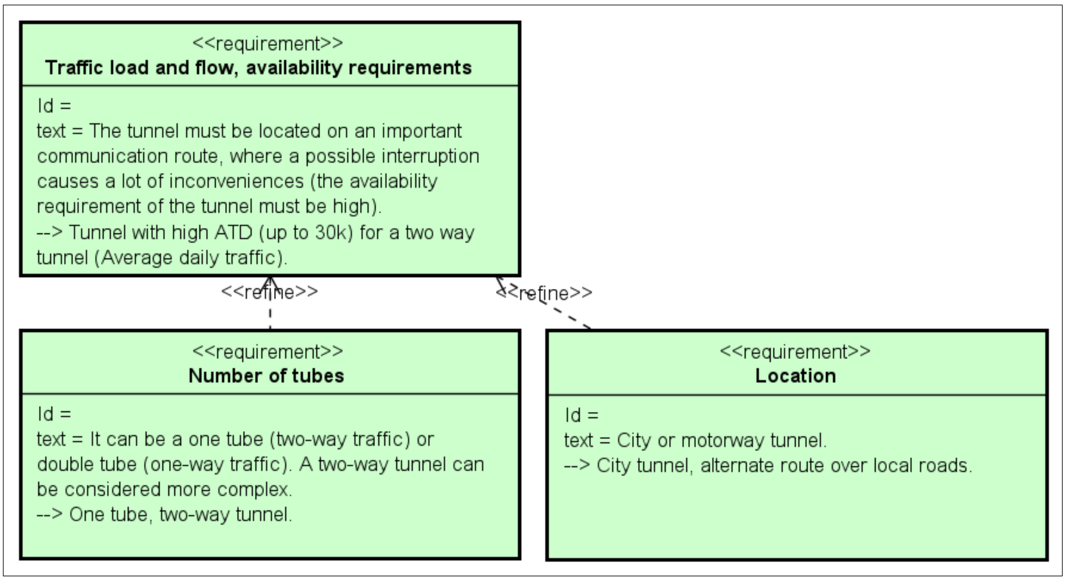

Figure 9: ULM package diagram for conformity requirement of tunnel Mappo-Morettina (traffic load)

3.2 Conclusion of WP1

WP1 included the selection of tools to carry out modelling in accordance with recognised standards (selection described in the introduction) and identified an ideal template for carrying out the next packages. A major challenge now is determined by the collection of the documents necessary to elaborate the modelling templates and to understand how the different systems are interfaced with each other. While processing WP1, it quickly became interesting how very good levels of synthesis and clarity can be achieved by applying UML models, e.g. by defining requirements and the relationships between the various classes and packages. The use of UML stimulates logical reasoning and enables us to synthesise processes we know in other forms in an optimal way, giving us greater confidence in understanding them.

4 WP2: Static: UML Usage for Structural Tunnel Modelling

4.1 Modelling layers theory

A metamodel, in its simplest terms, is a model that describes the structure or framework of another model, essentially serving as a "model of a model." Metamodeling, on the other hand, is the systematic process of creating these metamodels. It involves the in-depth analysis, design, and development of the fundamental structures, rules, constraints, and theoretical frameworks that shape and guide the modeling process for a specific class of problems. The purpose of metamodeling is to provide clarity and consistency in how models are built and understood, ensuring they accurately represent the systems or processes they are intended to simulate or describe.
As the name suggests, metamodeling merges the concepts of "meta" (beyond or above) and "modeling," and is widely used in both software engineering and systems engineering. In these fields, it plays a crucial role by establishing a higher-level perspective, offering tools and methodologies that improve the quality, reusability, and flexibility of models. Through metamodeling, engineers can better organize complex systems, identify patterns, and streamline the modeling process to make it more efficient and adaptable.
Metamodels come in various forms, each tailored to different applications and domains. They can be used to define modeling languages, create new methodologies, or even evaluate existing models for their effectiveness and accuracy. This versatility makes metamodeling an indispensable practice in areas where precision, consistency, and adaptability in modeling are key to solving intricate problems.

A metamodel, also known as a surrogate model, is essentially a simplified version of an actual model. It serves as a model of the model, whether it's applied to circuits, systems, or software-like entities. Metamodels can take the form of mathematical relations or algorithms that capture the relationships between inputs and outputs. While a traditional model is an abstraction of real-world phenomena, a metamodel represents yet another layer of abstraction, focusing on the characteristics and behaviors of the model itself. The relationship between a model and its metamodel is akin to how a computer program conforms to the grammar of the programming language in which it is written. In other words, the model follows the structure defined by its metamodel.
Metamodels come in various types, such as polynomial equations, neural networks, and Kriging models, each designed to capture specific aspects of the model they represent.

The process of "metamodeling" involves constructing a collection of concepts—such as terms, objects, or elements—within a particular domain. Typically, this involves analyzing the input-output relationships of a model and then selecting the appropriate metamodel to accurately reflect those dynamics. By simplifying the complexity of a

high-fidelity model, metamodels provide an efficient way to understand and predict system behaviors without the need for extensive computational resources.
Metamodels have a variety of common applications, including:

- Serving as schemas for semantic data that need to be exchanged or stored efficiently.
- Acting as a language to support specific methods or processes, enabling standardized communication and interpretation.
- Providing a way to express additional semantics for existing information, enriching the depth of understanding in a system.
- Functioning as tools for creating applications that can interact with a wide range of models at runtime.
- Offering schemas for modeling and automatically exploring the structures of sentences within a language, which has applications in areas like automated test generation and synthesis.
- Acting as an approximation of a more complex model to reduce the time, cost, or computational effort required for simulations or analyses.

Because of the "meta" nature of metamodeling, both the practice and theory of metamodels are relevant to broader fields such as metascience, meta-philosophy, metatheories, systemics, and even meta-consciousness. The concept extends its utility beyond theoretical applications and has practical uses in mathematics, computer science, and software engineering. In these fields, metamodels not only serve as efficient surrogates for more complex models but also provide valuable frameworks for understanding the nature of models and systems themselves.

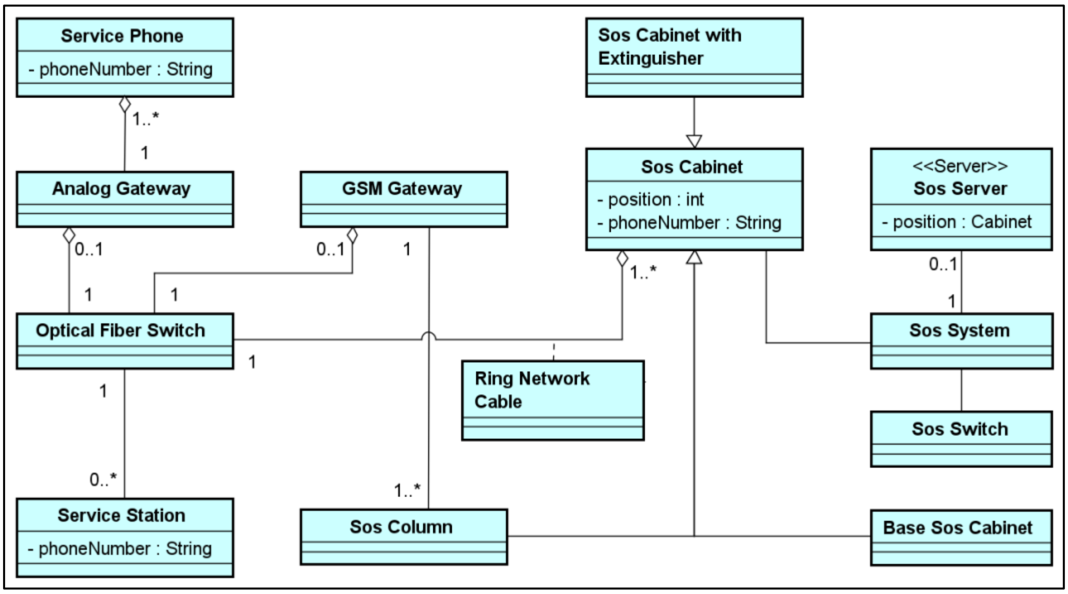

Figure 10: SOS Layer UML Class Model – M1 user model

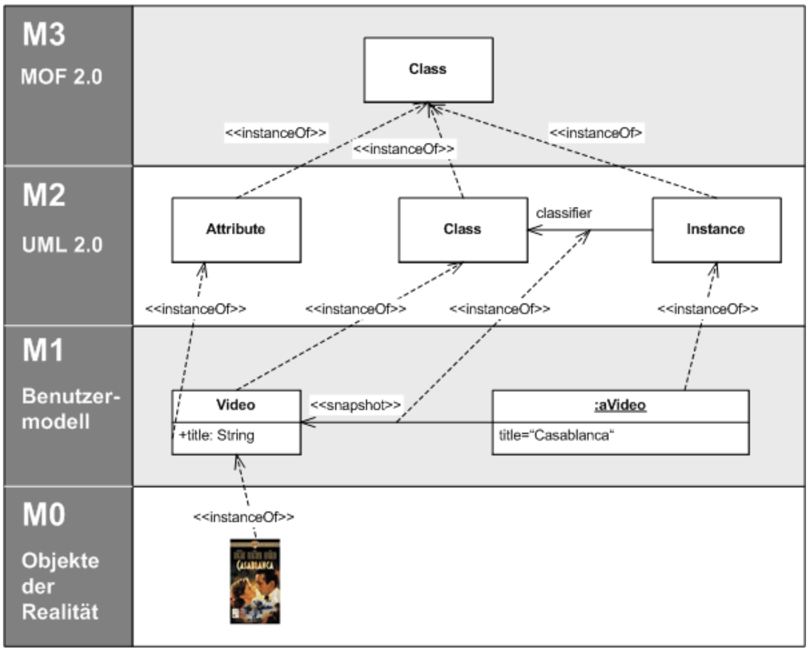

Figure 11: Metamodel Layers

In software engineering, models offer an alternative to traditional code-based development techniques by providing an abstract representation of systems, which can simplify complex processes and improve communication between stakeholders. These models serve as blueprints that describe system functionality, behavior, and structure, helping to bridge the gap between design and implementation. Every model is based on and conforms to a specific metamodel, which defines the rules and structure governing how the model is constructed and interpreted. This ensures consistency and standardization across different modeling efforts within a given domain.

One of the most active and prominent areas of research and application in Model-Driven Engineering (MDE) is the model-driven architecture (MDA) approach, which was introduced by the Object Management Group (OMG). MDA aims to enhance the design and development process by separating the specification of system functionality from the specifics of the underlying platform or technology. This separation allows for greater flexibility, enabling developers to generate code and system designs that can be adapted to different platforms without having to rewrite the entire system from scratch. By focusing on models rather than direct code, MDA improves reusability, scalability, and maintainability of software systems.

The Meta Object Facility (MOF) specification is central to the implementation of MDA. MOF provides a framework for defining, manipulating, and integrating metamodels, making it easier to manage the complexity of large-scale systems. Through the use of MOF, developers can create standardized metamodels that act as a foundation for building models across various domains, such as business processes, software architectures, and data structures. MOF also facilitates interoperability between different tools and platforms by providing a common, consistent way to represent models, thus supporting the broader goals of model-driven engineering.

As the field of software engineering continues to evolve, model-driven architecture and metamodeling are becoming increasingly vital in creating more efficient and flexible systems. These approaches help reduce development time, improve the quality of software, and allow for more straightforward adaptation to new technologies or system requirements.

4.2 UML modelling for static infrastructure

4.2.1 In Theory

In theory, the approach to modeling a tunnel would follow a structured process involving several key steps:

- First, gather all available data about the specific tunnel in question.
- The next step would be to focus on the data that can be processed automatically, ideally stored in a database for easy access and manipulation. However, data stored in formats such as .csv files (e.g., Microsoft Excel) would also be viable for this purpose.
- After gathering the data, the modeling process begins, using UML (Unified Modeling Language) as the primary modeling language. This would result in the creation of a metamodel that represents the tunnel's characteristics and behavior.
- The next phase would involve developing relatively simple tools that, by using the created metamodel and the gathered data, can process the information and produce an object model of the tunnel. This object model would effectively serve as a digital twin of the tunnel, offering a virtual representation of the tunnel's structure and function.
- Once the digital twin is established, additional tools can be built on top of it, enabling further analysis, simulations, or optimizations for various applications such as maintenance planning, safety assessments, or performance monitoring.

However, in practice, the data obtained for the Mappo-Morettina tunnel was far from meeting the ideal criteria of being neatly structured or easily processable due to the historical evolution and heterogeneity of the documentation. The data comprised several gigabytes of both structured and unstructured information, including complex binary files such as PDFs and image files. This unstructured nature of the data made automatic processing significantly more challenging, requiring additional steps to clean, organize, and extract useful information before any meaningful modeling could begin. The heterogeneity of the information and the inevitable differences in the documentation provided by different suppliers lead to a lack of a consistent format for the data and thus to increased complexity, which increased the time and effort needed to move from raw data to a digital twin model of the tunnel.

In such cases, overcoming the challenge of unstructured data might require employing advanced data processing techniques, such as data parsing algorithms, optical character recognition (OCR) for extracting text from PDFs, and image processing for extracting relevant information from pictures. These additional steps would need to be integrated into the workflow before the metamodel could be applied and a digital twin constructed, underscoring the need for adaptable tools capable of handling various data types in real-world scenarios.

4.2.2 In Practice

In practice, documentation consists of over **2 GB** of information spread across approximately **2,600 files.** This vast collection of documents posed significant challenges in terms of organization, analysis, and usability. The following figure presents a **treemap visualization** of this data set, highlighting several important insights and observations about its composition and structure.

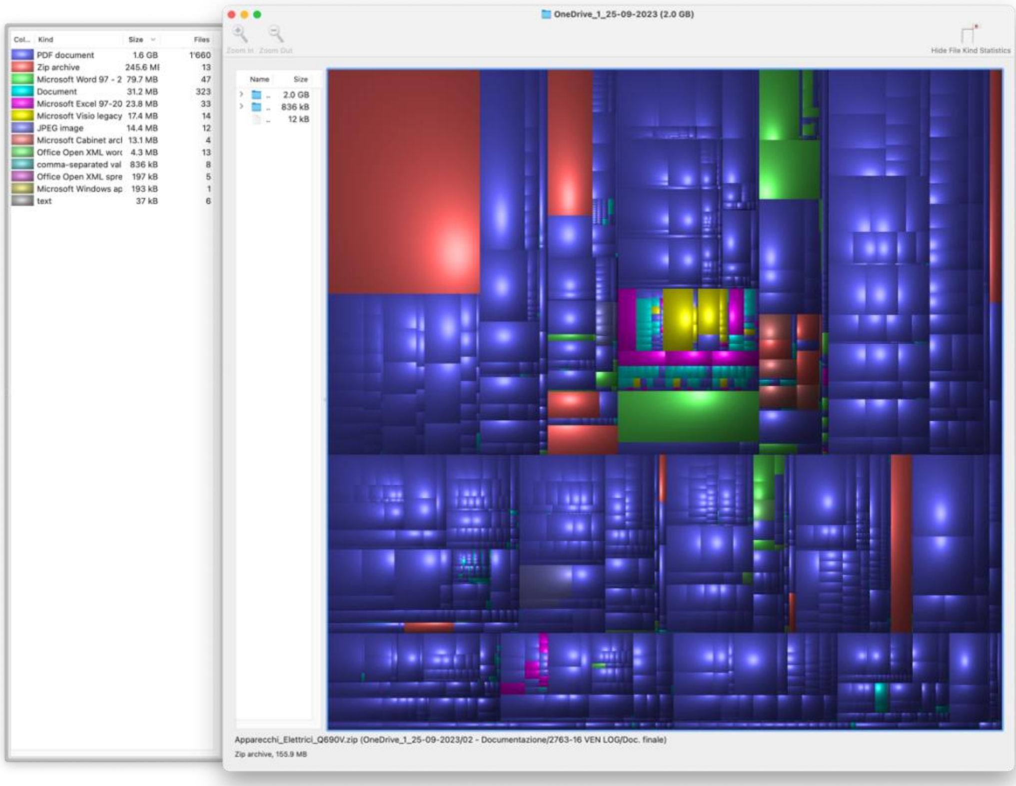

Figure 12: Treemap Visualization of the Mappo-Morettina Data Objects

Key Observations from the Treemap:

The Treemap functions as a visualization composed of nested rectangles. These rectangles represent certain categories within a selected dimension and are ordered in a hierarchy, or “tree.” Quantities and patterns can be compared and displayed in a limited chart space. Treemaps represent part to whole relationships.

1. **Dominance of PDF Files:** The bulk of the files are in **PDF format.** While highly suitable for human consumption and portable across different devices, they pose significant challenges when it comes to automated processing and data extraction. Machines cannot easily parse or analyze the contents of these documents without specialized tools or manual intervention, making it difficult to integrate the information into structured models or databases.
2. **Limited Number of Excel Files:** A relatively small proportion of the files are in **Excel format.** These files are particularly valuable as they often contain structured data that can be directly processed and analyzed. Excel files could potentially serve as a foundation for creating models or extracting relevant data

points. However, their limited presence in the dataset reduces the opportunities for automated data handling and forces reliance on less structured sources.

3. **Sparse Diagram Source Files:** There are also a few files in formats used by diagramming tools, such as **Visio.** These diagram files could theoretically provide valuable insights into system architecture or workflows. However, their scarcity means they cannot serve as a comprehensive source of visual models for the tunnel's infrastructure.

Additional Challenges Noted:
Beyond the file formats and their distribution, several other issues complicate the effective use of this dataset:

1. **File Changes timestamps:** A brief examination of file **timestamps** revealed significant variation, indicating that updates to the files occur in an **uncoordinated** manner. This lack of synchronization means that different parts of the documentation may be updated at different times, leading to inconsistencies and potential gaps in information. Without a clear update protocol, it becomes difficult to determine which files contain the most current and accurate data.
2. **Version Control:** Perhaps most critically, there is no evidence of a standardized **versioning mechanism** within the dataset. Proper version control is essential for tracking changes, ensuring consistency, and preventing the loss of important information. Without it, there is no systematic way to identify which documents are the latest, track the evolution of data over time, or revert to previous versions if needed. This lack of versioning significantly hampers efforts to create accurate and up-to-date models. FEDRO correctly enforces version tracking of the documents (each chapter of the final documentation must contain a version description normally enlisted in a MS Word document), but finally it is difficult to follow possible updates and corrections.

Implications for Modeling and Automation:
The current state of the data objects underscores the difficulty of using it as a reliable source for creating models, whether static or dynamic. The dominance of unstructured PDFs, coupled with inevitable updates and the complexity of version control, means that significant manual effort would be required to extract useful information. Automating this process is nearly impossible without first converting and organizing the data into more machine-friendly formats.

Recommendations:
To improve the usability of such datasets in the future, several steps should be considered:

- **Standardizing File Formats**: Encourage the use of structured, machine-readable formats such as Excel or XML for key data, while minimizing reliance on PDFs for critical information.
- **Implementing Version Control**: Introduce a versioning system to track document changes and ensure consistency across the dataset.
- **Centralizing Updates**: Establish a coordinated update protocol to ensure that all documents are updated in sync, reducing inconsistencies and confusion.
- **Automating Data Extraction**: Invest in tools and processes that can extract data from PDFs and convert it into structured formats, reducing manual workload.

In summary, while the dataset provides a wealth of information, its current state poses significant challenges for modeling and automation. Addressing these issues will be crucial for future projects that aim to create accurate and reliable digital models of tunnel infrastructure and operations.

We used the following process to reify the data:

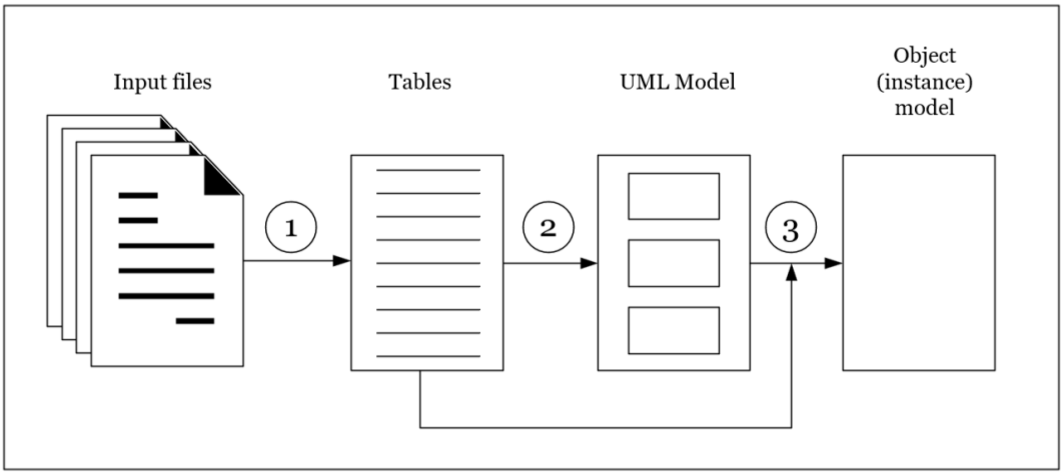

Figure 13: Data Reification Process

The process consists of:

- Manual analysis of the multitudes of files in the package, including PDFs and CSV files in order to identify the relevant entities that constitute the domain model.
- Coding of the domain model through the use of a single, very large Excel file, where the headers of the tables are the properties of the entities, and the rows represent the single instances of those entities.
- Creation of a single, large UML Class diagram encoding the whole domain model.
- Optional creation of an instance-level model of the tunnel (a so-called digital twin) by combining the class diagram with the information contained in the excel files.

In the following pictures, the overall complexity of the UML model can be observed. The first figure illustrates a section dedicated to 'Rooms', which is subsequently included in the second, extending the visualisation. Finally, the third figure further supplements the second, showing the entire UML schema. The images are provided for illustrative purposes only, with the sole purpose of highlighting the complexity of the entire model.

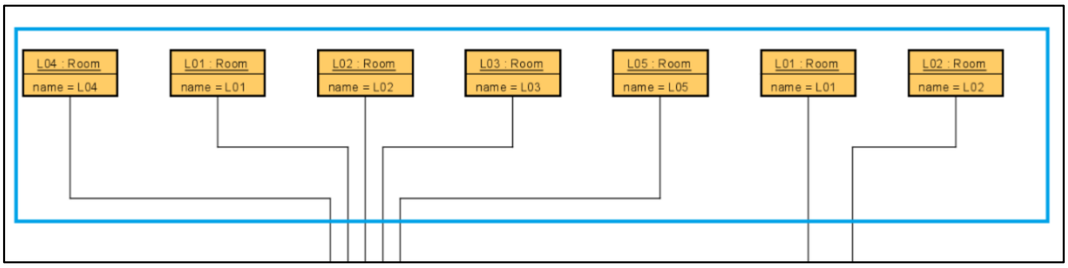

Figure 14: UML- View of the 'Room' Subsystem

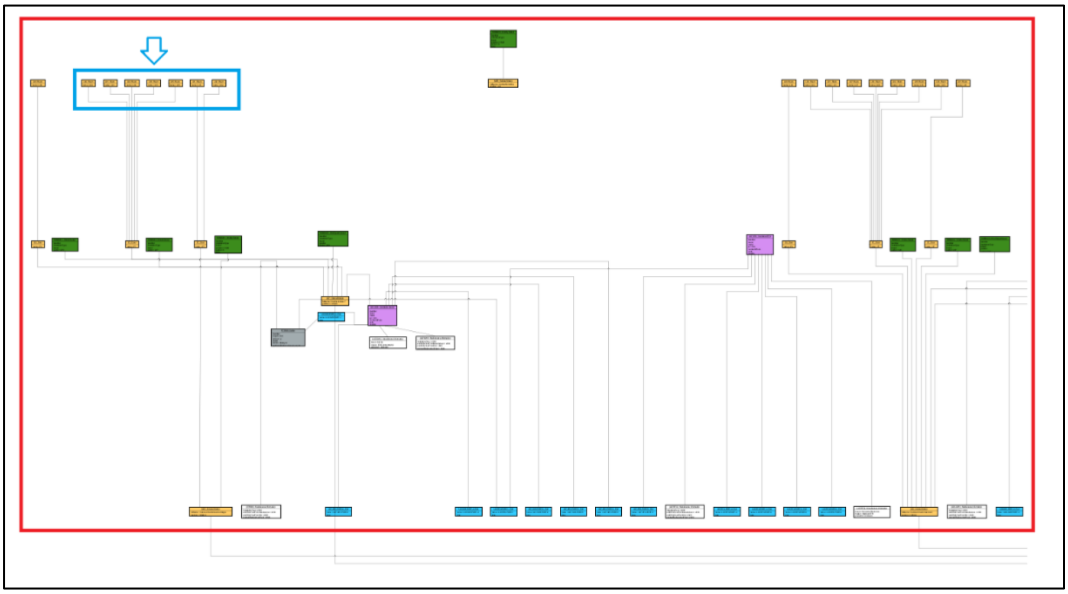

Figure 15: Overall Static Structure UML Class Diagram

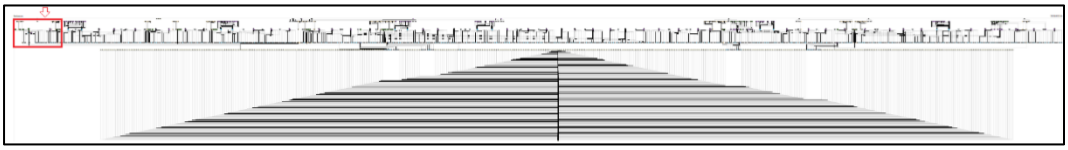

Figure 16: Overall Static Structure UML Class Diagram

UML class diagrams were created for the models of the single layers, for easier handling. In Figure 17 the SOS layer is represented.

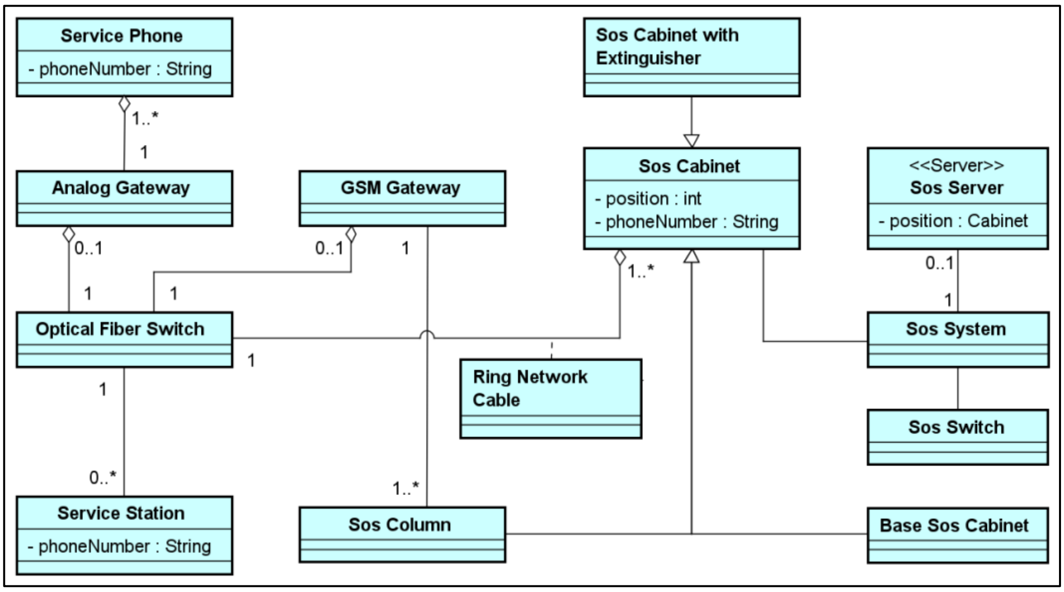

Figure 17: SOS Layer Model UML Class Diagram

In the following figure we see another example, in this case the ventilation layer.

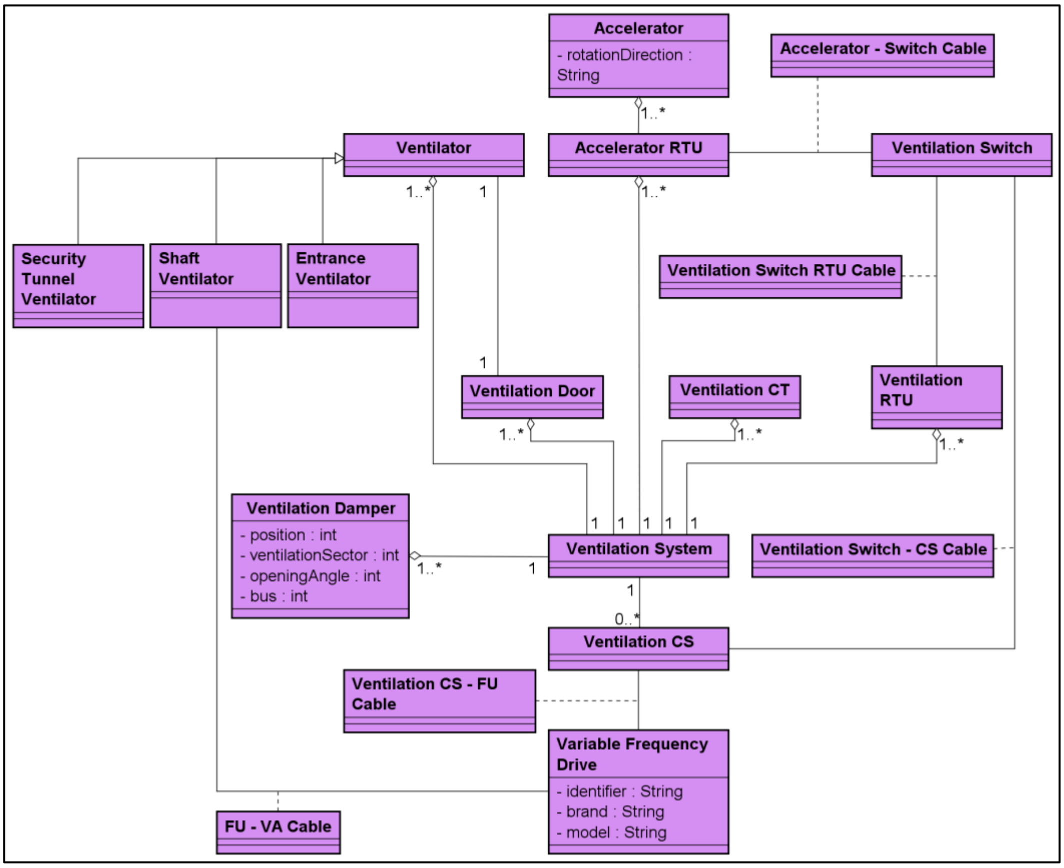

Figure 18: Ventilation Layer Model UML Class Diagram

4.3 Data management and possible data frameworks

4.3.1 IoT for tunnels

IoT systems are already widely used in various sectors to improve infrastructure management and safety. IoT systems are already being used to manage Smart-Cities where multiple aspects of traffic and safety can be monitored and managed. These systems enable operators to make quick and informed decisions, improving emergency management and optimising the use of resources.

By transferring these technologies to a digital tunnel, the benefits could be just as significant. Sensors installed along the tunnel would allow continuous data collection, monitoring parameters such as structural integrity, air quality and traffic flow in real time. In the event of anomalies or emergencies, the system would be able to automatically activate safety protocols, reducing the risk to users and improving the protection of the structure.

In addition to safety, an IoT system in the tunnel would improve operational efficiency. With real-time data available, operators could optimise the use of resources such as lighting and ventilation, adapting them to current conditions. Preventive maintenance would also be facilitated, allowing for appropriate intervention before failures or significant damage occurs.

4.3.2 Middleware

Middleware is a software component that acts as an intermediary between different applications or between an application and network resources, enabling them to communicate and share data efficiently. In practice, middleware provides a supporting infrastructure that simplifies the development, management, and interoperability of distributed systems, allowing applications to operate without having to directly handle the complexities of network communication, security, and data management.

In an IoT infrastructure such as a "digital tunnel," middleware plays a key role:

- Enables connection and message exchange among heterogeneous devices (e.g., sensors, actuators, control systems, supervision components).
- Manages authentication and authorization, which are vital for secure exchanges, especially in critical contexts like a road tunnel.
- Provides reliable communication services (e.g., guaranteed message delivery, possible persistence, topic- or queue-based management).
- Facilitates integration among multiple applications that need to share and process data in a coordinated way (monitoring software, maintenance systems, analytics platforms, etc.).

Below are some of the key aspects that middleware could provide in such projects, based on the basic features of a data management platform.

4.3.2.1 Connectivity

- **Definition:** Connectivity ensures that devices and systems can join the network and exchange information in real time. In a road tunnel context, this means ensuring each sensor (for air quality, visibility, temperature, etc.) can send data to the central platform, and that any actuators (ventilation, variable message signs, lighting) can receive commands.

- **Why it's important:** A digital tunnel relies on continuous data flows to monitor environmental and traffic conditions; without stable and robust connectivity, the system cannot respond quickly to potential anomalies or emergencies.
- **What might be sought:** The ability to handle lightweight protocols (e.g., MQTT), support secure connections (TLS/SSL), and maintain compatibility with resource-constrained devices.

4.3.2.2 Intelligence and identity

- **Definition**: This refers to the system's ability to extract knowledge from incoming data and to associate a unique identity with devices, helping track their state and behaviour over time.
- **Why it's important:** In a tunnel, knowing which sensor is sending data (and where it is physically located) makes it easier to diagnose issues and schedule targeted maintenance. Adding "intelligence" could also mean detecting anomalous patterns in real time.
- **What might be sought:** Identity management capabilities (authentication and permissions), integration with data processing and analysis modules, and the option to enrich messages with metadata (timestamps, coordinates, event types).

4.3.2.3 Scalability

- **Definition:** Scalability is the capacity of a system to accommodate increases (or decreases) in workload without degrading performance.
- **Why it's important:** The tunnel may need to handle variable data volumes over time (traffic peaks, new sensor installations). A scalable middleware allows adding or removing resources (nodes, brokers, databases) without redesigning the entire infrastructure.
- **What might be sought:** Clustering and load balancing support, message replication mechanisms, easy setup in containerized environments (e.g., Docker/Kubernetes), and robust performance with many connected clients.

4.3.2.4 Dynamism and self-adaptation

- **Definition:** Indicates the ability to react to changing conditions—shifts in the tunnel's environment, infrastructure modifications, network faults, and so forth.
- **Why it's important:** In a complex environment, the sensor and actuator ecosystem may evolve, and the middleware should handle failures or reassignments automatically, such as rerouting data flows if a node becomes unavailable.
- **What might be sought:** Failover mechanisms, automatic reconnection for devices, on-the-fly adjustments (e.g., QoS or transmission frequency), and dynamic configuration management.

4.3.2.5 Hybrid architecture

- **Definition:** The possibility of integrating products and components from different vendors, ensuring transparent communication across the IoT network.
- **Why it's important:** Tunnel systems typically include devices of various brands and generations (lighting, ventilation, cameras, SOS stations, etc.). A hybrid architecture allows these diverse technologies to coexist under a single communication paradigm.

- **What might be sought:** Support for multiple protocols (MQTT, AMQP, REST/HTTP, OPC-UA), the ability to create bridges between protocols, and compatibility with major SCADA/PLC systems.

4.3.2.6 Cybersecurity and data protection

- **Definition:** The set of mechanisms to prevent unauthorized intrusions, safeguard data confidentiality, and ensure system integrity (covering both message exchanges and user/device identification).
- **Why it's important:** A digital tunnel is a critical infrastructure. Any cyberattack could compromise traffic safety or expose sensitive information.
- **What might be sought:** Strong authentication (e.g., X.509 certificates), end-to-end encryption (TLS), role-based access control, and thorough logging of operations.

In the next chapters we analyse some examples of IoT framework available today.

4.3.3 Eclipse Mosquitto

4.3.3.1 Overview

Eclipse Mosquitto™ is an open-source MQTT broker widely used to enable efficient communication among IoT devices. Due to its lightweight design, it is particularly suited for scenarios with limited hardware resources, making it a popular choice for managing data transmission in constrained environments such as sensor networks. Mosquitto's small footprint and straightforward configuration process make it an ideal solution for projects requiring minimal overhead and rapid deployment, especially in contexts like home automation, industrial sensors, and microcontroller-based systems.

4.3.3.2 Key Features

Lightweight MQTT Broker:

- Focused on minimal resource usage, ideal for devices like sensors and microcontrollers.

Pub/Sub Model:

- Implements the publish/subscribe paradigm, promoting asynchronous, scalable communication.

Security Support:

- Offers TLS encryption and authentication via username/password or X.509 certificates.

Scalability:

- Can manage numerous simultaneous connections and supports clustering for higher throughput.

Cross-Platform:

- Runs on Linux, Windows, macOS, and ARM architectures, facilitating wide adoption.

MQTT QoS Compliance:

- Supports MQTT's various Quality of Service levels to fine-tune reliability.

4.3.3.3 Advantages & Limitations

Advantages:
- Open Source: Free and customizable.
- Efficient: Well-suited for low-latency or resource-constrained environments.
- Easy Setup: Simple to configure and integrate into existing systems.

Limitations:
- Basic Feature Set: Less sophisticated than some alternatives (e.g., RabbitMQ, Kafka).
- Limited User Management: Lacks advanced built-in role and user management capabilities.

4.3.4 RabbitMQ

4.3.4.1 Overview

RabbitMQ is an open-source message broker that relies on the Advanced Message Queuing Protocol (AMQP), enabling flexible and reliable communication among distributed systems. It provides a robust platform for queue management, supporting numerous routing patterns and transaction-like mechanisms. While it can be more complex than lightweight brokers, RabbitMQ excels in enterprise scenarios requiring advanced features, reliability, and extensive integration options.

4.3.4.2 Key Features

AMQP Protocol:
- Delivers advanced messaging functions such as acknowledgments, routing flexibility, and transactions.

Wide Language Support:
- Compatible with Java, Python, Ruby, and many more, easing integration across diverse ecosystems.

Scalability & High Availability:
- Offers clustering and federation to distribute workloads and maintain service continuity even if a node fails.

Advanced Queue Management:
- Provides options like message priority, TTL policies, and dead-letter exchanges to handle complex workflows.

Security:
- Supports TLS-encrypted communication, plus authentication and role-based authorization.

Plugin Ecosystem:
- Extensible via plugins (MQTT, STOMP, WebSockets) for diverse messaging scenarios.

4.3.4.3 Advantages & Limitations

Advantages:
- Robustness: Handles large message volumes without significant performance degradation.

- Flexibility: Supports multiple messaging models and can be extended through plugins.
- Mature Ecosystem: Well-established community, numerous libraries, and integrations.

Limitations:
- Complexity: Can be harder to configure and maintain, especially for large-scale systems.
- Overhead: AMQP's advanced features may introduce higher latency in resource-constrained setups.

4.3.5 Orion Context Broker

4.3.5.1 Overview

Orion Context Broker is an open-source component within the FIWARE ecosystem, designed for real-time context management in various IoT and smart-city applications. It implements the NGSI (Next Generation Service Interface) standard, allowing for storage, querying, and subscription to entities and their attributes. Orion's focus on context information sets it apart from traditional message brokers, making it a solid choice for projects that require semantic data modelling and event-driven updates across large-scale infrastructures.

4.3.5.2 Key Features

Context Information Management:
- Acts as a central hub for creating, updating, and retrieving entities, enabling real-time tracking of device states or application data.

NGSI-LD Support:
- Complies with the NGSI-LD API to handle linked data in IoT environments, easing interoperability.

Scalability:
- Designed to manage a high volume of entities and requests, suitable for city-wide or industrial deployments.

Real-Time Updates:
- Sends notifications when context changes occur, allowing systems to react quickly.

Ecosystem-Friendly:
- Integrates with FIWARE components and supports third-party systems through standardized interfaces.

Security:
- Ensures encrypted communications over HTTPS and provides mechanisms to safeguard data exchange.

4.3.5.3 Advantages & Limitations

Advantages:
- Real-Time Context Handling: Ideal for dynamically changing environments.
- Standards Compliance: NGSI ensures straightforward integration and interoperability.

- Scalable Architecture: Handles numerous entities and subscriptions efficiently.

Limitations:
- Learning Curve: Requires familiarity with NGSI standards and context-oriented data.
- FIWARE Dependency: Best leveraged in FIWARE projects; integration with other platforms may need additional effort.

4.3.6 ZeroMQ

4.3.6.1 Overview

ZeroMQ is an open-source messaging library designed for high-performance, asynchronous communication without relying on a central broker. By focusing on a lightweight, socket-based approach, it empowers developers to build custom messaging patterns—ranging from simple request/reply to complex pub/sub topologies—across distributed systems. ZeroMQ's low-level API offers exceptional flexibility, albeit with some added complexity in implementation.

4.3.6.2 Key Features

Peer-to-Peer Messaging:
- Eliminates a central broker, enabling direct communication and reducing single points of failure.

High Performance:
- Uses asynchronous I/O and efficient buffering to achieve minimal latency and high throughput.

Multiple Patterns:
- Supports request/reply, pub/sub, push/pull, and more, letting developers tailor solutions to specific needs.

Cross-Platform & Multi-Language:
- Works on various operating systems and supports numerous programming languages.

Lightweight & Embeddable:
- Features a small footprint suitable for embedded devices or resource-constrained systems.

Automatic Connection Handling:
- Manages reconnections, retries, and queuing during brief network interruptions.

4.3.6.3 Advantages & Limitations

Advantages:
- Low-Latency: Ideal for real-time workloads.
- Flexibility: Adaptable to diverse communication scenarios.
- No Central Broker: Reduces overhead and simplifies certain designs.

Limitations:
- No Built-In Persistence: Requires custom solutions for storing messages long-term.
- Learning Curve: Developers must handle many details manually.

- Minimal Feature Set: Advanced routing, security, or management features are not included by default.

4.3.7 HiveMQ

4.3.7.1 Overview

HiveMQ is an enterprise-level MQTT broker built for secure, scalable, and robust communication between IoT devices and enterprise systems. Optimized for large-scale deployments, it can manage millions of concurrent clients while providing high availability and in-depth monitoring. Its extension system and integration capabilities make it particularly suited to business-critical or mission-critical IoT environments.

4.3.7.2 Key Features

Enterprise-Grade Scalability:
- Orchestrates numerous MQTT clients and messages, expanding horizontally with clusters to maintain consistent performance.

High Availability:
- Uses clustering to balance loads and ensure continuous operation even if individual nodes fail.

Security:
- Implements TLS encryption, authentication via usernames/passwords or certificates, and OAuth2 support, plus fine-grained access controls.

Persistent Sessions:
- Retains client session state for offline periods, preventing data loss and easing reconnections.

Flexible Integration:
- Offers APIs and extension mechanisms for seamless linkage with enterprise systems, analytics, and cloud platforms.

Monitoring & Management:
- Features a web-based dashboard for real-time insights, plus compatibility with external tools like Prometheus and Grafana.

4.3.7.3 Advantages & Limitations

Advantages:
- Massive Scalability: Effectively handles millions of devices, suitable for demanding enterprise use.
- High Reliability: Designed for fault tolerance and minimal downtime.
- Robust Security: Multiple encryption and authentication options.
- Integration Friendly: Connects well with existing corporate or cloud platforms.

Limitations:
- Higher Cost: Commercial or enterprise offerings can be expensive for smaller teams.
- Complex Configuration: Advanced clustering and security features may require specialized knowledge.

4.3.8 Apache Kafka

4.3.8.1 Overview

Apache Kafka is a distributed event streaming platform renowned for handling high-throughput, low-latency data pipelines. Originally open-sourced by LinkedIn, Kafka is widely adopted for real-time analytics, streaming applications, and large-scale data integration. Its log-based architecture and replication model ensure fault tolerance, making Kafka an ideal choice for organizations processing substantial data flows in real time.

4.3.8.2 Key Features

High Throughput:
- Capable of processing millions of messages per second for large-scale data ingestion.

Distributed & Fault Tolerant:
- Clustering, partitioning, and replication ensure no single point of failure.

Event Streaming:
- Maintains an append-only log of events, allowing real-time processing and replay by multiple consumers.

Durability:
- Persists messages on disk with replication across brokers to safeguard against data loss.

Flexible Messaging Models:
- Supports publish/subscribe and point-to-point, meeting varied streaming and queuing needs.

Stream Processing Integration:
- Integrates seamlessly with Apache Spark, Apache Flink, and Kafka Streams to handle real-time transformations.

Rich Ecosystem:
- Provides connectors to databases, cloud services, and many other systems, expanding deployment possibilities.

4.3.8.3 Advantages & Limitations

Advantages:
- Highly Scalable: Designed to grow horizontally with demand.
- Reliable Data Persistence: Replicated logs prevent message loss.
- Low Latency: Suited for real-time analytics and high-speed data flows.
- Extensive Tools: Strong ecosystem for management and integration.

Limitations:
- Complex to Operate: Requires careful cluster management, monitoring, and tuning.
- Resource Intensive: Needs substantial storage and CPU resources, potentially overkill for smaller tasks.
- Monitoring Overhead: Additional solutions (e.g., Prometheus) often required for robust alerting and metrics.

4.3.9 Redis

4.3.9.1 Overview

Redis is an open-source, in-memory data structure store (BSD-licensed) widely used as a key-value database, cache, and basic message broker. Renowned for its high-speed data processing, Redis can handle a variety of data structures—strings, lists, hashes, sets, sorted sets, and more—while providing built-in replication, scripting, and clustering features. Although it can operate in a broker-like capacity, its primary design centres on in-memory storage and rapid data operations, making it ideally suited for use cases where persistence is less critical and low latency is dominant.

4.3.9.2 Key Features

In-Memory Data Store:

- Operates directly in main memory, enabling extremely fast read/write operations.
- Versatile Data Structures
- Supports strings, hashes, lists, sets, sorted sets, bitmaps, hyperlogs, geospatial indexes, and streams.

Pub/Sub Support:

- Offers publish/subscribe capabilities introduced in Redis 5.0, allowing one-to-many communication for messaging scenarios.

Persistence Options:

- Primarily in-memory but can offload data to disk/DB for snapshotting or append-only file (AOF) persistence, though not as robust as specialized brokers.

Scalability & High Availability:

- Provides Redis Sentinel for high availability and Redis Cluster for sharding, ensuring horizontal scalability and failover.

Advanced Caching & Tooling:

- Features Lua scripting, LRU eviction policies, and integrated mechanisms for caching and real-time data processing.

4.3.9.3 Advantages & Limitations

Advantages:

- Speed & Scalability: In-memory approach delivers exceptionally high throughput, suitable for short-lived messages and real-time applications.
- Ease of Use: Straightforward to configure, deploy, and integrate as both a cache and key-value store.
- Rich Data Handling: Advanced caching, data structures, and streams enable versatile development patterns.

Limitations:

- Not Primarily a Message Broker: Lacks advanced routing features commonly found in specialized brokers like RabbitMQ or Kafka.
- Less Suitable for Large Messages: Performance may degrade when handling bigger payloads.
- Limited Persistence: Although disk persistence exists, it is not as robust or granular as in dedicated messaging solutions, making data durability less guaranteed.

4.3.10 MQTT Protocol – Communication between IoT and devices

The MQTT protocol adopts a Publish/Subscribe paradigm, where the sender (publisher) and the receiver (subscriber) of messages communicate through channels known as "topics," remaining independent of each other. The connection between them is managed by a broker, whose primary role is to filter incoming messages and correctly distribute them to the interested subscribers. In this model, clients do not need to actively search for the information they require; the broker automatically provides the data to clients as soon as it becomes available.

The MQTT broker is the central element of any architecture based on this protocol. Depending on its implementation, a broker can simultaneously handle up to millions of connected MQTT clients. The broker is responsible for receiving all messages, filtering them, determining which clients are interested in each message, and sending them to the appropriate recipients. This type of architecture allows clients to be decoupled, ensuring a highly scalable solution without direct dependencies between data producers and consumers.

At the core of MQTT's functionality are the broker and MQTT clients. The broker manages the transmission of messages between the sender and designated recipients. An MQTT client can publish a message to a broker, and other clients can subscribe to this broker to receive published messages. Each MQTT message is associated with a specific topic: clients publish their messages on certain topics and subscribe to the topics from which they wish to receive information. The broker uses these topics and subscriber lists to route messages to the relevant clients.
Furthermore, an MQTT broker can temporarily store messages intended for clients that are not connected, a particularly useful feature in situations where network connections are unstable.
The main benefits of the MQTT protocol include:
- Lightweight and efficient, minimizing client resource usage and network bandwidth.
- Support for bidirectional communication between devices and servers, with the ability to send messages to groups of devices.
- Scalability to millions of devices.
- Definition of Quality of Service (QoS) levels to ensure message reliability.
- Support for persistent sessions between device and server, reducing reconnection times on unreliable networks.

MQTT uses *topics*—hierarchical strings that define routing paths for messages. Publishers post messages under certain topics, and subscribers receive messages from the topics to which they are subscribed. Some examples of topics might look like:
- myhome/groundfloor/livingroom/temperature: Temperature readings in the living room of a house.
- 5f...xyz/status: Status updates for a particular device or sensor.

Topics are **case-sensitive** and can include multiple levels separated by slashes (e.g., `building1/floor2/room3/light`), allowing a clear structure for organizing data from numerous devices.

4.3.10.1 Wildcards in MQTT

To simplify subscription management across large sets of similar topics, MQTT supports the use of wildcard characters in subscriptions:

+ **(single-level wildcard)**: Matches exactly one level in the topic hierarchy.
Example:

- ✓ `myhome/groundfloor/+/temperature` will capture `myhome/groundfloor/kitchen/temperature` or `myhome/groundfloor/livingroom/temperature`.

(multi-level wildcard): Matches one or more levels, typically at the end of a topic.
Example:

- ✓ `myhome/groundfloor/#` subscribes to everything under `myhome/groundfloor`, such as `myhome/groundfloor/kitchen/temperature` or `myhome/groundfloor/livingroom/lights/status`.

While the multi-level wildcard (#) allows broad subscriptions, it may become impractical if the broker handles a massive number of topics, potentially overwhelming a single subscriber with excessive data. In scenarios with very large datasets, broker extensions or filtering mechanisms are often implemented to keep data flows manageable.

MQTT also reserves certain *system topics* that begin with `$`. These are commonly used for broker-level information, such as statistics or diagnostics (e.g., `$SYS/broker/clients/connected`). These topics are not normally accessible to clients unless explicitly permitted.

By adopting this Publish/Subscribe model, structuring messages through well-defined topics, and enabling flexible wildcard subscriptions, MQTT delivers a streamlined, highly adaptable messaging system suited for IoT deployments and resource-constrained environments.

4.3.10.2 Quality of Service (QoS) – Givin priority to communication

The MQTT protocol provides three different Quality of Service (QoS) levels to control message delivery between publishers and subscribers. The choice of a QoS level in an IoT system—such as a digital tunnel—impacts both performance and reliability:

- QoS 0: At most once delivery
- QoS 1: At least once delivery
- QoS 2: Exactly once delivery

Each level entails different trade-offs among reliability, latency, and network overhead. When a publisher sends a message, both the sender and the receiver must agree on the QoS level: this choice determines the communication flow and the acknowledgment mechanism required to ensure (or not) successful delivery.

In the following paragraphs, we will detail the three QoS levels and offer practical advice on how and when to use them in a digital tunnel context.

4.3.10.3 QoS 0: At most once delivery

The lowest level, QoS 0, provides a best-effort delivery mechanism where the sender does not expect any acknowledgment or guarantee of message delivery. The recipient does not confirm receipt, and the sender neither stores nor retransmits the message. Commonly known as "fire and forget," QoS 0 works similarly to the TCP protocol, where the message is sent without further checks or confirmations.

Figure 19: QoS 0: At most once Delivery

Additional insights and advice:

- **Minimal overhead:** With no acknowledgment or storage mechanism, network and computational impact is very low.
- **Possible losses:** In case of network congestion or connection issues, messages may be lost.

When to use it:

- For sensors sending very frequent data (e.g., temperature, humidity) where the occasional loss of a reading is not critical.
- In scenarios with extremely limited network or computational resources, where full reliability is not required.

4.3.10.4 QoS 1: At least once delivery

QoS 1 in MQTT focuses on ensuring that the message is delivered at least once to the recipient. When a message is published with QoS 1, the sender retains a copy of the message until it receives a PUBACK packet from the recipient, confirming that the message has been received. If the sender does not receive the PUBACK packet within a reasonable time, it retransmits the message to ensure delivery. Once the message is received, the recipient can process it immediately. For instance, if the recipient is an MQTT broker, it distributes the message to all subscribed clients and responds with a PUBACK packet to confirm message reception.

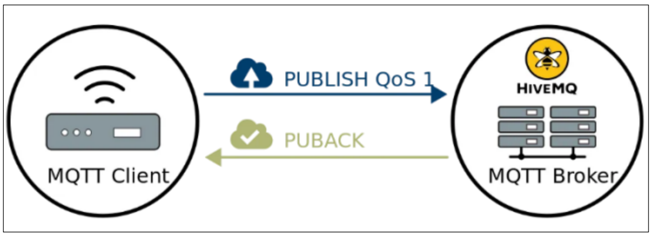

Figure 20: QoS 1: At least once delivery

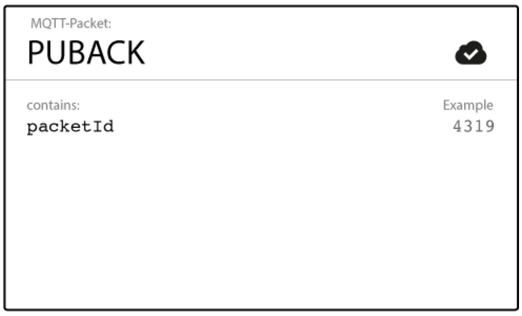

Figure 21: QoS 1: PUBACK

Additional insights and advice:

- **Possible duplicates:** If the acknowledgment is delayed or lost, the sender may retransmit the message, so the recipient might receive it multiple times.
- **More reliable than QoS 0:** Ensures that the message arrives, eventually, at least once; this is useful for important data that should not be lost.

When to use it:

- For status notifications, anomaly warnings, or monitoring data where receiving a duplicate is better than missing a message.
- The subscriber (or broker) should be capable of handling duplicates in an “idempotent” manner.
- Impact on network and computation: Moderate, since some acknowledgment packets (PUBACK) need to be exchanged.

4.3.10.5 QoS 2: Exactly once delivery

The highest level, QoS 2, ensures that the message is transmitted exactly once between the sender and the recipient. The process involves multiple steps:

- Publish: The publisher sends a message to the broker.
- PUBREC (Publish Received): The broker receives the message and confirms initial receipt to the publisher.
- PUBREL (Publish Release): The publisher sends a PUBREL message to the broker, confirming that communication with the subscriber can begin.
- PUBCOMP (Publish Complete): The broker sends a PUBCOMP message to the publisher to confirm that the delivery is complete.

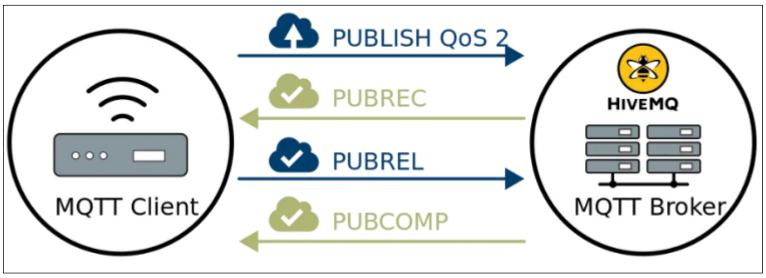

Figure 22: QoS 2: Exactly once delivery

Additional insights and advice:

- **No duplication:** The multi-step handshake prevents receiving the same message more than once.
- **Maximum reliability:** The ideal choice for critical commands (e.g., turning devices on/off, activating safety equipment) where neither duplicates nor missed deliveries are acceptable.
- **Overhead and latency:** This level requires the highest number of control packets, increasing both latency and resource consumption (for the client and the broker).

When to use it:

- Sending vital commands or data that must be unique and guaranteed to arrive.
- In scenarios where a duplicate could cause serious issues (e.g., a command accidentally triggered twice).

4.3.11 FEDRO Equipment Data Model

Goal of UML modelling covered by this project is a standardized description of tunnel elements. An important aspect of the process is the standardization of the type of information exchanged between the controllers belonging to different hierarchical levels, as well as the terminology used to describe that information.

Information between controllers is typically exchanged by means of software datapoints and often differs depending on historical or geographical factors (different Territorial Units may have different information needs).

With the aim of achieving information standardization, FEDRO's guideline 13032 defines object classes to which well-defined attributes are associated.

The object classes are determined on basis of the existing AKS-CH codification system governed by FEDRO's guideline 13013. In the object class definition, the information exchanged with the lower and higher hierarchical level is indicated and is categorized by means of 4 signal classes (alarms – A, failures – S, notifications – M and commands – B).

The goal of FEDRO's guideline 13032 is to standardize the content of the information exchanged and does not define interface requirements.

An example of object class according to guideline 13032 is depicted in the following figure.

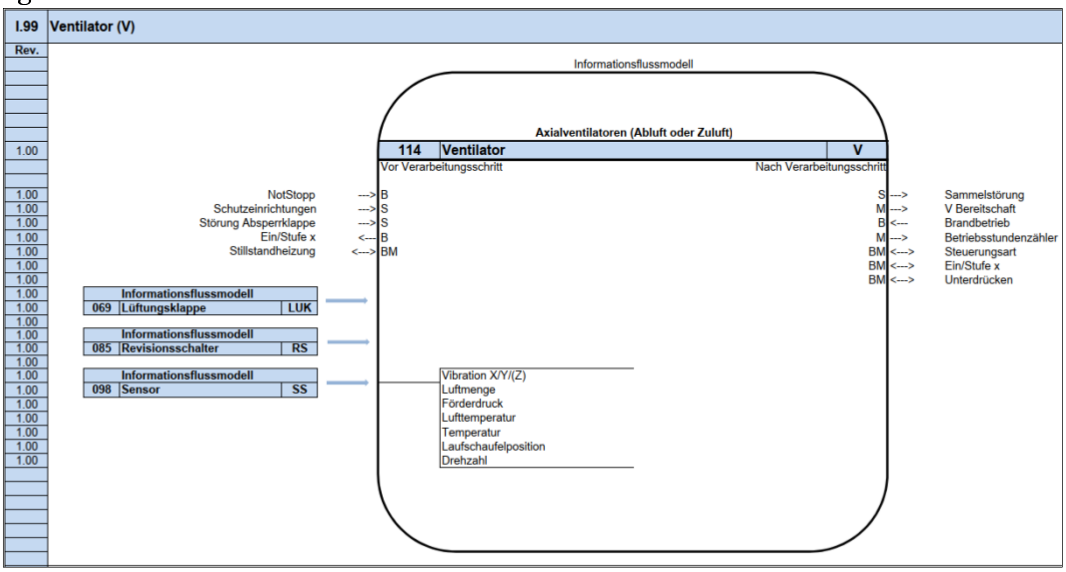

Figure 23: Ventilator object class according to FEDRO's guideline 13032 (German text).

In order to elaborate an UML model in compliance with relevant FEDRO's guidelines, the mentioned data model has been used (where applicable).

It is noted that the guideline will be composed of 6 parts and at the moment of the elaboration of the present project, only 2 parts are available.

4.3.12 Relationships between Middleware objects

Since the implementation of a middleware is not in the scope of this project, literature study has been performed with the aim of identifying relationships structures common to multiple existing data models.

The goal was to assess feasibility of using existing data models instantiate the meta-model implemented through UML.

Relationship description shall ensure enough flexibility to allow for model changes. Addition of new relationships or removal of existing ones shall be possible within the framework in an automized and efficient way.
The description of relationships as entries/attributes stored directly withing the object declaration should be avoided (option 1 in Figure 24). For flexibility purposes, declaration of relationships as separate entities/attributes shall be preferred (option 2 in Figure 24): in case of modification of properties for existing relationships, approach 1 would require checking each object for the presence of the field corresponding to the relationship to be modified, leading to potentially high computation effort and times. With option 2 modification of the relation type can be performed directly within the relationship object, allowing therefore to reduce the number of processes needed.

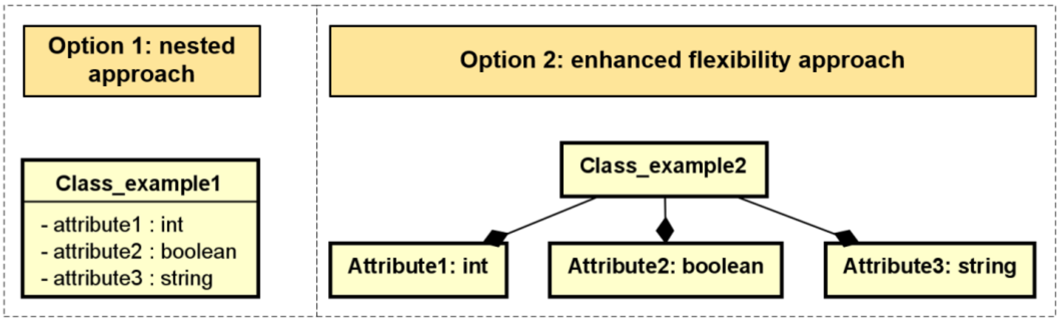

Figure 24: Relationship description approaches.

4.3.13 OPC-UA Data Information Model

OPC UA data information model is an open data model regulated by the OPC UA Foundation's specifications. OPC UA relies on object programming techniques and defines structure and semantics to be used for the exchange of data.
Objects are represented as Nodes in the OPC UA address space, which are described by means of attributes and references (relationships with other nodes).

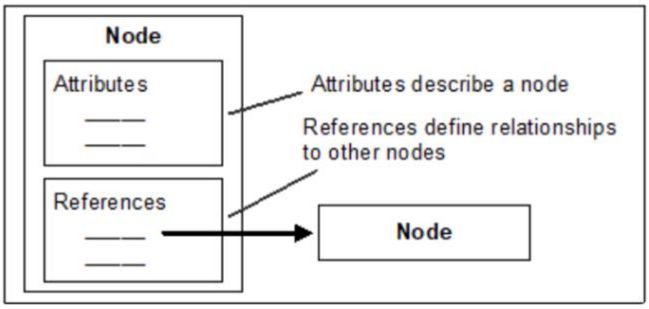

Figure 25: OPC UA Address Space Model (source: OPC Foundation)

References are classes themselves and are divided into hierarchical and non-hierarchical ones. An overview of the reference class structure is given in the following figures.

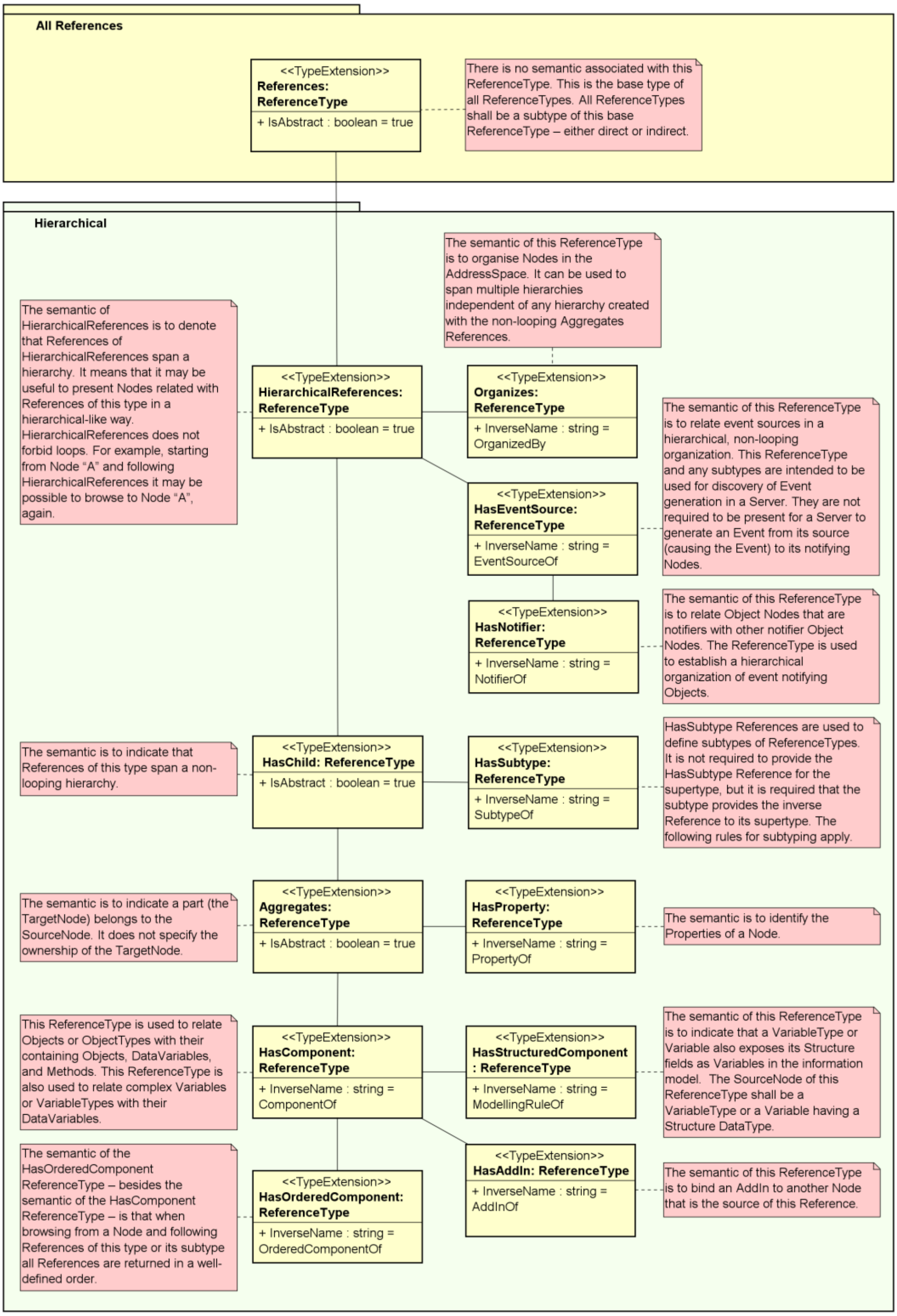

Figure 26: Hierarchical references, OPC UA data model

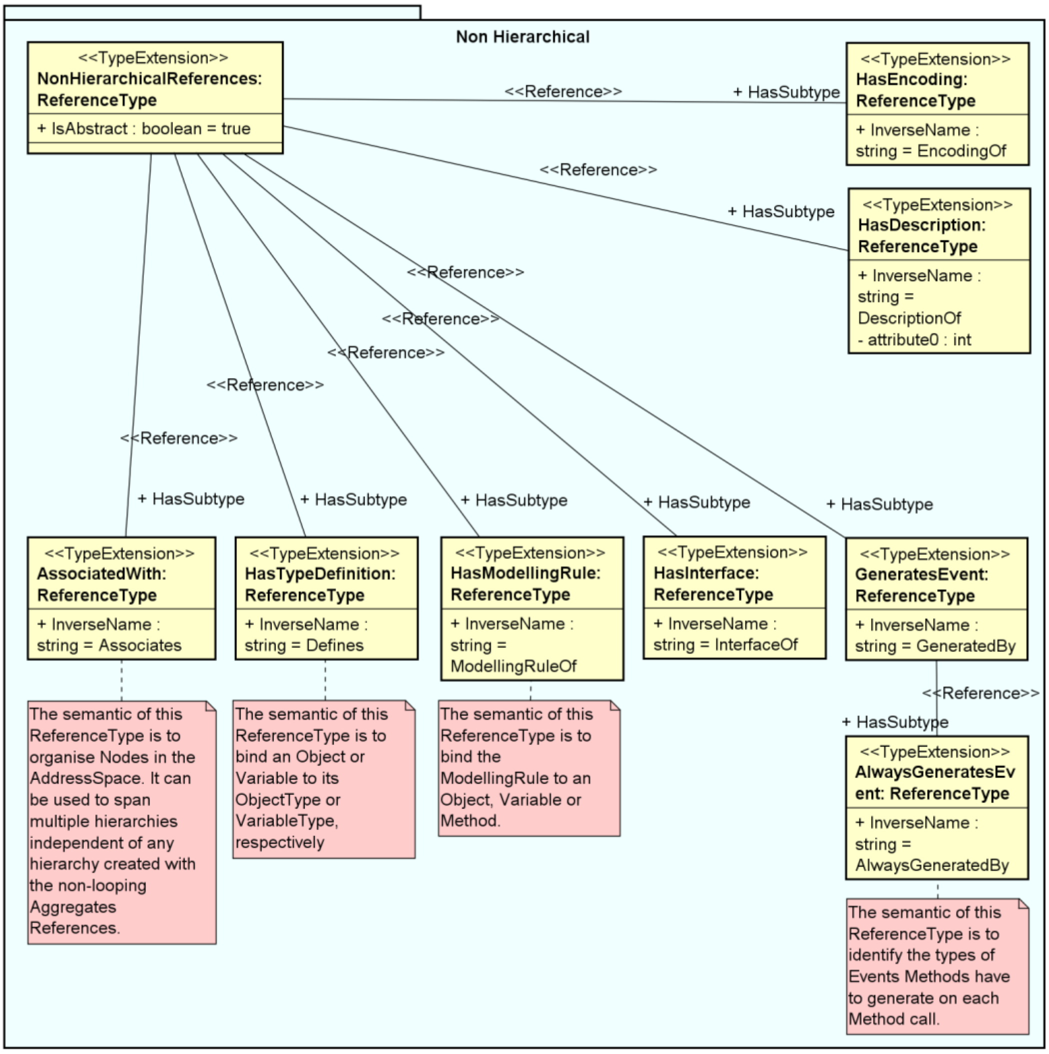

Figure 27: Non-hierarchical references, OPC UA data model

OPC-UA Data Model is well suited for the representation of data points according to FEDRO 13032 guideline.

4.3.14 IFC Data Model

Industry Foundation Class (IFC) is an open data model used as exchange format between Building Information Modeling Softwares (BIM). IFC makes use of simple ASCII files, making it a great exchange format in terms of interoperability. These files are imported and managed by IFC compliant software. IFC was developed by buildingSMART, an international organization with the aim of improve interoperability between software applications.

IFC relies on the object-oriented programming strategy and describes each item as object class. The available IFC classes are predefined and regulated by the Industry Foundation Class Specifications. In compliance with the object-oriented programming structure, inheritance of attributes between classes and subclasses is a key element of IFC data model. It is noted that not every class can be instantiated as object, but rather some classes serve as inheritance purposes only (these are denoted as abstract classes).

The IFC specification uses a unique identifier for object instances that follows the universal unique identifier standard (UUID).
The IFC relies on four conceptual layers (definitions according to *buildingSMART International*), which are summarized in Figure 28.

- Domain/module layer: includes all individual schemas containing resource definitions, those definitions do not include a globally unique identifier and shall not be used independently of a definition declared at a higher layer.
- Interoperability layer: includes schemas containing entity definitions that are specific to a general product, process or resource specialization used across several disciplines, those definitions are typically utilized for inter-domain exchange and sharing of construction information.
- Core layer: includes the kernel schema and the core extension schemas, containing the most general entity definitions, all entities defined at the core layer, or above carry a globally unique id and optionally owner and history information.
- Resource layer: includes schemas containing entity definitions that are specializations of products, processes or resources specific to a certain discipline, those definitions are typically utilized for intra-domain exchange and sharing of information. In this picture the 'tunnel' domain is transparent since this is not part of the current IFC 4.3 yet but is aimed to be included in a future update.

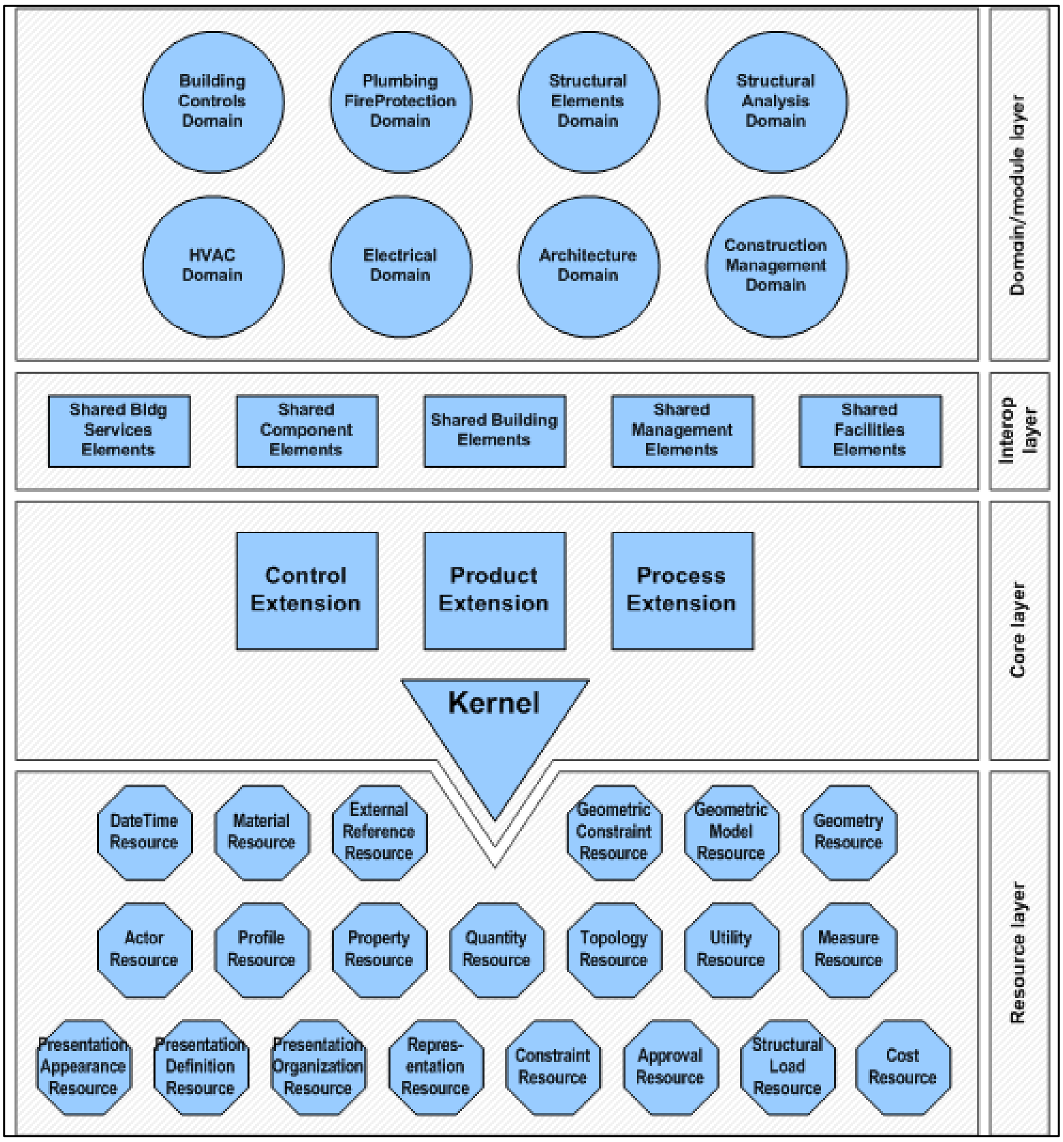

Figure 28: IFC architecture, source: buildingSMART International.

Attributes are defined as *a unit of information within an entity, defined by a particular type or reference to a particular entity* (buildingSMART definition).
IFC foresees four types of attributes:

- Simple direct attributes: attributes pointing to simple data types, which are directly declared within the class (ex: name of the object).
- Inverse attributes: describe the opposite relation between an object on one side and a relationship entity on the other side. They are calculated from inverse relationships and are not explicitly stored in the entity definition.
- Complex direct attributes: reference to another entity which stores attributes (ex: an object can refer to the class IfcOwnerHistory, which gives access to the attributes *LastModifiedDate* and *CreationDate*).
- Derived attributes: unit of information computed from other attributes using e defined expression.

Relationships between classes are defined by an intermediate class (*IfcRelationship* class), which contains the description of the relationship type as attribute. Since relationships are object classes themselves, these are often called objectified relationships. Relationships are realized by means of inverse attributes. For example: in order to describe a wall with a void, the object IfcWall is pointed by the relationship object IfcRelVoidsElements, which has two attributes: *RelatingBuildingElement* and *RelatedOpeningElement*. The attribute *RelatedOpeningElement* references to the Void element whereas the *RelatingBuildingElement* references to the Wall itself. The attributes refer to the object by means of the GUID. The object IfcRelVoidsElements is able to associate to which wall the void belongs to through the use of the stored attributes.

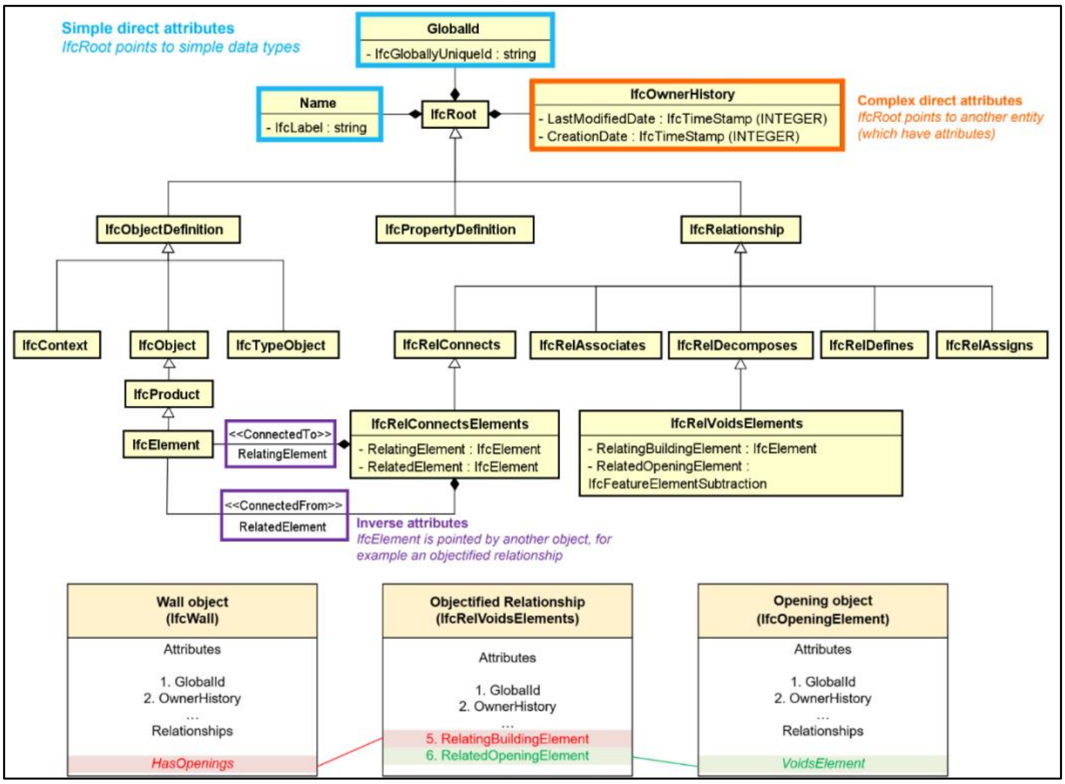

Figure 29: Examples of IFC attributes.

The use of intermediate objectified relationships by means of GUID may be suitable to guarantee the needed flexibility in terms of model modifications inside the IoT-Frameworks.

4.4 Conclusions of WP2

The outcomes of this work package reveal that, on one hand, the modeling of the static structure of a road tunnel is less complex than initially anticipated, particularly once the desired level of granularity for the required information is clearly defined. Given that a tunnel is fundamentally a long-term, low-entropy structure as a whole, the modeling process becomes relatively straightforward. When the upstream information is appropriately pre-processed to allow for automatic ingestion, the resulting models

can be established with minimal ongoing maintenance. This is due to the inherent stability of the tunnel's static structure.
However, the situation becomes notably more intricate when the level of granularity is extended to include information related to Internet of Things (IoT) components. In this domain, the complexity increases significantly, as does the potential for innovation. Unlike the static elements of a tunnel, which remain relatively unchanged over time, the hardware and IoT components undergo frequent updates and modifications, introducing an additional layer of complexity that requires careful consideration.

Furthermore, as will be explored in the subsequent chapter, the modeling of the static structure is not merely a standalone endeavor but serves as an essential prerequisite for the modeling of the dynamic structure. The dynamic aspects of the tunnel, which include factors such as traffic flow, environmental monitoring, and sensor data, depend heavily on the accurate representation of the static structure as their foundation. Therefore, the successful modeling of the static structure sets the stage for further advancements in dynamic modeling, highlighting the critical role of this initial step in the broader framework of tunnel management and monitoring.

5 WP3: Dynamic: BPMN for Maintenance & Operation Tunnel Modelling

5.1 BPMN modelling for dynamic procedures

This work package focuses on employing a modeling language to accurately represent the dynamic aspects of a tunnel, particularly in terms of maintenance and operational procedures. The primary objective was to utilize BPMN, the Business Process Modeling Notation, to formalize and visualize these processes.
However, an immediate challenge arose: the maintenance and operational procedures were not explicitly documented within the tunnel's existing records. Due to the nature of tunnel equipment supplies, maintenance descriptions are contained in the documentation of each part of the equipment normally supplied by different contractors.
As a result, extracting and interpreting this information required a careful, detailed analysis of the available documents.

Two specific maintenance use-cases were initially analysed. This provided a preliminary understanding of how BPMN can be applied in this context and helped to highlight its potential for capturing and structuring the complex, often implicit workflows associated with tunnel management.

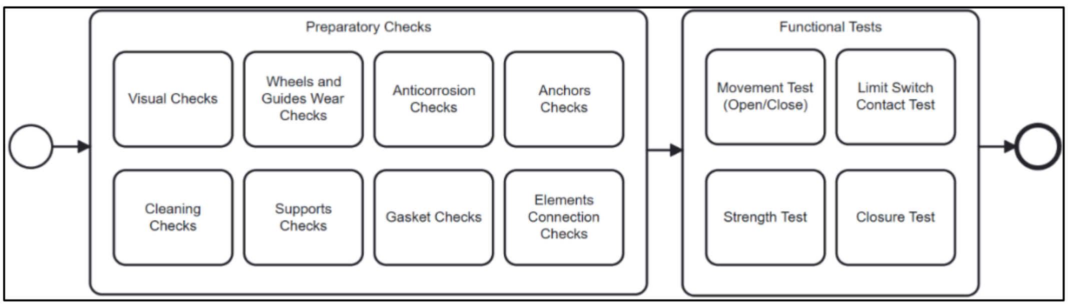

Figure 30: BPMN Process: Door Maintenance

A preliminary example is illustrated in the figure above, focusing on the maintenance procedures for doors within the tunnel system. The process shown outlines a straightforward sequence of steps involved in door maintenance. However, upon closer examination, it becomes evident that the use case presented is relatively basic and lacks complexity. It merely consists of a linear arrangement of tasks, with no branching paths, decision points, or conditional logic to account for varying scenarios or potential contingencies.

Due to this simplicity, it is clear that BPMN's capabilities are not fully utilized in this instance. BPMN excels in modeling complex workflows that involve intricate decision-making, parallel processes, and event-driven tasks. In contrast, for a simple, sequential process like this one, the full power and flexibility of BPMN are not required, as there are no opportunities to showcase its strengths in handling more dynamic or conditional workflows. Consequently, while this example serves as an introduction, it highlights the need for more complex cases to better demonstrate BPMN's potential.
The following figure shows a better example, which deals with how phone calls are routed.

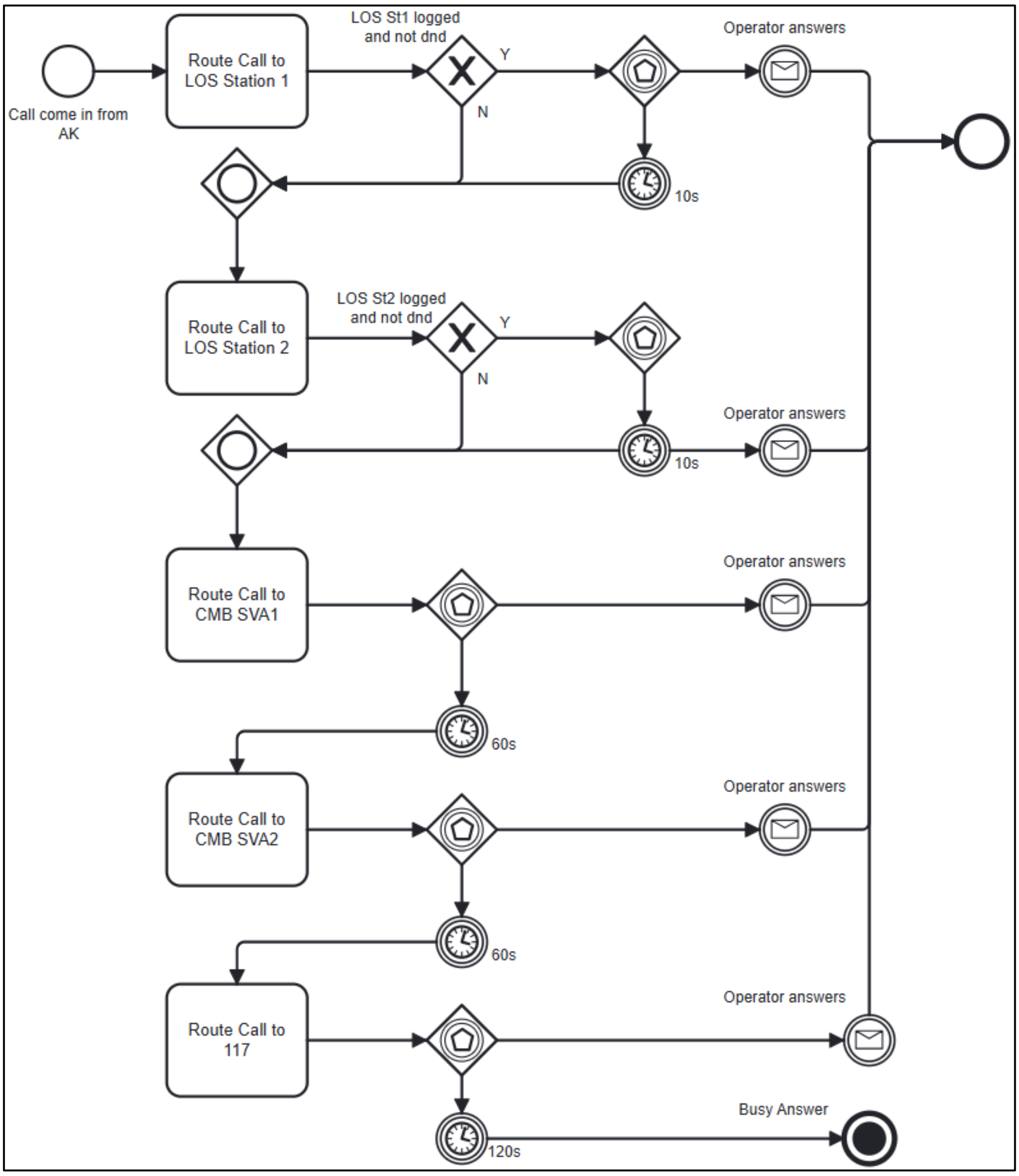

Figure 31: BPMN Process: Phone route call

Figure 32: BPMN Elements

As observed, the process under consideration incorporates elements that explicitly address time as a fundamental component of the model. This inclusion positions time not as an afterthought but as a first-class citizen within the process framework, allowing for precise scheduling, deadlines, and temporal constraints to be modeled directly. In this context, BPMN proves its value, demonstrating its ability to capture and represent such temporal dynamics effectively.

However, this raises an important question: is this level of granularity truly necessary? While the model's detail may enhance clarity and provide a comprehensive view of the process, it is worth considering whether such precision is essential, particularly in scenarios where human judgment and flexibility play a crucial role. Many of these processes, despite being modeled rigorously, are ultimately interpreted and executed by human operators, who may adapt or modify steps based on situational needs. As a result, the balance between formal modeling and practical execution remains a point of discussion, especially when human discretion is a key factor in process outcomes.

5.2 UML modelling for dynamic activities

The exploration of BPMN highlighted an important observation: in cases where the problem at hand lacks significant complexity, employing a robust and complex notation language like BPMN may not be warranted. While BPMN is powerful and well-suited for modelling processes with complex workflows, decision points, and multiple interacting elements, its use can become excessive for scenarios that are straightforward. In simpler cases, the overhead associated with BPMN's detailed structure might lead to unnecessary complexity in the modelling process.
Given this insight, it can be stated that UML represents a valid modelling tool for a majority of dynamic activities and might offer a more practical solution. UML provides various diagram types, such as sequence diagrams, state machine diagrams, and activity diagrams, which are designed to represent dynamic behaviours effectively. We hypothesized that these diagrams could capture the essential aspects of the processes within the given context without introducing the additional complexity that BPMN entails. By investigating UML's capabilities, we aimed to determine whether it could

offer a balance between simplicity and sufficient detail, making it a more appropriate choice for modelling processes that do not require the full power of BPMN.

Further research on applying UML to dynamic modelling was undertaken, deriving for example experience from the software tool that was developed for the maintenance of the local canton roads (include Mappo-Morettina before it shifted to FEDRO). The following figure introduces a general model for the maintenance activity of the ventilation equipment:

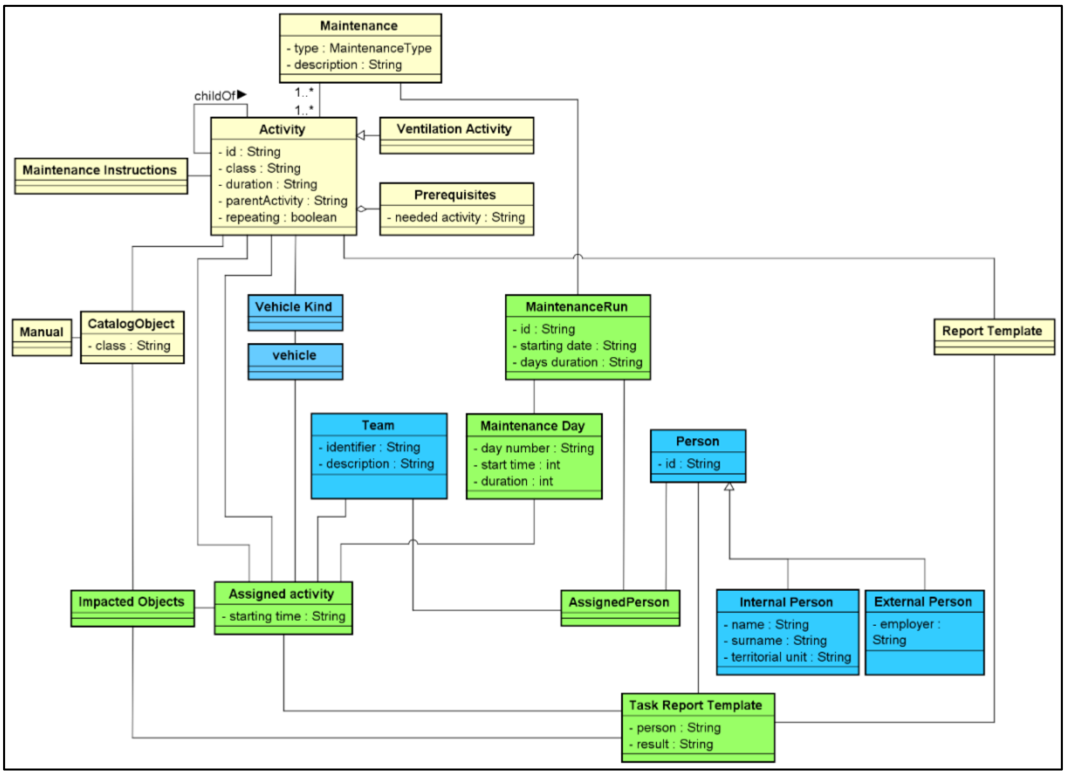

Figure 33: UML: Maintenance Class Diagram

Following a thorough and detailed analysis of the various dynamic scenarios identified within the documentation data objects, a comprehensive UML diagram was developed. This diagram was designed to encapsulate the overarching concept of maintenance, along with the diverse range of activities associated with it. The analysis involved meticulously extracting relevant information from the documentation, identifying key maintenance tasks, and understanding the relationships and dependencies between these activities.

The UML diagram not only represents maintenance as a central concept but also breaks it down into its constituent parts, capturing the sequential flow of tasks, potential decision points, and any interdependencies that exist. By doing so, the diagram provides a holistic view of the maintenance process, offering clarity and structure to an otherwise complex set of activities. The various dynamic elements, such as conditions, triggers, and the roles involved, are also incorporated, ensuring that the model reflects the real-world nuances of maintenance operations.

The outcome of this effort is illustrated in the figure above, showcasing how UML can effectively organize and visualize complex processes in a way that is both accessible and informative. This model serves as a foundation for further analysis and refinement, supporting the goal of creating a robust framework for understanding and executing maintenance procedures. An accurate analysis of the various dynamic scenarios that we found in the documentation data objects, a UML diagram was designed that models the concept of maintenance in general, together with the various activities that go with it.

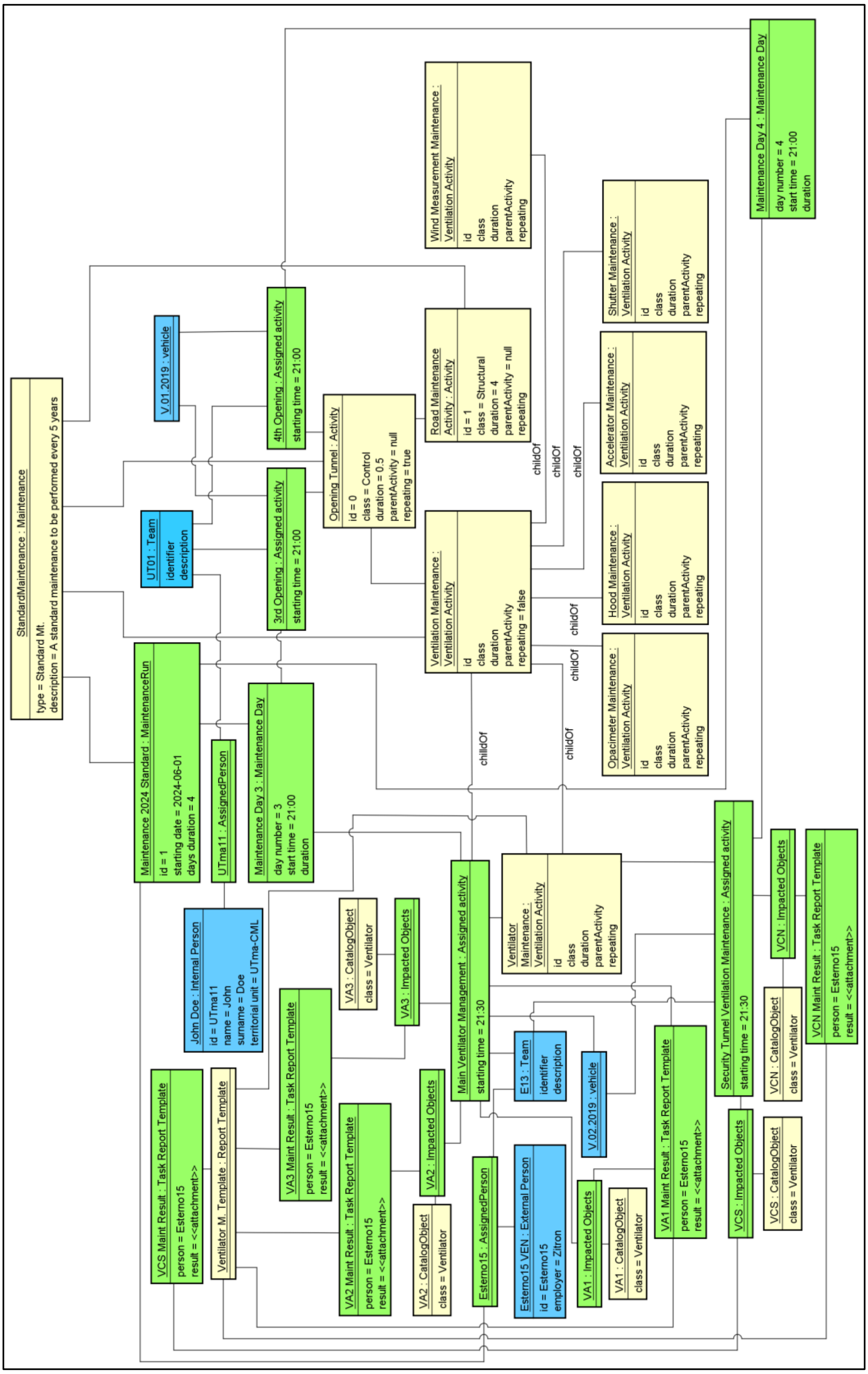

Figure 34: UML: Maintenance Instantiation Example

The above figure shows then an instantiation of the previous UML class diagram for a concrete maintenance procedure. Although still at a rather coarse-grained level, what becomes evident is that UML is indeed capable of correctly capturing the facts on maintenance procedures.

The flexibility and adaptability of UML for modelling dynamic processes are particularly evident when analysing the examples provided in the following figures. These examples demonstrate how UML can be applied to a series of concrete use cases, enabling the creation of clear, structured representations of complex workflows. This includes the ability to effectively model parallel processes, which are essential in many modern systems, as well as processes where individual steps are further enriched with additional meta-information. By leveraging these capabilities, UML proves to be a versatile tool for capturing and communicating the intricacies of dynamic processes in a visual and accessible manner.

However, there is one notable area where UML demonstrates certain limitations: its ability to encode constraints, such as time-based constraints, in a straightforward manner. While UML offers robust general-purpose modelling features, it struggles to accommodate scenarios requiring precise constraint representation without extensive manual effort or additional layers of complexity. In such cases, a specialized tool like BPMN (Business Process Model and Notation) might provide a more suitable alternative. BPMN is well-suited for handling constraint-driven information and offers a framework for representing intricate dependencies, including temporal constraints, with greater precision. However, this advantage comes at the cost of increased complexity in both the design and interpretation of the resulting models. Consequently, the choice between UML and BPMN often depends on the specific needs of the project, balancing flexibility and simplicity against the requirement for detailed constraint representation.

Example: ventilation maintenance

To delve deeper into the UML for maintenance management, it is possible to analyse the following example of ventilator maintenance, split into multiple activity diagrams highlighting its phases and sub-phases. Below, each image is accompanied by a caption that explains the key aspects of the maintenance flow.

Ventilation Maintenance Overview

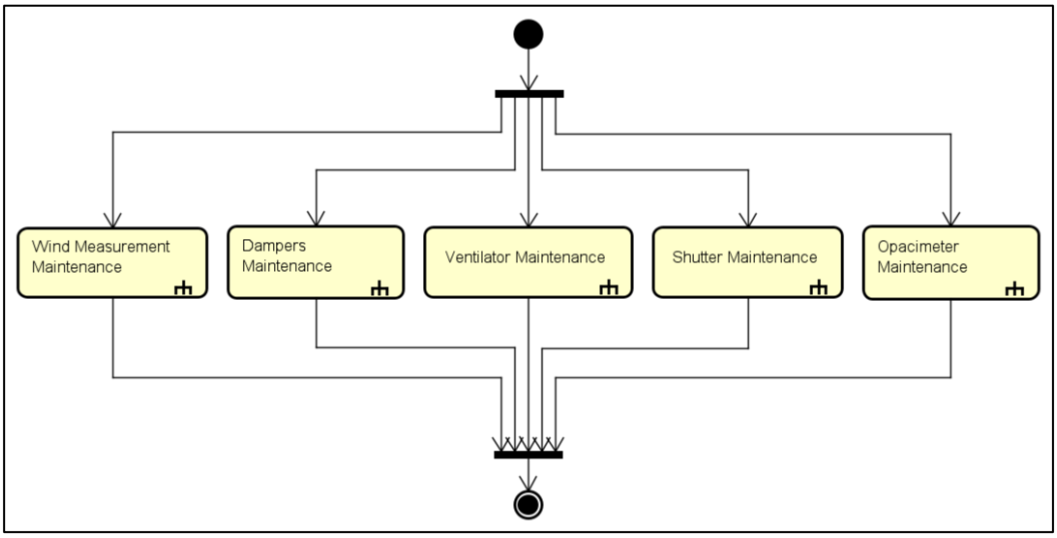

Figure 35: UML: Ventilation High Level Sequence Diagram

This high-level activity diagram illustrates the parallel tasks involved in the ventilation system's maintenance process—encompassing Wind Measurement, Dampers, Ventilator, Shutter, and Opacimeter Maintenance. It is possible to trace the logical flow for each individual operation, ensuring that all components are serviced appropriately. The following sections will focus in particular on the Ventilator Maintenance portion, examining its workflow step by step.

Ventilator Maintenance

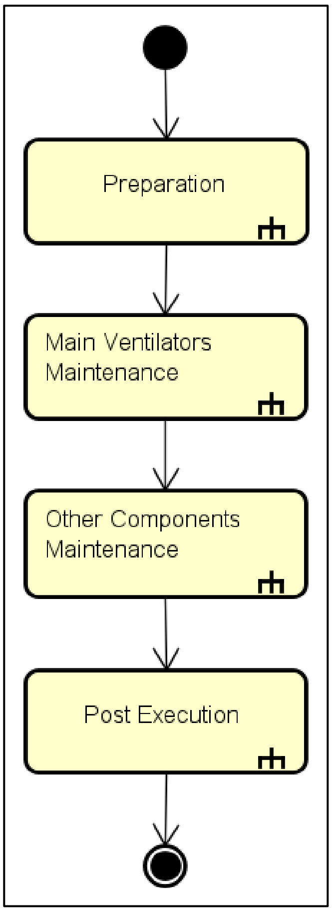

Figure 36: UML: Ventilator Maintenance

This activity diagram provides an overview of the ventilator maintenance process, organized into four main phases:

- Preparation
- Main Ventilators Maintenance
- Other Components Maintenance
- Post Execution

By structuring the workflow in a clear, sequential manner, this diagram highlights the logical flow from setup to final validation, underscoring the importance of each phase in maintaining overall system reliability.

Preparation

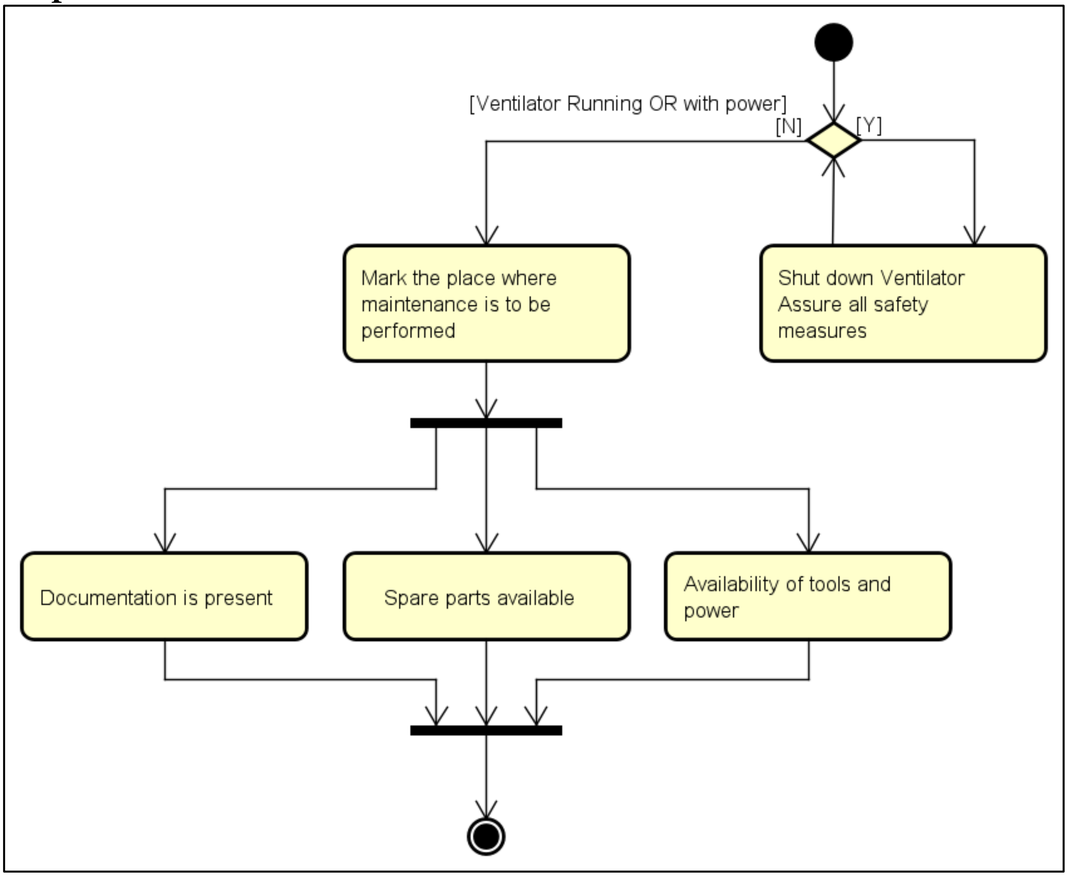

Figure 37: UML: Preparation

This activity diagram offers a detailed view of the Preparation phase, which begins with verifying whether the ventilator is running or receiving power. If it is active, the process includes a step to shut it down and take all measures to ensure safe operation for activities on the ventilator; otherwise, maintenance can proceed directly. Subsequently, the location for the maintenance work is identified, and the following checks are conducted in parallel:

- Documentation is present – Ensuring all relevant manuals and guidelines are available.
- Spare parts available – Confirming the availability of required replacement items.
- Availability of tools and power – Verifying that the necessary tools and a suitable power source are on hand.

By using a decision node for the ventilator's operational status and a fork/join structure to manage multiple parallel checks, the diagram clearly portrays the logical progression leading up to the main maintenance activities.

UML could also allow classes to be set up that contain the necessary checklists for the different steps, but it is normally more advisable to link to these checklists using a unique identifier for the different documents and a mapping service to link the identifiers to the document location.

Main Ventilators Maintenance

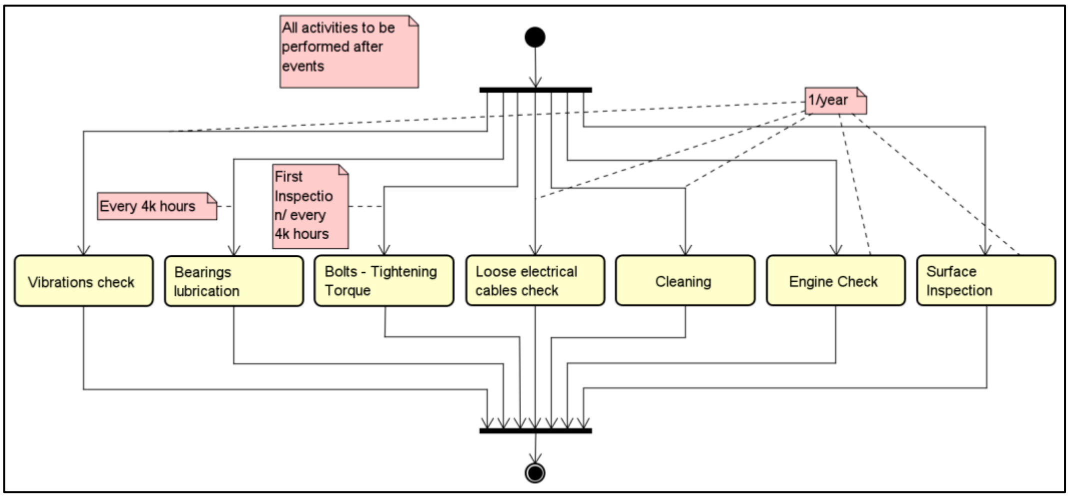

Figure 38: UML: Main Ventilators Maintenance

This activity diagram focuses on the specific tasks involved in maintaining the main ventilators, including both routine and event-driven procedures. After an initial fork, the following activities occur in parallel:

- Bearings lubrication – Recommended every 4,000 hours.
- Loose electrical cables check – Checking for unsecured or exposed cables.
- Bolts – Tightening Torque – Ensuring correct torque values.
- Cleaning – General cleaning of the ventilators.
- Engine Check – Inspecting mechanical components.

  - Surface Inspection – Verifying the integrity of external surfaces.

Notes and dashed lines indicate time-based triggers (e.g., every 4k hours, 1/year), as well as the possibility of performing these activities after specific events. All tasks converge at a join node, ensuring that each operation is completed before the maintenance process advances, thus maintaining consistent and comprehensive coverage of the ventilators' critical components.

Other Components Maintenance

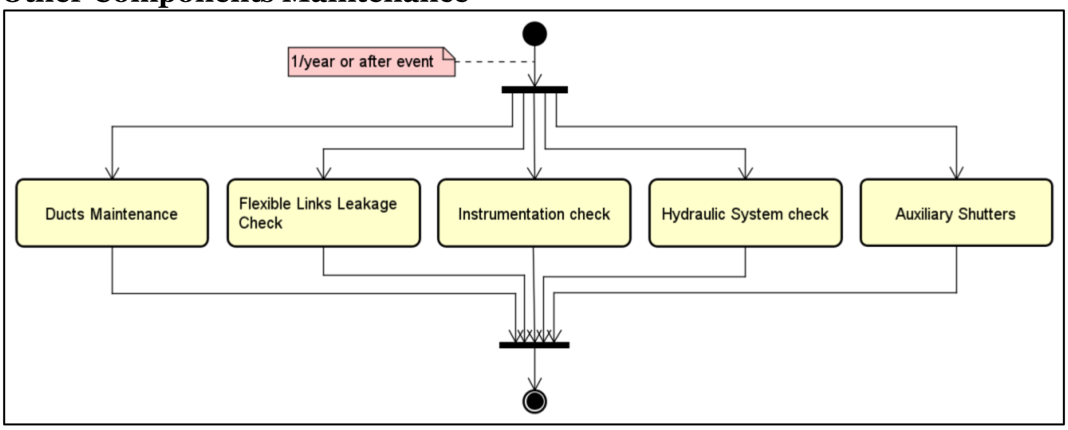

Figure 39: UML: Other Components Maintenance

This activity diagram outlines the maintenance workflow for secondary or complementary components of the ventilation system. After an initial fork, the following checks can be carried out in parallel, typically once a year or after specific events:

- Ducts Maintenance – Inspecting and cleaning internal duct elements.
- Artificial Ventilation check – Verifying the functionality of any auxiliary ventilation systems.
- Flexible Links Leakage Check – Ensuring that flexible joints and connections remain airtight.
- Instrumentation check – Examining sensors, gauges, and other monitoring devices.
- Hydraulic System check – Checking hydraulic lines, pumps, and related mechanisms, if present.

All parallel paths converge at a join node, indicating that each operation must be completed before the overall process concludes, thereby ensuring a thorough approach to maintaining all supporting elements of the ventilation system.

Post execution

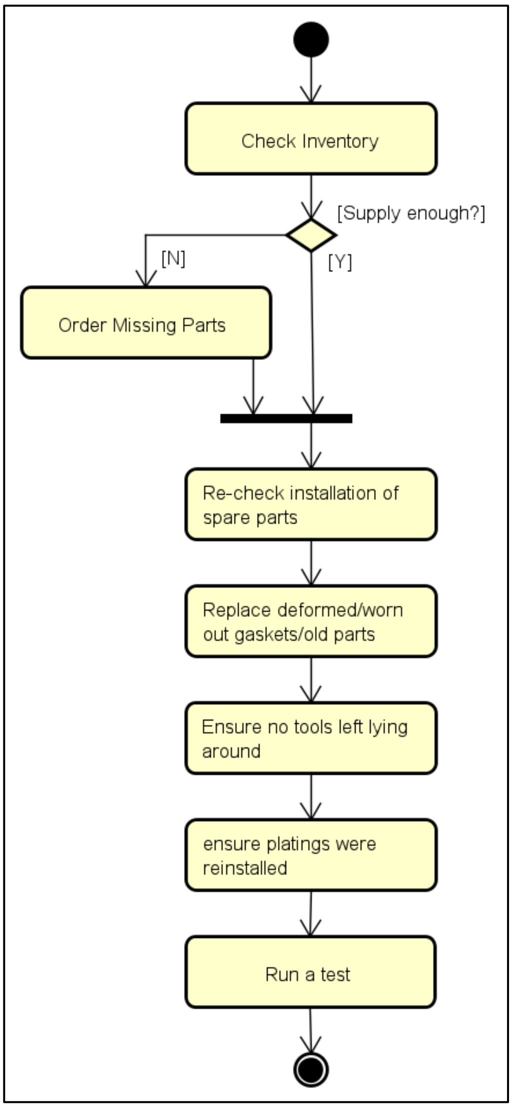

Figure 40: UML: Post execution

This Post Execution activity diagram illustrates the concluding phase of ventilator maintenance, ensuring that all remaining tasks are addressed before returning the system to operation. The sequence begins with an inventory check, followed by a decision node:

- Check inventory – If supplies are insufficient, proceed to Order Missing Parts; otherwise, continue.
- Re-check installation of spare parts – Verifying that newly installed items are correctly fitted.
- Replace deformed/worn-out gaskets – Ensuring any damaged components are switched out.
- Ensure no tools left lying around and ensure all parts were reinstalled – Removing foreign objects and confirming final reassembly.
- Run a test – Performing a final functional verification to validate that all maintenance steps have been successfully completed.

- By covering these final checks, the diagram underscores the importance of a structured closeout to maintain system reliability and safety.

The different steps can be also easily converted to *json* standard files and many other formats depending on the application accessing the data; an example is given in *Appendix A* of the present document.

5.3 From UML to a Relational Database

Rational databases are often used to implement maintenance tools, and the processes are coded inside the tables and relations (foreign keys) of the database. The modelling through UML of relational databases allows to profit from existing systems and to migrate an existing structure to different frameworks. In this chapter an UML modelling example of an existing database used for the Mappo-Morettina tunnel is performed. The existing database is based on Oracle and was used for maintenance planning and execution before the shift of the tunnel from the Canton to FEDRO.
UML class diagrams that describe maintenance processes can clearly identify the relevant entities (e.g., *Maintenance*, *Activity*, *Asset*) and how these entities relate to one another. By converting this conceptual model into a relational database schema, it becomes possible to manage all operational data in an organized, consistent manner. This approach supports:

- **Structured data storage**: maintenance tasks, schedules, and resources can be stored systematically.
- **Preservation of constraints**: UML relationships (one-to-many, many-to-many) become foreign keys or linking tables, enforcing referential integrity.
- **Ease of querying and reporting**: Standard SQL-based tools can be applied to analyse and monitor maintenance operations more effectively.

Reasons and Potential Benefits:
Converting the UML model to a **relational database** may be useful for:

- **Making the model operational**: Rather than remaining a mere diagram, it becomes a set of actual tables, complete with constraints that mirror the cardinalities and associations drawn in UML.
- **Preserving historical data**: Every intervention or single activity performed can be saved as a record, allowing for future analysis and the adoption of predictive approaches.
- **Integrating with other systems**: Orchestration tools, BPMN modules, or monitoring solutions can draw from the same database to operate coherently, avoiding duplicated data.

Main Steps for UML → DB Mapping:
Classes → Tables
Each UML class (e.g., *Maintenance*, *Activity*) can be turned into a table (e.g., `MAINTENANCE`, `ACTIVITY`). The attributes (for instance, `startDate` and `description`) become columns.

Relationships → Foreign Keys:

- **One-to-many (1–*) relationships** in UML: The table on the "many" side adds a foreign key referencing the PK of the "one" side.
- **Many-to-many (–)**: A "bridge" table is created with two foreign keys.
- **Inheritance**: If the UML model includes subclasses, it may be necessary to adopt a suitable strategy (e.g., table-per-subclass, table-per-hierarchy).

5.3.1 Example of Mapping from UML to a Relational Database

The following figures Figure 41 and Figure 42 shows how a UML diagram can be translated into a relational database schema in SQL (demonstrated here with OracleDB). On the left side is the UML view, comprising classes (for instance, man_activity and man_activitytype) along with their associations, while on the right side are the corresponding tables, inheriting the names and structures originally defined in UML.

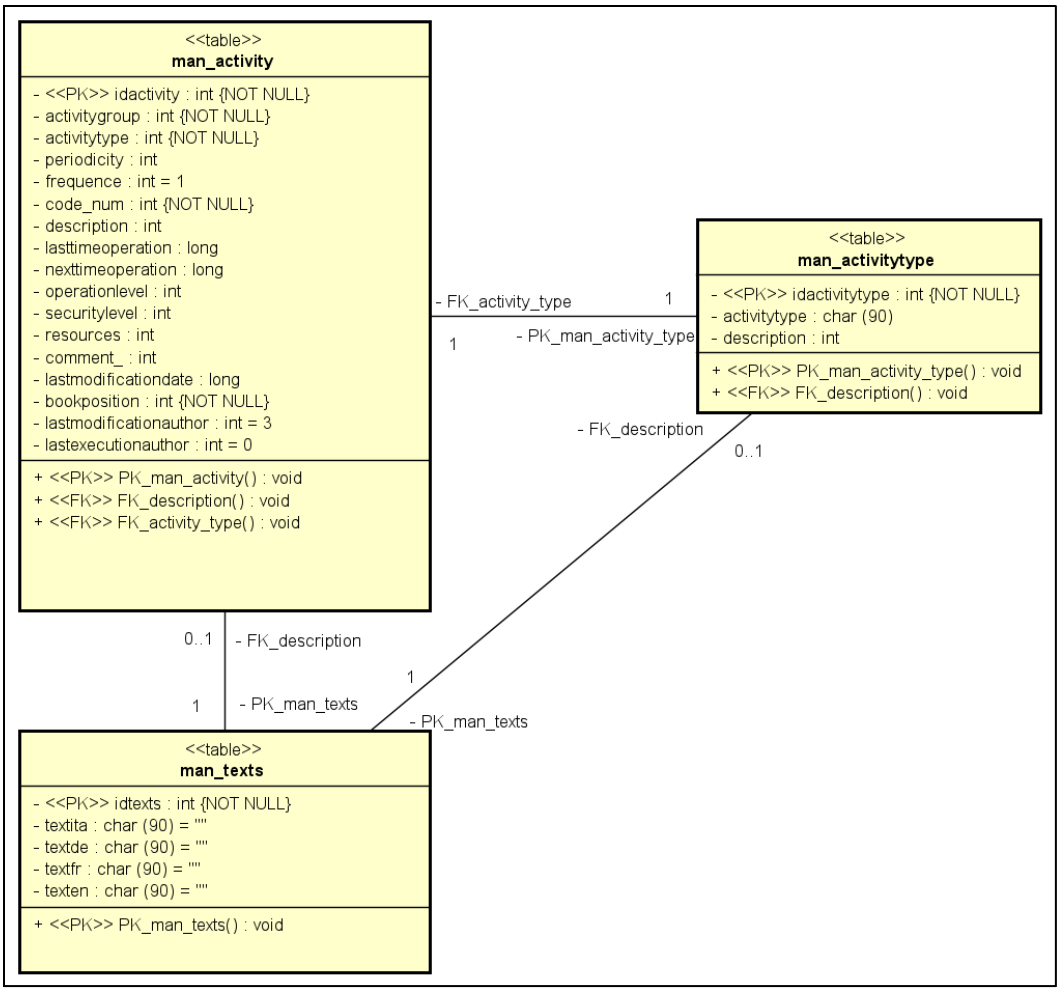

Figure 41: UML Diagram corresponding to relational DB

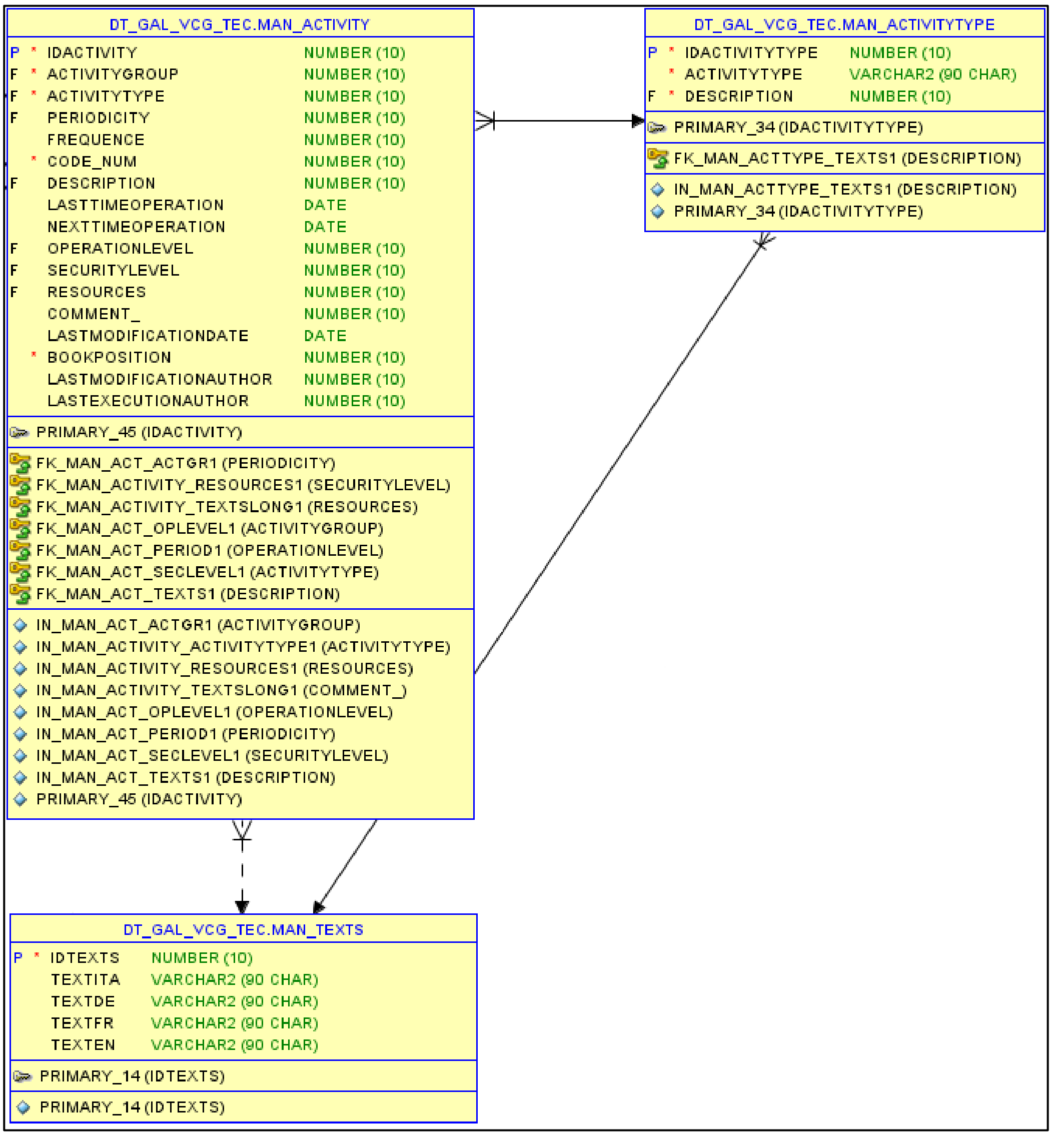

Figure 42: Oracle DB (SQL) diagram

5.3.2 Analysis of UML Components

In this example, one of the central classes is man_activity, which includes attributes such as idactivity, activitytype, and periodic. These attributes are translated directly into columns in the resulting database schema (for instance, activitytype, description, frequency), thereby maintaining the semantic alignment with the UML model. The original UML relationships (whether one-to-one, one-to-many, or many-to-many) are enforced through foreign key constraints or, in the case of many-to-many associations, through linking tables. This method upholds the cardinalities and constraints depicted in the class diagram and ensures that the underlying data remains consistent with the conceptual design.

5.3.3 Converting into Tables and Columns

Each UML class is represented as a table, and every attribute defined within that class becomes a corresponding column. While it may be necessary to adapt naming conventions, for instance, using uppercase identifiers or underscores, these adjustments do not alter the fundamental relationship between UML entities and their database counterparts. In this manner, a class in the model retains its identity in the physical schema, preserving the link between conceptual and implementation layers.

5.3.4 Setting Primary and Foreign Keys

The translation process also involves establishing primary keys for each table, mirroring the unique identifiers that appear in the UML specification. Relationships among tables are expressed through foreign keys, which reference the primary keys of related entities. By mapping UML associations in this way, the database design maintains the same referential integrity rules and relationship types (e.g., one-to-many) stated in the original diagram.

5.3.5 Benefits for Maintenance Management

Adopting an approach in which UML components are realized as relational entities offers clear advantages for maintenance operations. Entities such as *man_activity* or *man_texts* hold structured data related to scheduling, resources, and procedural details. Moreover, the interlinked nature of these tables facilitates efficient retrieval of textual documentation, technical parameters, and additional information needed to plan or monitor interventions.

5.3.6 Advantages of a Relational Database Derived from UML

Constructing a relational schema that closely follows the UML model ensures a high degree of alignment with the domain requirements. This alignment minimizes the likelihood of misinterpretation or omission during implementation. Furthermore, evolving the database schema to accommodate additional entities or attributes identified in subsequent iterations of the UML diagram becomes more straightforward, as the primary structure remains consistent with the conceptual model.

5.4 Swim-lane Diagrams

Swim-lane diagrams are very appropriate for a representation of the division of processes between different domains where the activities, sequences and interfaces of and between the different domains are identified. Swin-lane diagrams are part of the UML Activity diagrams and of the BPMN diagrams. Examples of swim-lanes are given in the following figures.

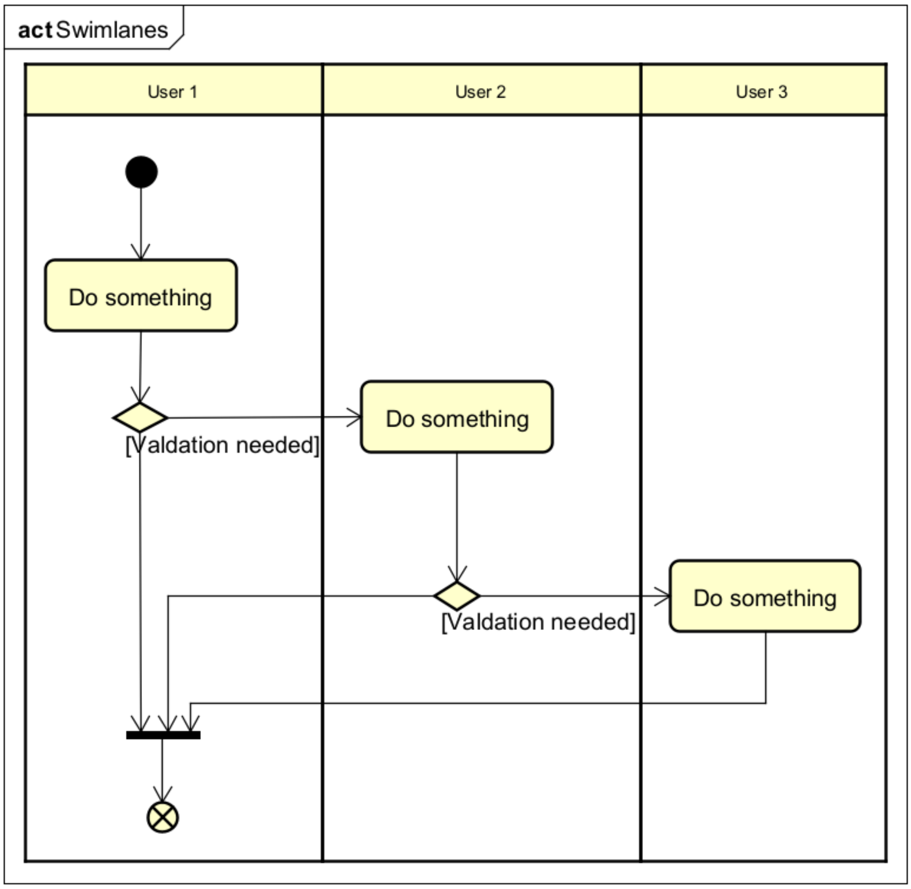

Figure 43: UML activity diagram with swim-lanes.

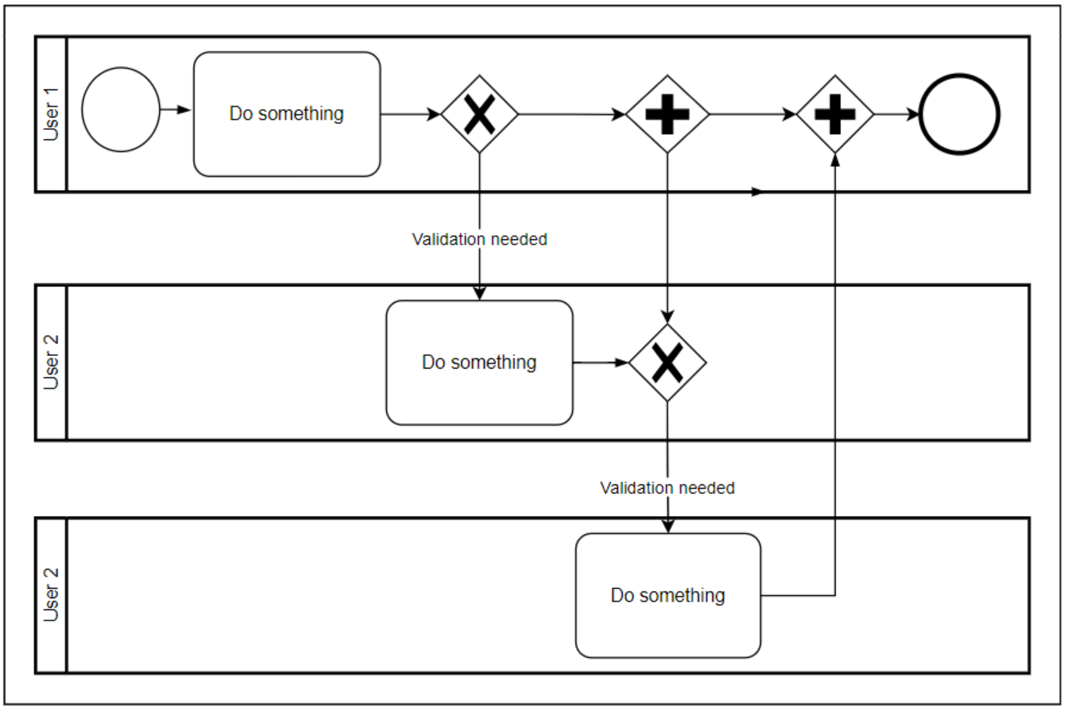

Figure 44: BPMN diagram with swim-lanes.

5.5 Conclusions of WP3

The primary objective of Work Package 3 (WP3) was to determine the feasibility and appropriateness of utilizing BPMN (Business Process Model and Notation) for modelling the dynamic aspects of road tunnels. These dynamic elements included specific use cases, such as maintenance activities and emergency response procedures, which are critical for the operational safety and efficiency of tunnel infrastructure. Given the complexity and high stakes of managing such scenarios, selecting a suitable modelling language was crucial for ensuring both clarity and precision in representation.

An initial exploration of BPMN revealed some promising features tailored to process modelling, particularly its ability to define workflows with time-based constraints and complex interdependencies. However, this deeper analysis quickly uncovered a significant drawback: BPMN introduces a considerable degree of additional complexity. While its framework excels in capturing intricate, constraint-heavy processes, the accompanying overhead in terms of model creation, interpretation, and maintenance presented challenges that outweighed its benefits in this context. This prompted a re-evaluation of alternative approaches that could balance expressive power with practical usability.

In response to these findings, the team turned its focus to UML (Unified Modelling Language) to investigate how effectively it could handle the representation of dynamic processes - UML proved not only capable but remarkably versatile. Its established framework for modelling behaviour through activity diagrams, state machines, and sequence diagrams allowed for the detailed encoding of dynamic processes. These capabilities were applied to capture maintenance workflows and emergency response procedures with clarity and efficiency, meeting the requirements of the use cases defined in WP3.

Furthermore, the use of UML provided an unexpected but valuable advantage: its seamless integration with the static models developed during Work Package 2 (WP2). This compatibility enabled the creation of a unified, cohesive framework that bridged both the static and dynamic elements of the tunnel system. By maintaining consistency across these interconnected aspects, UML facilitated a more comprehensive and manageable representation of the overall system.

Ultimately, the findings of WP3 strongly suggest that UML offers a robust and effective approach for modelling the dynamic components of road tunnels. While BPMN may be more appropriate for highly complex, constraint-driven processes in other domains, UML strikes an optimal balance for the requirements of this particular application. This conclusion not only underscores UML’s adaptability but also positions it as a forward-looking solution for addressing the dynamic modelling needs of modern infrastructure systems like road tunnels.

6 WP4: Project Outlook

6.1 General Observations

The TUML project aimed to explore whether the Unified Modelling Language (UML) could effectively serve as a visual modelling tool to represent road tunnels and their infrastructure from a static perspective. Additionally, the project sought to determine if these UML models could be enhanced with Business Process Model and Notation (BPMN) to capture the dynamic aspects of tunnel operations and processes.

6.1.1 Findings and Insights

The project concluded with a definitive **YES** regarding UML's ability to model the static structure of road tunnels. UML's diverse set of diagrams—such as class diagrams, component diagrams, and deployment diagrams—proves sufficiently robust to capture the intricate details of tunnels and their associated components, from overarching structures down to granular elements. BPMN contributes to the modelling capabilities where UML reaches some functionality limitations; however, these limitations only manifest themselves in particularly complex modelling cases.

6.1.2 Challenges in Creating Consistent, Usable Models

While UML can model static structures effectively, building comprehensive, consistent models for subsequent use - particularly for creating digital twins - remains challenging due to several upstream issues, many of which stem from historical and administrative factors:

1. Unambiguous Terminology:

A significant obstacle is the lack of unified, non-ambiguous terminology; the application of the AKS-CH terminology is mandatory for reaching a consistent result. Switzerland's multilingual context adds complexity.

2. Component Catalogs:

Even if linguistic inconsistencies were resolved, there remains a challenge in cataloging all potential tunnel components in a comprehensive and unambiguous manner:

- **Current Documentation:** FEDRO provides various documents related to tunnel components that are taken as reference for the correct naming of components and processes.
- **Terminology Standard:** The reference standard for uniform and language-independent coding is the FEDRO standard 13013 concerning the coding standard (AKS-CH) used on Swiss national roads. This coding standard has been adopted by the latest projects, but the previous ones used a regional coding system (for example in Ticino the CDR code) and the Mappo-Morettina is an example with documentation created with the old coding. The AKS-CH is very efficient for the coding of components, cables and data points, but it is still not always correctly referenced in the final documentation of the equipment.

3. Lack of Standardized Documentation Formats:
Another critical issue is the format and organization of tunnel-related documentation:

- **Documentation Disarray:** Ideally, each tunnel would have a well-organized, coherent set of documents, systematically structured and easily accessible. The documentation structure has evolved with time, and not all sets of documentation are aligned and reach the new requirements.
- **Current Reality:** In practice, documentation often exists as a collection of hundreds of files in varying formats. This makes automated processing more intensive and difficult.
- **Manual Labor Requirements:** Extracting the necessary information from these documents to build UML diagrams is labor-intensive, requiring significant manual effort to sift through and organize data.

4. Unclear Ownership of Documentation:
One critical issue that became apparent is the challenge of maintaining up-to-date documentation once a tunnel becomes operational. This issue is not merely administrative but has significant implications for the long-term usability and accuracy of the documentation. During our manual analysis of the extensive "documentation data set", it was evident that while updates to the documentation did occur (as indicated by file timestamps), adding complexity to a correct interpretation of the information.

The absence of a structured change protocol results in documentation that quickly becomes outdated or misaligned with the actual state of the tunnel. Over time, this disconnect can lead to a phenomenon known as **documentation quality decay**, where the accuracy and reliability of the documentation degrade progressively. As changes to the tunnel's infrastructure, equipment, or processes occur—whether due to maintenance, upgrades, or operational adjustments—the documentation should ideally reflect these modifications in a timely and consistent manner. However, without a clear system for tracking, approving, and implementing updates, inconsistencies and errors creep in.
This drift between the documentation and the real-world state of the tunnel poses several risks:

- **Operational Risks:** Outdated documentation can lead to incorrect assumptions during maintenance or emergency interventions, increasing the likelihood of errors or delays.
- **Regulatory and Compliance Issues:** Inadequate documentation may fail to meet legal or regulatory standards, exposing operators to potential penalties or legal liabilities.
- **Increased Maintenance Costs:** Engineers and technicians may need to invest additional time and resources to verify or update information manually, leading to inefficiencies and higher operational costs.

To address this issue, it is crucial to implement a robust **change management framework** for documentation. Such a framework should include:

1. Version Control Systems: Using tools that automatically track changes, maintain version histories, and allow for rollbacks if needed.

2. Clear Approval Workflows: Establishing a formal process for reviewing and approving changes, involving relevant stakeholders at each stage.
3. Regular Audits: Conducting periodic audits of the documentation to ensure it remains aligned with the physical state of the tunnel and its systems.
4. Training and Awareness: Ensuring that all personnel involved in documentation management understand the importance of accurate record-keeping and are trained in best practices.

In summary, maintaining up-to-date, accurate documentation is a continuous process that requires diligent oversight and structured protocols. Without these measures, the documentation's value diminishes over time, undermining its utility and increasing operational risks. By addressing the gaps in change management, organizations can ensure that their documentation remains a reliable and valuable resource throughout the tunnel's lifecycle.

6.1.3 Path Forward: Recommendations and Considerations

To overcome these challenges and move towards creating digital twins of tunnels, several actions are recommended:

- **Standardizing Terminology:** Create a standardized terminology based on the AKS-CH standard that can address all the modelling challenges.
- **Developing a Comprehensive Catalog:** Enhance the existing AKS-CH component catalog, ensuring it is exhaustive and language-agnostic.
- **Implementing Documentation Standards:** Creating a standardized format for tunnel documentation—ideally in a processable, structured format—would streamline the modeling process and reduce manual workload.

In conclusion, while UML is a powerful tool for modeling the static aspects of road tunnels, achieving the broader vision of consistent, digital-twin-ready models, requires addressing these foundational issues. Collaboration between stakeholders, standardization of terminology, and improvements in documentation practices will be essential steps in making this vision a reality.

The answer to the second question—whether BPMN (Business Process Model and Notation) can effectively model the dynamic aspects of a tunnel's operations, such as maintenance routines and emergency procedures—is **"Yes",** however with some limits. BPMN, while a powerful tool for capturing complex workflows and business processes, proved to be excessive and unnecessarily complicated for this specific use case. In fact, **UML,** which was originally intended to handle the static structures, turned out to be more than capable of representing the dynamic aspects as well.
When we attempted to model maintenance procedures using BPMN, the resulting diagrams became unwieldy and excessively intricate, far exceeding the complexity of the actual procedures they were meant to represent. This situation highlights a key issue in process modeling: the model itself should not become more complicated than the system it seeks to describe. Overly complex models not only defeat the purpose of simplification but also introduce new challenges in comprehension and maintenance.
BPMN seems to be indicated for describing sensible processes concerning e.g. the intervention procedures in case of an event; these kinds of procedures imply a strong coordination between entities and a time relevant execution path. We must consider that the intervention plans documentation contain clear flow charts implemented for

the same reason UML and BPMN diagrams were proposed for this project: clarity, synthesis, unambiguousness and digitisation.

6.1.4 Challenges in Modeling Dynamic Procedures

Beyond the extended complexity of BPMN, several additional issues emerged that complicate the use of UML for dynamic modeling in this context:

1. Informal Nature of Procedures

Maintenance and emergency procedures within tunnel operations are often defined in an informal, flexible manner. This flexibility is intentional and, in many cases, desirable. In dynamic and high-stakes environments like tunnel management, strict adherence to rigid protocols may hinder effective decision-making and quick responses:

- **Interpretation and Adaptation:** Operators and maintenance teams need the freedom to adapt procedures based on real-time conditions. As a result, procedures are often documented as guidelines rather than rigid step-by-step instructions, leaving room for interpretation.
- **Conflict with Formal Modeling:** UML, and particularly BPMN, rely on precise, well-defined processes. Attempting to formalize inherently flexible procedures into strict models can lead to overspecification, which reduces the very flexibility needed in critical situations.

2. Integration with Static Models

Even if UML were used to successfully model dynamic processes, another significant challenge remains: these dynamic models would need to be seamlessly integrated into pre-existing UML models of the tunnel's static structure. This integration is currently challenging due to several factors:

- **Incomplete Static Models:** As discussed earlier, the static models of tunnels are often incomplete or inconsistent due to documentation issues and terminology challenges. Without a solid foundation, adding dynamic models would only compound the problem.
- **Tool and Workflow Limitations:** Most modeling tools are not designed to handle the simultaneous management of both static and dynamic aspects in a cohesive way. The workflows required to maintain this integration can be complex and resources intensive.

3. Balance Between Flexibility and Formalization

A key tension exists between the need for **flexibility** in dynamic procedures and the desire for a **full-blown specification** in modeling. While formalized models can provide clarity and consistency, they can also limit the ability to adapt to unexpected situations. In the context of tunnels, where safety and efficiency are paramount, finding the right balance is crucial.

6.1.5 Conclusion and Recommendations

In summary, while UML is sufficient for modeling both static and dynamic aspects of tunnel operations, the practical challenges associated with formalizing dynamic procedures suggest that a more pragmatic approach is needed. Rather than striving for overly detailed models, focusing on **high-level frameworks** that capture key processes while allowing for flexibility may be a better solution. Additionally,

improving the underlying static models and addressing documentation issues should take precedence before attempting to model dynamic aspects in detail.
For future efforts, adopting a **hybrid approach** that combines the strengths of both formal modeling and flexible, guideline-based documentation could offer the best of both worlds. This would enable tunnels to be safely managed while maintaining the adaptability necessary for real-world operations.

6.2 Digital twins

6.2.1 Introduction

A Digital Twin is a comprehensive virtual replica of a physical object, process, or system. Unlike traditional three-dimensional or BIM-based models that emphasize geometry and structure, a true Digital Twin incorporates all relevant information about the real-world entity it represents, including data streams, operating rules, and interactions that may not be purely graphical. In the context of tunnel management, a Digital Twin unifies static representations, such as BIM models (used during the design phase) and UML diagrams (which map out tunnel assets and structures), with dynamic process descriptions in UML or BPMN, real-time sensor data, and higher-level analytics. This integrated virtual model is constantly fed by on-site measurements and operational information so that it mirrors the actual tunnel's current state.

6.2.2 How Digital Twins Work

A Digital Twin is created by collecting and integrating data from sensors, IoT devices, monitoring systems, and other sources. This data feeds into the virtual model, which faithfully replicates the physical object or system. The Digital Twin is continuously updated with new data, allowing it to accurately reflect the current state of the physical system.
The bidirectional connection between the Digital Twin and its physical counterpart enables real-time monitoring and the ability to predict future performance through simulations. For instance, a Digital Twin can simulate the operating conditions of industrial machinery, test hypothetical scenarios without disrupting real production, and identify issues before they occur, thereby improving efficiency and reducing maintenance costs.

General advantages of Digital Twins:

- **Operational Optimization:** Digital Twins allow for the simulation of scenarios and the optimization of processes without interrupting physical operations.
- **Predictive Maintenance:** By continuously monitoring the state of a system, Digital Twins can predict failures or maintenance needs, preventing costly issues.
- **Cost Reduction:** Through simulation and optimization, Digital Twins can help reduce operational costs and improve overall efficiency.
- **Product Development:** They enable the testing of new products and processes in a virtual environment before implementing changes in the real world.

6.2.3 Digital Twin in a digital tunnel

The adoption of a Digital Twin in a complex infrastructure such as a digital tunnel would represent an evolution in the field of infrastructure management and maintenance. A Digital Twin in a tunnel is an accurate virtual replica of the physical infrastructure that allows monitoring and simulation of operating conditions in real time, integrating data from sensors, IoT devices and monitoring systems.

Advantages of using Digital Twin in a digital tunnel:

- **Continuous monitoring and improved safety:** Thanks to its integration with IoT systems, Digital Twin enables continuous monitoring of the tunnel's internal conditions. This allows anomalies or potential hazards (such as poor air quality or equipment malfunctions) to be detected proactively, ensuring greater safety for tunnel users. We must point out that in any case, a tunnel guarantees safety functionality through its own control and supervision system; the adoption of a Digital Twin refers to a less essential level.
- **Predictive maintenance:** Continuous data analysis makes it possible to predict possible failures or malfunctions, minimising maintenance downtime. In this way, tunnel operators can plan maintenance work more efficiently and intervene only when necessary, avoiding costly unexpected repairs.
- **Resource optimisation:** By simulating tunnel operation under varying conditions, Digital Twin optimises the use of resources such as energy for ventilation and lighting, improving the energy efficiency of the infrastructure and reducing waste.
- **Testing of hypothetical scenarios:** A further advantage of the Digital Twin is the possibility of testing hypothetical scenarios without affecting actual tunnel operations. For example, operators can simulate traffic behaviour in the event of a partial or complete tunnel closure, thereby optimising traffic management plans to minimise disruption to users.

6.2.4 Infrastructure Management via Digital Twin

The integration of a Digital Twin into a digital tunnel not only offers immediate benefits in terms of monitoring and maintenance but also represents a strategic tool for long-term management of the infrastructure.
All this is made possible by the management and data collection system integrated in the Digital Twin. In fact, by having all the information concerning the progress and status of the various devices available in a DB, it is possible to keep track of the general situation and prevent different cases.
As indicated in the previous chapter (Advantages of using the Digital Twin in a digital tunnel), such a system allows for the testing of hypothetical scenarios, which means that, having the general state of the tunnel, it would be possible to simulate possible system failures or emergency scenarios, offering the possibility of testing response strategies to the various situations, without having to interrupt real operations.

6.2.5 The importance of predictive maintenance

As previously mentioned, by implementing a structure based on a Digital Twin within the tunnel, a highly efficient predictive maintenance system could be developed. By monitoring data trends in real-time and leveraging advanced analytical techniques, the system would be able to provide a comprehensive and dynamic view of the future state of the infrastructure and systems, such as the ventilation system or environmental monitoring sensors.
Thanks to the continuous processing of historical and real-time data, it would be possible to predict anomalies and potential failures before they occur. This would

enable timely and targeted interventions, thereby reducing downtime and maintenance costs. For instance, predictive analysis could identify a sensor that is progressively providing out-of-range values or detect a discrepancy between expected and measured data, signalling the need for maintenance or replacement in advance.

Moreover, the Digital Twin would allow the simulation of various failure scenarios, such as the breakdown of a critical component or a malfunction of the ventilation system, enabling the assessment of the impact on the tunnel's operation and proactive planning of corrective actions.

6.2.6 Minimal operation conditions

A Digital Twin can also contribute to the assessment of the compliance of the tunnel state to the minimal operational conditions, and for the different state scenarios it could automatically apply mitigation measures to ensure the required level of safety.

6.2.7 Possible framework based on Digital Twins

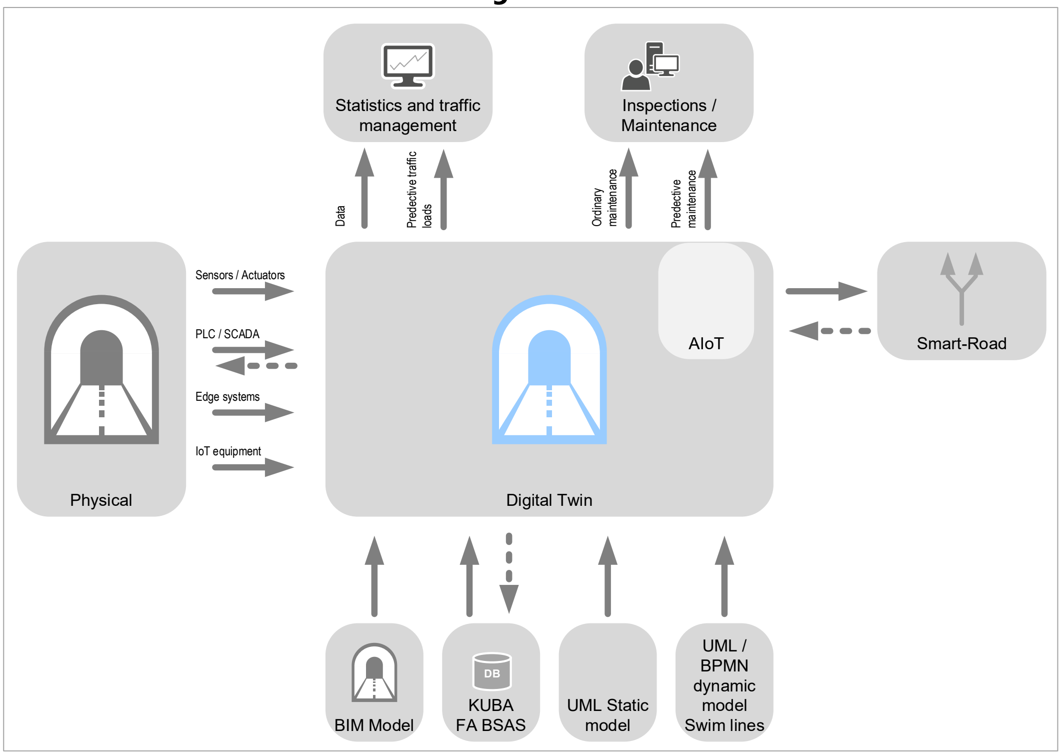

Figure 45: Possible framework based on Digital Twin

6.2.7.1 General Introduction

The Figure 45 represents a possible general structure for an advanced tunnel management system based on the integration of a Digital Twin. This system enables the creation of an accurate virtual replica of the physical infrastructure, modelled through tools such as BIM (Building Information Modelling), KUBA, BSAS, and UML static and dynamic models. The Digital Twin integrates static and real-time data to provide an interactive platform for analysis and management.
Once modelled, the system establishes bidirectional communication with the physical tunnel through a network of sensors, actuators, and edge computing platforms. This

configuration allows for continuous monitoring of operating conditions, planning of maintenance interventions, and optimization of operations. Additionally, the system allows for the simulation of complex scenarios, testing operational strategies in a virtual environment without interfering with the real infrastructure.

6.2.7.2 System Structure

Digital Twin and Modelling:
The Digital Twin forms the core of the system. With data provided by BIM models and databases such as KUBA and BSAS, the system acquires detailed information about the physical structure and equipment within the tunnel. These digital models enable a detailed representation of the infrastructure and its components, facilitating management and maintenance through their integration with real-time operational data.

By leveraging advanced modelling tools, the system represents both the structural characteristics of the tunnel and its dynamic interactions with the surrounding environment. The use of databases and digital models helps centralize and standardize information, making it accessible for in-depth analysis and predictive simulations.

Sensors, Middleware, and Automation Systems:
The collection and management of operational data rely on a network of sensors distributed throughout the tunnel, capable of monitoring critical parameters in real-time. These devices include IoT sensors, PLCs (Programmable Logic Controllers), SCADA (Supervisory Control and Data Acquisition) systems, and edge computing units, all working in synergy to ensure a continuous and reliable data flow.

- IoT and Environmental Sensors: Provide real-time data on various operational and structural tunnel parameters, contributing to a detailed view of operating conditions.
- PLCs and SCADA: Traditional automation systems that control and supervise critical infrastructure, such as ventilation, lighting, and safety systems. These can communicate directly with the middleware to transmit up-to-date operational data.
- Middleware: Acts as the central connection point, ensuring that data from IoT sensors, PLCs, and SCADA systems is processed coherently and sent to the Digital Twin for analysis and simulation.
- Edge Computing: Although specifically designed for some advanced industrial applications, edge computing systems provide a first level of local data processing, reducing latency and improving system responsiveness.

The adoption of this architecture allows for decentralization of part of the data processing, reducing the load on central systems and ensuring a faster response to constantly evolving operational conditions. Middleware plays a crucial role in orchestrating these data flows, ensuring interoperability among heterogeneous devices and standardizing the transmission of information to the Digital Twin.

6.2.7.3 System Functionalities

Monitoring and Maintenance:
One of the key aspects of the system is the continuous monitoring of tunnel operating conditions. Through the analysis of collected data, the Digital Twin enables the

identification of anomalies and the prediction of potential problems before they cause disruptions or malfunctions.
Thanks to predictive maintenance, targeted interventions can be planned, reducing costs and downtime. The analysis of historical and real-time data allows for the identification of trends that signal emerging issues, enabling the adoption of timely preventive measures.

AIoT Integration:
The use of artificial intelligence combined with the Internet of Things (AIoT) allows the system to dynamically adapt to operating conditions. AI analyses large volumes of data and identifies complex patterns, improving operational efficiency and reducing response times to critical events.

This technology also enables the automation of many management operations, reducing the need for manual interventions and optimizing resource usage. The integration of AIoT makes the system smarter and more proactive, enhancing overall tunnel management.

6.2.8 Monitoring, Predictive Maintenance, and Advanced Simulations

6.2.8.1 Monitoring and Advanced Data Analysis

Advanced monitoring is made possible through the integration of the Digital Twin, which enables the collection and analysis of data from IoT devices, environmental sensors, and control systems. These data are centralized in a structured database to ensure a comprehensive and up-to-date view of the infrastructure, facilitating the timely detection of anomalies and critical issues.
Through data analysis, the system allows precise monitoring of various tunnel components, such as ventilation, lighting, and safety systems. This approach enables operators to adopt more effective maintenance strategies, reducing operational risks and optimizing available resources.
Another key aspect is the ability to simulate hypothetical scenarios, such as critical failures or extraordinary events. For example, the consequences of a power outage or a road accident inside the tunnel can be assessed. These simulations help develop quick and efficient response strategies, ensuring greater safety for users and operators.

6.2.8.2 Predictive Maintenance

The implementation of the Digital Twin enables a predictive maintenance system capable of anticipating failures and malfunctions through advanced analysis of historical and real-time data. This approach minimizes downtime and reduces maintenance costs while improving infrastructure reliability.

6.2.8.3 Benefits of Predictive Maintenance

Thanks to continuous monitoring and predictive analysis capabilities, it is possible to:
- Predict anomalies and failures before they occur, preventing unexpected disruptions.
- Perform targeted and timely interventions, ensuring operational continuity.
- Analyse correlations between events and operating conditions, improving resource management.
- Develop customized predictive models, adapting maintenance to the tunnel's specific conditions.

For example, the system can detect abnormal variations in temperature or pressure sensor data, signalling the need for a technical check in advance. Additionally, by analysing past performance, it is possible to plan interventions more effectively, optimizing resource utilization and minimizing waste.

6.2.8.4 Advanced Simulations and Hypothetical Scenarios

One of the most innovative aspects of the Digital Twin is its ability to perform advanced simulations to test infrastructure response under various operating conditions. These simulations help identify system vulnerabilities and optimize intervention strategies in emergencies.

6.2.8.5 Types of Simulations

Among the main simulations that the system can execute are:

- Critical failure of essential components, such as ventilation fans or signalling systems.
- Ventilation system blockage, analysing the impact on air quality.
- Road accidents inside the tunnel, to evaluate evacuation procedures.
- Environmental emergencies, such as fires or smoke generation, to optimize operational responses.
- Emergency team response times, to minimize delays and improve safety.

6.2.8.6 Practical Applications of Simulations

The use of simulations allows for:

- Enhancing operational preparedness, by training operators with realistic scenarios.
- Optimizing emergency management, reducing reaction times.
- Refining intervention procedures, minimizing safety risks.

Furthermore, the ability to continuously update simulations based on real-time data enables rapid adaptation of operational strategies to new requirements. The Digital Twin thus proves to be an essential tool for the continuous improvement of infrastructure safety and reliability.

6.2.9 BSAS as a Support for the Digital Twin

Integrating digital models and operational data could help improve the management of road infrastructure, particularly through the use of the Digital Twin. A digital model can provide a more complete view of infrastructure status by combining structured historical data with real-time operational information. Ensuring the consistency and accessibility of these data sources is essential for effective decision-making in maintenance and infrastructure planning.

BSAS is the nationwide application used for the inventory and condition assessment of infrastructure equipment. Originally designed to standardize data collection and ensure uniformity in infrastructure monitoring, it serves as a centralized reference system for managing asset location, maintenance history, and technical specifications. Currently, BSAS is a temporary solution, in use until a final system is implemented.

Linking BSAS to the Digital Twin could help improve data integration and accessibility, reducing inconsistencies between existing documentation and real-world infrastructure conditions. A direct connection between these systems could simplify technical data retrieval and facilitate a more structured approach to infrastructure management.

In the context of maintenance, BSAS is primarily used to plan periodic maintenance activities, while real-time monitoring systems track ongoing infrastructure performance. Integrating these two sources of information could provide a more comprehensive view of asset conditions, making it easier to detect anomalies, identify long-term trends, and optimize intervention planning. This would allow for more targeted maintenance efforts, reducing unexpected failures and improving overall infrastructure reliability.
Beyond maintenance, BSAS could also contribute to enhancing infrastructure visualization within the Digital Twin, particularly through its structured geographic data. When combined with GIS systems and 3D modelling tools, this information could help create a more detailed representation of infrastructure networks, supporting better spatial analysis and operational planning.
While BSAS is already a well-established tool for infrastructure management, integrating it with the Digital Twin could enhance data consistency, prevent redundancy, and improve coordination between different management systems. Future developments could explore ways to make BSAS more adaptable to new digital technologies, facilitating a smoother transition towards a fully integrated infrastructure management approach.

6.2.10 Edge Computing and Fog Computing

After analysing the concept of the Digital Twin, it is appropriate to consider technologies that enable advanced data and information management, optimizing the integration between the physical and digital worlds. In this context, Edge Computing and Fog Computing represent relevant solutions to enhance the efficiency of IoT systems and ensure distributed data processing directly within the infrastructure.
Edge Computing involves shifting data processing closer to IoT devices, reducing latency and improving system responsiveness. This approach minimizes data traffic and supports real-time applications for predictive maintenance and continuous infrastructure monitoring. A practical example is **Siemens Edge**, a platform integrating edge devices and dedicated applications to optimize data management and processing directly in the field. This technology enhances system performance, ensuring faster responses and greater control over processes.
Fog Computing extends Edge Computing by distributing data processing across multiple layers between devices and network infrastructure. This model allows for a more scalable resource management approach, improving system resilience and ensuring operational continuity even in cases of connectivity disruptions. Furthermore, it facilitates interoperability between different platforms and IoT devices, making the management of large data volumes more efficient in complex environments.
The combined adoption of Edge and Fog Computing, in synergy with Digital Twins, enables the development of more efficient, resilient, and scalable IoT architectures, unlocking new intelligent management scenarios for infrastructures. This approach improves operational efficiency, reduces downtime, and optimizes maintenance, bringing significant advantages in managing complex environments such as a digital tunnel.

6.2.11 Artificial Intelligence of Things (AIoT)

AIoT represents the convergence of the Internet of Things (IoT) and Artificial Intelligence (AI), enabling devices to operate autonomously and improve their decision-making capabilities based on data. By integrating AI with IoT, connected systems within the digital tunnel can analyse vast amounts of information in real time, recognize patterns, and optimize responses dynamically.
This combination enhances predictive capabilities, allowing for proactive interventions in maintenance, safety, and operational efficiency. AI-driven models can assess sensor data, detect anomalies, and suggest corrective actions without human intervention. The continuous learning mechanisms inherent to AI ensure that these systems evolve over time, adapting to changing conditions and improving their performance.
Considering the growing complexity of digital twin environments, AIoT plays a key role in refining simulations and real-time monitoring. By leveraging AI, digital twins can provide deeper insights, automate processes, and facilitate decision-making in a more intelligent and responsive manner.

6.2.12 6G & IoT Quantum Networks

The evolution from 5G to 6G introduces significant advancements in IoT connectivity, ensuring ultra-low latencies, higher reliability, and seamless communication between devices in digital twin environments. These improvements enable real-time interactions with minimal delay, making complex simulations and remote operations more efficient and responsive.
Quantum Networks further enhance security in IoT ecosystems by leveraging quantum encryption techniques. Unlike traditional encryption methods, quantum-based security mechanisms offer absolute protection against cyber threats, ensuring that sensitive data remains secure and unalterable. This innovation is particularly relevant for digital twins, where secure and reliable data exchange is critical for accurate modelling and decision-making.
With 6G and Quantum Networks, digital twin systems will not only operate with unprecedented speed and precision but also guarantee enhanced security and resilience. These advancements will support the next generation of interconnected infrastructures, reinforcing trust and efficiency in digital transformation initiatives

6.3 Possible research developments

The TUML project has revealed an extraordinary potential for numerous future developments and groundbreaking initiatives. By establishing a foundational framework, it has opened up avenues for a wide range of possibilities and innovations. If the ultimate objective of the confederation is to create a comprehensive, holistic, and highly detailed understanding of all road tunnels throughout Switzerland—essentially achieving the ambitious task of developing digital twins for each individual road tunnel—then the scope of potential undertakings is vast and multifaceted. This goal could serve as a transformative step in enhancing infrastructure management, enabling advanced predictive maintenance, and fostering greater operational efficiency. Furthermore, the paths that could be explored are varied, promising, and abundant, paving the way for cutting-edge solutions in both technology and tunnel management.

However, many of those developments are bound to much needed upstream work by the various parties involved.

On one hand, there is the critical challenge of establishing a unique, unambiguous, and multilingual nomenclature for all the constituent entities and components that comprise road tunnels. Despite some preliminary efforts to address this issue, the current state remains suboptimal and lacks the consistency required for seamless communication and integration across various systems and stakeholders. To address this gap, we propose extending the AKS-CH standardized nomenclature framework to support four languages: the three primary national languages—German, French, and Italian—and English as a fourth, universal language. Incorporating English as a unifying linguistic element would not only facilitate broader accessibility but also significantly streamline the process of modelling and documentation. This approach would enhance clarity, reduce potential misunderstandings, and ensure the interoperability of data and systems, laying the groundwork for more efficient and effective management of road tunnel infrastructure.

On the other hand, a significant challenge lies in transforming the existing road tunnel models into a unified, coherent system by developing the necessary meta-models. This endeavour would require substantial effort and meticulous coordination among all stakeholders involved across the three distinct language regions of Switzerland. The expertise and domain-specific knowledge held by these regional contributors are indispensable to achieving this goal. Successfully integrating their insights into a comprehensive framework would not only enhance the consistency and usability of the models but also ensure that the final meta-models reflect the nuances and specificities of each region. Such an undertaking demands a collaborative approach, leveraging cross-disciplinary expertise and fostering open communication to bridge gaps in understanding and harmonize the diverse perspectives that are critical to the project's success.

Once such models, structured in alignment with a corresponding and well-defined meta-model, are successfully established, the logical progression toward achieving fully functional digital twins would involve utilizing these models to develop increasingly sophisticated, web-based interactive dashboards. These dashboards would serve as the primary interface for interacting with the models, offering a user-friendly and centralized platform for visualization and management.
In their initial stages, these dashboards would primarily represent the static structure of the models, providing a comprehensive yet foundational overview of the underlying data. However, for these tools to reach their full potential, consistent and systematic usage would be essential to incrementally enrich their features and capabilities. This enrichment process would enable the integration of dynamic elements, ultimately paving the way for the next critical phase: leveraging the existing tunnel IoT infrastructure to generate real-time information. This live data would then be seamlessly integrated into the dashboards, transforming them into powerful, dynamic tools capable of providing actionable insights and supporting advanced decision-making processes. Such a progression would mark a significant milestone in the journey toward the realization of true digital twins for road tunnel systems.

The final step in this transformative process would be the systematic encoding of all dynamic processes associated with road tunnels. This would begin with the relatively straightforward documentation and modelling of routine maintenance procedures, serving as a foundational layer. From there, efforts would progress toward capturing and formalizing more complex and critical processes, such as emergency response protocols.
This advanced stage of development would necessitate active collaboration with the various stakeholders responsible for managing emergency situations, including first responders, emergency services, and other relevant authorities. Their specialized knowledge and operational experience are essential to ensure that the encoded processes accurately reflect real-world requirements and challenges. By integrating these dynamic processes into the digital framework, the system would achieve a new level of operational comprehensiveness, enabling more efficient and coordinated responses to both routine and extraordinary scenarios. This culmination of efforts would significantly enhance the safety, reliability, and functionality of road tunnel management systems, representing a pivotal step toward achieving a fully integrated and intelligent infrastructure.

We fully recognize that the steps outlined above represent an exceptionally ambitious, multi-phase initiative that would extend over several years and require the active involvement of a wide range of stakeholders. The complexity and scale of such a project demand not only careful planning and execution but also a high degree of collaboration and coordination across diverse domains and regions.
To ensure the success of such an undertaking, having dedicated and knowledgeable academic partners to serve as companions throughout the project would be of paramount importance. An institution like the Software Institute, as exemplified in the TUML project, would play a critical role in providing the necessary research expertise, methodological rigor, and technological innovation to guide the project's development. Such a partnership would act as a cornerstone, ensuring that the project adheres to the highest standards of academic and practical excellence while also fostering the exchange of knowledge between academia and industry. This collaborative approach would be instrumental in overcoming challenges, advancing progress, and ultimately achieving the ambitious goals set forth for this transformative initiative.

6.4 Conclusions of WP4, closing words

The vision presented in the preceding section emphasizes the significant potential advantages of adopting a comprehensive, digital approach to encoding road tunnels across Switzerland. However, as highlighted, this initiative is confronted with a set of challenges that are both complex and multifaceted, divided into two broad categories. The first category encompasses issues that are primarily technical in nature, or at most, managerial in scope. The second category, on the other hand, is distinctly political, involving policy considerations, regulatory frameworks, and broader societal implications.

To draw a parallel from the healthcare sector, consider the implementation of fully integrated digital patient records. The process of establishing such systems is not purely a technical challenge; it involves extensive political and legislative processes that must address privacy, security, and data governance concerns. In a similar manner, the successful realization of the vision outlined for the TUML project, which aims to digitize and encode road tunnels, must navigate a comparable set of political hurdles. These challenges are not merely technical in nature but require careful consideration of policy, regulation, and stakeholder interests that extend far beyond the immediate scope of the project. Consequently, the path forward for the TUML project, much like the transition to digital health records, will need to carefully balance both technical innovation and political engagement in order to overcome the obstacles that lie ahead.

The digitisation of tunnels and the entire road network is a necessary step towards a large-scale autonomous driving concept. This project focused on the evolution of an existing paradigm but opens the door in the direction of a mobility revolution. In the future, mobility will be integrated into a concept that will not only encompass transport but also important aspects of optimised energy management, autonomous and non-personal vehicles, optimised travel and more sustainable and efficient management of communication routes.

Today, 100% digitisation seems to be inevitable, the challenge is for it to happen in a coordinated and conscious manner, and to avoid digitisation imposing itself in an uncoordinated and inefficient manner. UML and other standard modelling methods have precisely the objective of allowing us to define how things are to be done, in what way and with what interfaces, finally allowing us to implement models that correspond to what is planned.

7 Appendix A: Procedure – Example of json Generated Output


```
[
 {
  "name": "Maintenance 2024 Standard",
  "type": "Standard Mt.",
  "description": "A standard maintenance to be performed every 5 years",
  "startingDate": "2024-06-01",
  "duration": 2,
  "maintenanceDays": [
   {
    "dayNumber": 3,
    "scheduledDay": "2024-06-03",
    "activities": [
     {
      "startingTime": "21:00",
      "activity": {
       "name": "Opening Tunnel",
       "parentName": "null",
       "repeating": true,
       "duration": 0.5
      },
      "team": {
       "name": "UT01",
       "description": "",
       "members": {
        "UTma11": {
         "name": "John Doe",
         "surname": "Doe",
         "territorialUnit": "UTma-CML"
        }
       }
      },
      "vehicle": {
       "plate": "V.01.2019"
      },
```

```
 "impactedOBjects": []
},

 {
 "startingTime": "21:30",
 "activity": {
  "name": "Ventilation Maintenance",
  "parentName": "",
  "repeating": false,
  "duration": 0.0,
  "steps": {
   "Start": {
    "In parallel": [
     {
      "Shutter Maintenance": {
       "In parallel": [
        {
         "In-loco check": {
          "Visual Check": [
           "Surface Check",
           "Foreign Body check",
           "Slat Bearings and Drive Linkage Check",
           "Fasteners Check",
           "Damaged Coverings repairing"
          ],
          "[After Event]": {
           "[Y]": [
            "Signal Check",
            "Functional Check"
           ],
           "[N]": "Go to 'Functional Check'"
          },
          "Call Supplier"
         }
        },
        {
         "Shutter O/C Check": {
          "[Anomaly detected?]": {
           "[N]": "No anomaly, proceed",
           "[Y]": "Go to 'In-loco check'"
          }
         }
        }
       ],
       "Then": "Go to 'omitted'"
      }
     },
```

```
{
 "Ventilator Maintenance": {
  "Preparation": [
   {
    "[Ventilator Running OR with power]?": {
     "[Y]": [
      "Shut down Ventilator",
      "Go to 'Ventilator Running OR with power'"
     ],
     "[N]": [
      "Mark the place where maintenance is to be performed",
      {
       "In parallel": [
        "Documentation is present",
        "Spare parts available",
        "Availability of tools and power"
       ]
      }
     ]
    }
   }
  ],
  "Main Ventilators Maintenance": {
   "In parallel": [
    "Bolts - Tightening Torque",
    "Bearings lubrication",
    "Surface Inspection",
    "Loose electrical cables check",
    "Engine Check",
    "Cleaning",
   ]
  },
  "Other Components Maintenance": {
   "In parallel": [
    "Channel Components Maintenance",
    "Flexible Links Leakage Check",
    "Shutter?",
    "Hydraulic System check",
    "Instrumentation check",
    "Artificial Ventilation check"
   ]
  },
  "Post Execution": [
   "Check Inventory",
   {
    "[Supply enough?]": {
     "[N]": "Order Missing Parts",
```

```
          "[Y]": "Ok"
        }
      },
      "Re-check installation of spare parts",
      "Replace deformed/worn out gaskets/old parts",
      "Ensure no tools left lying around",
      "Ensure platings were reinstalled",
      "Run a test"
    ]
  }
},
{
  "Hood Maintenance": {
    "In parallel": [
      {
        "Hood Check": [
          "General Cleanliness Check",
          "Loose or worn parts check",
          "Slats closing check",
          "Fixing Tensioning check"
        ]
      },
      {
        "Engine Check": [
          "Engine Check (corrosion, dust, fixing)",
          "Electric connection, cable fixing and cable glands check"
        ]
      }
    ]
  }
},
{
  "Opacimeter Maintenance": [
    {
      "In parallel": [
        "Visual check on the control unit",
        "Electrical control installed in tunnel",
        "Mechanical control of the fixing",
        "Air filters replacement"
      ]
    },
    "Calibration/parameter check"
  ]
},
{
  "Wind Measurement Maintenance": [
    "Set FLOWSIC to Maintenance Mode",
```

```
                "External Component Cleaning",
                "Set FLOWSIC to Measure Mode"
              ]
            }
          ]
        }
      }
    },
    "team": {
      "name": "E13",
      "description": "",
      "members": {
        "Esterno15": {
          "name": "Esterno15 VEN",
          "employer": "Zitron"
        }
      }
    },
    "vehicle": {
      "plate": "V.02.2019"
    },
    "impactedOBjects": [
      {
        "objectName": "VA2",
        "objectClass": "Ventilator",
        "manual": "link-to-manual.pdf",
        "report": "link-to-report.pdf"
      },
      {
        "objectName": "VA1",
        "objectClass": "Ventilator",
        "manual": "link-to-manual.pdf",
        "report": "link-to-report.pdf"
      },
      {
        "objectName": "VA3",
        "objectClass": "Ventilator",
        "manual": "link-to-manual.pdf",
        "report": "link-to-report.pdf"
      }
    ]
  }
]
},
{
  "dayNumber": 4,
  "scheduledDay": "2024-06-04",
```

```
"activities": [
 {
  "startingTime": "21:00",
  "activity": {
   "name": "Opening Tunnel",
   "parentName": "null",
   "repeating": true,
   "duration": 0.5
  },
  "team": {
   "name": "UT01",
   "description": "",
   "members": {
    "UTma11": {
     "name": "John Doe",
     "surname": "Doe",
     "territorialUnit": "UTma-CML"
    }
   }
  },
  "vehicle": {
   "plate": "V.01.2019"
  },
  "impactedOBjects": []
 },
 {
  "startingTime": "21:30",
  "activity": {
   "name": "Ventilator Maintenance",
   "parentName": "",
   "repeating": false,
   "duration": 0.0,
   "steps": {
    "Start": {
     "Preparation": [
      {
       "[Ventilator Running OR with power]?": {
        "[Y]": [
         "Shut down Ventilator",
         "Go to 'Ventilator Running OR with power'"
        ],
        "[N]": [
         "Mark the place where maintenance is to be  performed",
         {
          "In parallel": [
           "Documentation is present",
           "Spare parts available",
```

```
          "Availability of tools and power"
         ]
        }
       ]
      }
     }
    ],
    "Main Ventilators Maintenance": {
     "In parallel": [
      "Bolts - Tightening Torque",
      "Bearings lubrication",
      "Surface Inspection",
      "Loose electrical cables check",
      "Engine Check",
      "Cleaning",
     ]
    },
    "Other Components Maintenance": {
     "In parallel": [
      "Channel Components Maintenance",
      "Flexible Links Leakage Check",
      "Shutter?",
      "Hydraulic System check",
      "Instrumentation check",
      "Artificial Ventilation check"
     ]
    },
    "Post Execution": [
     "Check Inventory",
     {
      "[Supply enough?]": {
       "[N]": "Order Missing Parts",
       "[Y]": "Ok"
      }
     },
     "Re-check installation of spare parts",
     "Replace deformed/worn out gaskets/old parts",
     "Ensure no tools left lying around",
     "Ensure platings were reinstalled",
     "Run a test"
    ]
   }
  }
 },
 "team": {
  "name": "E13",
  "description": "",
```

```
        "members": {
          "Esterno15": {
            "name": "Esterno15 VEN",
            "employer": "Zitron"
          }
        }
      },
      "vehicle": {
        "plate": "V.02.2019"
      },
      "impactedOBjects": [
        {
          "objectName": "VCN",
          "objectClass": "Ventilator",
          "manual": "link-to-manual.pdf",
          "report": "link-to-report.pdf"
        },
        {
          "objectName": "VCS",
          "objectClass": "Ventilator",
          "manual": "link-to-manual.pdf",
          "report": "link-to-report.pdf"
        }
      ]
    }
  ]
 }
 ]
}
]
```

Bibliography

[1] Bundesamt für Strassen ASTRA, Abteilung Strassennetze N Standards und Sicherheit der Infrastruktur SSI: *ASTRA 76001. Weisungen Sicherheit Betrieb Nationalstrassen*. Ausgabe 2018 V1.00.

[2] Bundesamt für Strassen ASTRA, Abteilung Strassennetze N Standards und Sicherheit der Infrastruktur SSI: *ASTRA 16050. Richtlinie operative Sicherheit Betrieb*. Ausgabe 2018 V1.50.

[3] Bundesamt für Strassen ASTRA, Abteilung Strassennetze N Standards und Sicherheit der Infrastruktur SSI: *Weisungen Umsetzung der Störfallverordnung auf den Nationalstrassen*, Ausgabe 2018 V2.00.

[4] C. Allen, M. Averso, C. Hargraves, S. McNeil. *Guide for the preservation of highway tunnel systems*, Final report, Gannett Fleming, Inc., University of Delaware, 2015.

[5] K. Chatzoglou, E. Rokou, I. P. Tatsiopoulos, K. Kirytopoulos. *On designing educational aids for users' safety in road tunnels. In Safety and reliability: methodology and applications*, ISBN 978-1-138-02681-0. Taylor & Francis group, London, 2015.

[6] A. Habermaier, J. Knapp, J. Leupolz, W. Reif. *Fault-Aware Modelling and Specification for Efficient Formal Safety Analysis*. In FMICS-AVoCS. Springer, 2016.

[7] U. Schtalheim, S. Ott, C. Heitz, G. H. Lilach. *Risk Based Maintenance (RBM): Minimierung der Nutzerrisiken und Betriebskosten mit einer risikobasierten Methode für den Unterhalt der BSA*, Bundesamt für Strassen ASTRA, doi: 10.21256/zhaw-3341, 2019.

[8] F. Sandrone, V. Labiouse J.-F. Mathier. *Data collection for Swiss road tunnels maintenance*. Geomechanik und Tunnelbau. Felsbau - Rock and Soil Engineering. 8-14, 2007.

[9] B. V. Farahani, F. Barros, P. J. Sousa, P. J. Tavares, P. M. G. P. Moreira. *A railway tunnel structural monitoring methodology proposal for predictive maintenance*. Struct Control Health Monit., 27:e2587. https://doi.org/10.1002/stc.2587, 2020.

[10] US Department of Transportation, Federal Highway Administration. Publication No. FHWA-HIF-15-005. *Tunnel Operations, Maintenance, Inspection, and Evaluation (TOMIE) Manual*, 07.2015.

[11] P. Liu, B. Deng. *Design of Integrated Management and Control System for Mechanical and Electrical Equipment of New Generation Highway (Road) Tunnel*. Tunnel Construction, 39(S1): 478-485, 2019.

[12] C. Ren. *A design model for enterprise asset management system of coal preparation plant*, 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), pp. 436-439, doi: 10.1109/EDT.2010.5496541, 2010.

[13] M. A. López Campos, L. Fumagalli, J. F. Gómez Fernández, A. C. Márquez, M. Macchi. *UML model for integration between RCM and CBM in an e-Maintenance architecture*, IFAC Proceedings Volumes, Volume 43, Issue 3, Pages 110-115, 2010.

[14] PIARC, Technical Committee 3.3 Road Tunnel Operation, *Best practice for life cycle analysis for tunnel equipment*, ISBN 978-2-84060-370-2, 2016

[15] PIARC, Technical Committee 3.3 Road Tunnel Operation, *Urban road tunnels – Recommendations to managers and operating bodies for design, management, operation and maintenance*, ISBN 2-84060-216-4, 2008.

[16] PIARC, Technical Committee C.4 *Road Tunnel Operation, Life cycle aspects of electrical road tunnel equipment*, ISBN 978-2-84060-272-5, 2012.

[17] PIARC, Technical Committee C.4 *Road Tunnel Operation, Recommendations on management of maintenance and technical inspection of road tunnels*, ISBN 978-2-84060-279-2, 2012.

[18] PIARC, Technical Committee 5 *Road Tunnels, Good Practice for the Operation and Maintenance of Road Tunnels*, ISBN 2-84060-175-3, 2004.

[19] H. Koç, A. M. Erdoğan, Y. Barjakly, S. Peker. *UML Diagrams in Software Engineering Research: A Systematic Literature Review*. MDPI Proceedings 2021, 74, 13. https://doi.org/10.3390/proceed- ings2021074013, 2021.

[20] J. P. S. da Silva, M. Ecar, M. S. Pimenta, G. T. A. Guedes, L. P. Franz, and L. Marchezan. *A Systematic Literature Review of UML-based Domain-Specific Modelling Languages*. SEAMS '18, May 28–29, 2018, Gothenburg, Sweden, 2018.

[21] I. Zafar, F. Azam, M. W. Anwar, W. H. Butt, B. Maqbool, A. K. Nazir. *Business Process Models to Web Services Generation: A Systematic Literature Review*, 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018, pp. 789-794, doi: 10.1109/IEMCON.2018.8615096, 2018.

[22] https://www.omg.org/spec/BPMN/2.0/

[23] https://www.omg.org/spec/UML/2.4.1/Superstructure/PDF

[24] I. Sommerville. Software Engineering, 10th Edition, Addison Wesley, 2021.

[25] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. ISBN 978-0-201-63361-0. 1995. *Despite the publication year, this book remains a recognized reference in the field.*

[26] R. Miles and K. Hamilton. *Learning UML 2.0*, O'Reilly, 2006.

[27] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*, Addison-Wesley, 2005.

Project conclusion

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für
Umwelt, Verkehr, Energie und Kommunikation UVEK
Bundesamt für Strassen ASTRA

FORSCHUNG IM STRASSENWESEN DES UVEK

Version vom 09.10.2013

Formular Nr. 3: Projektabschluss

erstellt / geändert am:	28.02.2025

Grunddaten

Projekt-Nr.:	BGT_20_11A_01
Projekttitel:	TUML - Using UML and BPMN to model the Structure, Operation and Maintenance of Road Tunnels. Comprehensive Digitalization of a Tunnel
Enddatum:	28.05.2025

Texte

Zusammenfassung der Projektresultate:

Das Projekt konzentrierte sich auf die Anwendung von UML (Unified Modelling Language) und BPMN (Business Process Model and Notation) zur Verbesserung des Managements und der Wartung von Strassentunneln.
Die Hauptziele waren:

- Entwicklung eines statischen digitalen Modells zur besseren Visualisierung und Dokumentation der Tunnelinfrastruktur.
- Verfeinerung der dynamischen Modellierung, um die Effizienz von Arbeitsabläufen und die Reaktionszeiten bei Notfällen zu optimieren.
- Nahtlose Integration zwischen den verschiedenen Teilsystemen des Tunnels zur Erleichterung des Datenaustauschs.
- Automatisierung der UML-Diagrammerstellung, um die Dokumentation und Prozesseffizienz zu verbessern.
- Erforschung von IoT- und KI-Technologien, um die Echtzeitüberwachung und vorausschauende Wartung zu verbessern.

Das TUML-Projekt hat seine Ziele erfolgreich erreicht und gezeigt, dass eine Digitalisierung von Strassentunneln mit UML, BPMN, IoT und KI möglich ist. Die nächsten Schritte umfassen die Pilotimplementierung von "Digital Twins" in ausgewählten Tunneln und die Entwicklung standardisierter digitaler Modelle für das Management intelligenter Verkehrsinfrastrukturen

Das Projekt TUML ist in vier Hauptarbeitspakete (Work Packages, WP) unterteilt, welche die verschiedenen Aspekte der Modellierung, Digitalisierung und technologischen Entwicklung abdecken:

WP1 - Projektaufbau und Zieldefinition
Dieses Arbeitspaket legte die Grundlagen für die Digitalisierung von Strassentunneln.
Schwerpunkte:
- Auswahl geeigneter Modellierungswerkzeuge (UML für statische Modellierung, BPMN für dynamische Prozesse).
- Definition der Projektziele und des methodischen Rahmens.
- Auswahl eines repräsentativen Tunnels, der als Fallstudie dient (Mappo-Morettina-Tunnel).
- Entwicklung eines Workflows für die Digitalisierung, einschliesslich Datenstrukturierung und Integration von Sensordaten.

WP2 - Statische Modellierung der Tunnelstruktur
In diesem Arbeitspaket wurde die statische Struktur eines Tunnels mithilfe von UML modelliert, um eine klare und standardisierte Dokumentation zu ermöglichen.
Schwerpunkte:
- Entwicklung detaillierter UML-Modelle, darunter Klassen-, Objekt-, Komponenten- und Verteilungsdiagramme.
- Erfassung und Strukturierung von Daten (Tunnelgeometrie, elektromechanische Systeme, Sicherheitsausrüstung).
- Integration mit bestehenden Datenbanken (KUBA, FA BSA), um Infrastrukturinformationen effizient zu verwalten.

WP3 - Dynamische Modellierung von Betriebs- und Wartungsabläufen
Dieses Arbeitspaket konzentrierte sich auf die dynamische Modellierung von Tunnelbetriebs- und Wartungsprozessen mit BPMN.
Schwerpunkte:
- Definition standardisierter Prozessabläufe.
- Erstellung von BPMN-Diagrammen zur Visualisierung dieser Prozesse.
- Integration mit UML-Modellen aus WP2, um eine vollständige digitale Tunnelabbildung zu ermöglichen.

WP4 - Zukunftsperspektiven und technologische Entwicklungen
Dieses Arbeitspaket untersuchte zukünftige Innovationen zur Verbesserung des Tunnelmanagements. Schwerpunkte:
- Einführung von Digital Twins – virtuelle Tunnelmodelle für Simulationen, Echtzeitüberwachung und prädiktive Wartung.
- Nutzung von KI-Technologien.
- IoT-Integration und Edge Computing zur Reduzierung von Latenzzeiten in Tunnelsteuerungssystemen.
- Erforschung von 6G- und Quantennetzwerken für ultra-schnelle Datenverarbeitung und Kommunikation.

Das grösste Hindernis für die Verwirklichung der im Projekt entwickelten Konzepte, ist die Tatsache, dass es keinen digitalen Rahmen gibt, in den die aus der Studie resultierenden Daten und Modelle eingefügt werden können.

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für
Umwelt, Verkehr, Energie und Kommunikation UVEK
Bundesamt für Strassen ASTRA

Zielerreichung:

WP1 - Projektaufbau und Zieldefinition
Auswahl geeigneter Modellierungswerkzeuge: UML und BPMN wurden als beste Methoden für die Darstellung der Tunnelstruktur bzw. der betrieblichen Abläufe identifiziert.
Identifizierung eines repräsentativen Tunnels: Der Mappo-Morettina-Tunnel (5,5 km) wurde aufgrund seiner Komplexität und der hohen Datenverfügbarkeit ausgewählt.
Entwicklung eines Digitalisierungs-Workflows: Festlegung des Prozesses für Datenerfassung, statische Modellierung (UML), dynamische Modellierung (BPMN) und IoT-Integration.

WP2 - Statische Modellierung der Tunnelstruktur
Erstellung eines detaillierten UML-Modells: Entwicklung von Klassen-, Objekt-, Komponenten- und Verteilungsdiagrammen zur Dokumentation der Tunnelinfrastruktur.
Integration mit bestehenden Datenbanken: Nutzung von Plattformen wie KUBA und FA BSA, um Infrastrukturinformationen zu speichern und abzurufen.
Bewältigung technischer Herausforderungen: Schaffung eines skalierbaren und wiederverwendbaren UML-Modells, um Probleme mit heterogenen Datenformaten und Systemkompatibilität zu lösen.

WP3 - Dynamische Modellierung von Betriebs- und Wartungsabläufen
Erstellung von BPMN-Diagrammen zur Modellierung der wichtigsten Tunnelbetriebsprozesse.
Ausrichtung an den UML-Modellen: Die BPMN-Diagramme wurden mit den UML-basierten Tunnelstrukturmodellen verknüpft.
Standardisierte, automatisierbare Prozesse: Die definierten Abläufe können nun mit IoT-basierten Steuerungssystemen automatisiert werden.

WP4 - Zukunftsperspektiven und technologische Entwicklungen
Entwicklung eines Digital-Twin-Frameworks: Erstellung eines virtuellen Tunnelmodells zur Simulation, Überwachung und vorausschauenden Wartung.
Implementierung KI-gesteuerter Systeme zur Effizienzsteigerung.
Integration von IoT-Netzwerken: Smarte Sensoren, Edge Computing und verbesserte Cybersecurity für ein effizienteres Management.
Vorbereitung auf Zukunftstechnologien: 6G und Quantenkommunikation für extrem schnelle Datenübertragung zwischen Tunnelsteuerungssystemen.

Folgerungen und Empfehlungen:

Das Projekt wurde auf theoretischer Ebene durchgeführt und bildet die Grundlagen für eine künftige Umsetzung auf einer konsolidierten Softwarebasis, insbesondere für einen digitalen Zwilling eines Tunnels sowie eines Objekts im Allgemeinen. Das vorliegende Projekt hatte nicht die Umsetzung eines solchen Softwareframework zum Ziel, der die logische Folge der durchgeführten Arbeiten sein sollte:

- Identifizierung eines Rahmens für die Implementierung eines oder mehrerer digitaler Zwillinge.
- Schnittstelle des digitalen Zwillings mit dem realen Objekt.
- Anwendung der Modellierung des vorliegenden Projekts auf das digitale Modell.
- Standardisierung von Schnittstellen und Datenmanagement.

Sobald das digitale Modell erstellt und mit Daten gefüllt ist, wird es möglich sein, Applikationen für die Durchführung sehr spezifischer Aufgaben zu erstellen, wie z. B. die ordentliche, prädiktive und ausserordentliche Wartung eines Tunnels.

Publikationen:

G. Nodiroli, M. Lanza, M. Artiglia, F. Socchi, D. Paolo Tua, C. Pedrinis.

TUML - Using UML and BPMN to model the Structure, Operation and Maintenance of Road Tunnels. Comprehensive Digitalization of a Tunnel.

Forschungsauftrag ASTRA/BGT_20_11A_01

Der Projektleiter/die Projektleiterin:

Name: Nodiroli   Vorname: Gian Paolo

Amt, Firma, Institut: Lombardi AG Beratende Ingenieure

Unterschrift des Projektleiters/der Projektleiterin:

Nodiroli G.P.

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für
Umwelt, Verkehr, Energie und Kommunikation UVEK
Bundesamt für Strassen ASTRA

FORSCHUNG IM STRASSENWESEN DES UVEK
Formular Nr. 3: Projektabschluss

Beurteilung der Begleitkommission:

Beurteilung:

Der Einsatz von UML und BPMN zur Modellierung von Infrastrukturen und Prozessen stellt einen Ansatz dar, der potenziell eine Vereinheitlichung der Tunnelverwaltung ermöglicht und die Grundlage für die Schaffung eines vollständigen digitalen Modells bildet, das über den gesamten Lebenszyklus eines Objekts hinweg genutzt werden kann.

Das Projekt erfolgte in vier aufeinanderfolgenden Arbeitspaketen, deren Hauptziel es war, die Machbarkeit und Wirksamkeit einer Modellierung auf der Grundlage standardisierter, im IT-Bereich seit Langem etablierter Sprachen zu bewerten.

Im Rahmen des Projekte wurde ein Beispielobjekt ausgewählt, für das eine Modellierung auf Basis der verfügbaren Dokumentation durchgeführt wurde. Das Ergebnis verdeutlichte, wie rasch sich ein Modell zu einer umfassenden digitalen Repräsentation weiterentwickeln kann – und wie essenziell dabei ein klar strukturierter Rahmen (Framework) ist, um dessen Einführung und Verwaltung effektiv zu gestalten. Die Nutzung eines solchen Frameworks selber war im Rahmen des vorliegenden Projekts ebenso wenig vorgesehen wie die operative Umsetzung eines digitalen Modells; dies stellte das grösste Hindernis für die praktische Anwendung der Modelle auf das Beispielobjekt dar, ohne jedoch die übergeordneten Ziele der Studie zu beeinträchtigen.

Für jedes Arbeitspaket wurde eine Schlussfolgerung formuliert, die den progressiven Fortschritt des Projekts ermöglichte. Im Bereich der Tunnelautomatisierungssysteme wurden Aspekte im Zusammenhang mit der neuen SA-CH-Architektur analysiert, deren normative Weiterentwicklung eine gewisse Verzögerung im Projektzeitplan verursachte.

Die vollständige Digitalisierung des Tunnels ist ein ambitioniertes Ziel, das mit vielfältigen Herausforderungen einhergeht: Diese betreffen unter anderem die Vielzahl bestehender IT-Lösungen, die Notwendigkeit einer wartungsfreundlichen und erweiterbaren Modellstruktur sowie deren kontinuierliche Anpassung an die sich dynamisch entwickelnden betrieblichen und technologischen Anforderungen der Branche.

Das Projekt umfasst einerseits eine theoretische Komponente, die weitgehend technologieunabhängig ist, und andererseits eine ausgeprägte technische Komponente, die eng mit dem aktuellen Stand der Technik in den Bereichen IT, Datenmanagement und Telekommunikation verknüpft ist. In diesem Zusammenhang wurden durch das Projekt wesentliche Grundlagen geschaffen, die zukünftige Entwicklungen in Richtung einer vollständigen Umsetzung eines digitalen Tunnelmodells ermöglichen könnten.

Umsetzung:

Die Realisierung der im Projekt entwickelten Konzepte setzt zwangsläufig eine leistungsfähige, wartungsfreundliche und moderne IT-Infrastruktur voraus. Die erarbeiteten Modelle sowie deren Anwendung auf das Beispielobjekt sind in ein System einzubettern, das eine dynamische Visualisierung, gezielte Abfragen und eine effektive Nutzung ermöglicht.
Das Projekt liefert darüber hinaus einen bedeutenden Beitrag, indem es eine strukturierte Methode zur Verknüpfung heterogener Informationen auf Basis UML-basierter Metamodelle entwickelt. Als greifbares Ergebnis entstand eine modellhafte Umsetzung in Form übersichtlicher, grafisch visualisierter UML- und BPMN-Diagramme, die das Potenzial der Methode eindrucksvoll demonstrieren – auch wenn der Fokus vorerst auf der theoretischen Ebene liegt.

weitergehender Forschungsbedarf:

Das Projekt eröffnet Perspektiven für weitere Forschung im Bereich der Umsetzung des digitalen Zwillings. Die wesentlichen Anforderungen an ein unterstützendes Framework, das die Implementierung des digitalen Zwillings begleiten soll, wurden definiert. Die Identifikation einer Plattform, die all diesen Anforderungen gerecht wird und eine effiziente sowie konsistente Nutzung des digitalen Zwillings gewährleistet, stellt eine zentrale Herausforderung dar und ebnet den Weg für neue Forschungsprojekte.

Einfluss auf Normenwerk:

Die Öffnung der Automatisierungssysteme gegenüber neuen digitalen Modellansätzen ist derzeit in den geltenden Richtlinien nicht vorgesehen. Die normativen Prioritäten liegen aktuell auf der Effizienz und Sicherheit der Anlagen zum Schutz der Verkehrsteilnehmenden. Eine standardisierte Schnittstelle zu einem Framework für den digitalen Zwilling ist daher bislang nicht vorgesehen.

Der Präsident/die Präsidentin der Begleitkommission:

Name: Avanti Vorname: Roberto

Amt, Firma, Institut: GESTE Engineering SA

Unterschrift des Präsidenten/der Präsidentin der Begleitkommission:

GESTE
GESTE ENGINEERING SA
VIALE OFFICINA 8
CH – 6500 BELLINZONA
T +41 91 600 19 58 INFO@GESTE.GROUP WWW.GESTE.CH