Linked Data for Structural Diagnostics: A Semantic Framework for Sustainable Infrastructure Management

Paul-Christian Schuler[1,*], Mahsa Mirboland[1], Yannic Stark[2], Abdullah Al Mohammad[3] and Christian Koch[1]

[1]*Chair of Intelligent Technical Design, Bauhaus-Universität Weimar, Germany*
[2]*Marx Krontal Partner GmbH , Zum Hospitalgraben 2, 99425 Weimar, Germany*
[3]*customQuake, Donnerstraße 10, 22763 Hamburg, Germany*

Abstract

The growing demand for sustainable infrastructure management highlights the need for effective integration and utilization of diagnostic data. This research examines the application of Linked Data principles in the construction industry, proposing a framework that may replace traditional workflows with modern semantic data approaches. The study investigates the potential of Linked Data methodologies within the context of structural diagnostics, emphasizing the role of ontologies in enhancing the semantic representation of diagnostic processes. Addressing interoperability and reusability challenges, the present work focuses on developing workflows that manage and leverage Linked Data, the integration of ontologies, and the implementation of API-driven data retrieval. The proposed approach comprises ontology design, data modeling, and software development methods. The Structural Information Ontology (SIO) is defined to provide a semantic framework for diagnostic processes, while the Information Container for Linked Document Delivery (ICDD) ensures an effective data organization. A prototype API is developed to enable the querying and processing of containerized data, validated through a non-destructive test employed for bridge inspections. The findings show that the proposed workflow effectively integrates and queries structural diagnostics data using Linked Data principles. Moreover, the SIO ontology is proven to be modular and extensible, supporting flexible semantic representations. It is shown that the API facilitates efficient data extraction and processing, highlighting data traceability and decision-making capabilities of the proposed framework. This research advocates for the adoption of Linked Data methods in the construction industry by introducing a scalable and interoperable framework for data relevant to structural diagnostics, paving the way for improved semantic data integration and consequently for a sustainable infrastructure management.

Keywords

Structural Diagnostics, Linked Data, Ontologies, Structural Information Ontology (SIO), Information Container for Linked Document Delivery (ICDD)

1. Introduction

Engineering structures as vital components of modern infrastructure play a pivotal role for an effective economy and a functional civilization. Therefore, ensuring a reliable performance and an extended service life of the built environment, e.g., of transportation networks and power plants, is of crucial importance. To this end, maintenance management strategies are employed to achieve long-term durability, balancing criteria relevant to the cost efficiency, service life and environmental sustainability of the infrastructure.

As essential tools for maintenance management projects, structural diagnostics and inspection tests can provide a wide range of structural information, such as information corresponding to the structural integrity (i.e. the as-is state), the remaining service life of a structure, and requirements for reconstruction processes. Moreover, structural diagnostics methods help to prevent costly and unsustainable replacements. Inspection tests comprise on-site assessments and laboratory tests that are typically conducted in accordance with established guidelines and national standards, e.g., the DIN

LDAC 2025: 13th Linked Data in Architecture and Construction Workshop, July 09–11, 2025, Porto, Portugal

*Corresponding author.
paul-christian.schuler@uni-weimar.de (P. Schuler); mahsa.mirboland@uni-weimar.de (M. Mirboland); yannic.stark@marxkrontal.com (Y. Stark); aam@customquake.com (A. A. Mohammad); c.koch@uni-weimar.de (C. Koch)
0000-0003-3758-1286 (P. Schuler); 0000-0003-0238-6050 (M. Mirboland); 0000-0002-6170-7611 (C. Koch)

1076 [1] Standard in Germany. Data collected from inspection tests serve as the basis for delivering assessment reports and decision-making instructions relevant to maintaining structural integrity and ensuring public safety.

Structural diagnostics and inspection tests are conducted at discrete time intervals and often employ non-destructive testing (NDT) methods, such as visual inspections and ultrasonic measurements. In addition, destructive testing (DT) techniques that entail the removal of small samples for analysis either on site or in a laboratory are commonly used in diagnostics projects.

The increasing number of diagnostics projects for the aging infrastructure globally, as well as the growing demand for sustainable construction practices necessitate the modernization and digitalization of inspection workflows. Despite the importance, many inspection workflows are currently conducted through analog and manual processes with little integration of digital tools for data management. Consequently, inspection data are being recorded in diverse and isolated formats that are difficult to reuse or analyze, increasing the risk of information loss. Recent advances in Building Information Modeling (BIM) may offer solutions to overcome challenges associated with the digitalization of inspection workflows. Additionally, leveraging Linked Data concepts for collected inspection data provides the potential to dynamically share cross-divisional structural information and to present a comprehensive overview of as-is states.

This paper introduces a framework for digitalization of inspection workflows, with the goal of improving accessibility and reusability of inspection data. For a seamless integration of heterogeneous inspection data and to ensure semantic interoperability of data formats involved in inspection workflows, the Structural Information Ontology (SIO) is developed. The SIO provides the basis for storing inspection data in linked data containers. Moreover, an authoring application for reading and creating linked data containers in alignment with the ISO 21597 [2, 3] Standard is introduced. Lastly, an application programming interface (API) is presented that can access and query data from linked data containers.

2. Background and related work

2.1. Semantic Web frameworks

A semantic triple is a core data model in the Semantic Web [4]. It comprises three elements: the subject (the resource being described), the predicate (the relationship or property), and the object (a literal value or another resource). These triples are defined using the Resource Description Framework (RDF), which structures data for machine readability. The definition of each instantiated component is expressed through the use of HTTP URIs, which facilitates their unique identification. A triplestore is a database that stores, manages, and queries semantic triples. By leveraging RDF and its subject–predicate–object model, triplestores enable the integration and retrieval of semantically rich information from various sources.

Ontologies are a formalized and structured representation of knowledge within a particular domain. An ontology defines terms, their meanings and the relationships between them, thereby facilitating a common understanding and interoperability between systems. Ontologies are frequently employed in the field of computer science, particularly within the context of the Semantic Web, with the objective of rendering machine-readable and linkable data [5].

The RDF Schema (RDFS) provides a straightforward model for the description of hierarchies and types within ontologies. As an extension of the RDF, RDFS defines fundamental concepts such as classes, properties and relationships, thereby enabling developers to construct rudimentary knowledge structures [6].

Building upon RDFS, the Web Ontology Language (OWL) is employed to delineate more intricate and logically exact relationships. It is frequently employed, when more rigorous modeling requirements are necessary, i.e. via defining constraints, cardinalities or class operations [7].

A variety of query languages have been developed to facilitate the aggregation of information from data structured according to standardised schemas. Among these, SPARQL[1], endorsed by the World Wide Web Consortium (W3C), emerged as the leading standard, effectively replacing earlier languages such as RDQL and SeRQL. In parallel, the openCypher[2] initiative has introduced a compelling alternative to SPARQL. Initially designed by Neo4j Inc. for adopting graph databases, the Cypher query language was subsequently made available through the OpenCypher initiative [8]. This transition has expanded its applicability across diverse platforms, establishing Cypher as a competitive and versatile option within the field of query languages.

2.2. Ontologies for the AECO industry

A variety of specialized ontologies are available for applications in the Architecture, Engineering, Construction and Operation (AECO) industry. In the construction sector, ontologies are created to address diverse tasks and data requirements that arise across different phases and domains of construction projects. Ontologies may focus on distinct areas of expertise, including *spatial modeling* (e.g., the Building Topology Ontology (BOT) [9] and Brick Ontology [10]), *process and building management* (e.g., RealEstateCore [11], Org [12], and the Financial Industry Business Ontology (FIBO) [13]), *sensor integration* (e.g., the Semantic Sensor Network Ontology (SSN) [14] and its lightweight counterpart (SOSA) [15]), and *data exchange* (e.g., the Information Container for Linked Document Delivery (ICDD) [2, 3], ifcOWL [16], and DCAT2 [17]).

In the domain of data integration and the provision of building-related information, several scientific contributions have focused on ontologies specifically designed to support the storage, annotation, and dissemination of sensor-generated data. Notable examples include approaches targeting structural health monitoring, which emphasize the semantic modeling of sensor networks and the interpretation of measurement data [18, 19, 20]. In addition, other research efforts have proposed ontologies for representing surface-level damage in structural elements [21], as well as for describing material testing procedures and the characterization of their specific properties [22]. Furthermore, some ontologies have been developed to capture and formalize the repair and rehabilitation processes themselves [23], thereby extending semantic modeling to include maintenance and operational workflows.

While many of these ontologies are used independently, there are instances, where ontologies are linked to one another, enabling cross-domain modeling. However, not all ontologies are aligned with one another, which can leave gaps in information and definitions, reducing the overall effectiveness of the semantic integration. Furthermore, aligning multiple ontologies is a complex task, particularly when the semantic integration is required across different knowledge domains. Establishing connections between ontologies for every use case is often impractical and inefficient.

To address this challenge, the Design and Construction Ontology (DICON) was developed as a central framework that simplifies the alignment process [24]. Rather than requiring direct connections between ontologies, new or existing ontologies may be integrated by establishing a single linking point with DICON. This method reduces complexity and ensures a more systematic and scalable approach to the ontology alignment.

In addition, DICON introduces proprietary structures, which are derived from various knowledge domains relevant to the construction industry. The built-in structures enable the connection of additional ontologies that might not directly align with others, ensuring flexibility, extensibility, and a comprehensive knowledge coverage essential for construction workflows.

[1] https://www.w3.org/TR/sparql11-query/
[2] https://opencypher.org/

2.3. Information Container for Linked Document Delivery (ICDD)

The ICDD is an open framework based on the ISO 21597 standard, which is designed to support the exchange and integration of heterogeneous and distributed data. The ICDD supports interoperability across various domains by employing linked data concepts, thereby enabling seamless connections between diverse data sources and file formats. The ICDD is implemented as a compressed ZIP archive (**.icdd*), in which data and metadata are organized according to a standardized ontology. Its ontology-driven structure adheres to Linked Data principles, enabling modular and reusable data representation [25].

The ICDD offers a number of significant advantages that enhance its utility and applicability within the construction industry. The framework utilizes the RDF and associated Semantic Web frameworks to establish links at both document and sub-document levels. For example, linking specific attributes within an Industry Foundation Classes (IFC) file to an external file (for example, the PDF project report) is possible. Provides a standardized mechanism for organising, linking and exchanging data, the ICDD can be implemented at different stages of the lifecycle of a structure. Through seamless integration and robust interoperability across tools, formats and disciplines, the ICDD improves workflow efficiencies and collaborations, and thus, can contribute vastly to the digital transformation of the construction industry.

The ICDD structure is divided into three primary components:

- **Index file** acts as the central registry for all contents within the container. It lists documents, their associated metadata, and semantic relationships between them.
- **Ontology resources** comprises in general two separate resources, which define the container framework and semantic linking capabilities:
  - **Container ontology**, specifying the structure of the container, metadata for contents, and the representation of documents;
  - **Linkset ontology**, defining semantic relationships and links between container documents and enabling advanced linking mechanisms.
- **Payload** contains the actual data and semantic triples used for linking:
  - **Payload documents**, which may store all project files, e.g., text-based documents, BIM-based data (IFC models, BCF files, ...), and image-based data;
  - **Payload triples**, which contain RDF-serialized data representing semantic relationships and linksets.

Recent research underscores the rising importance of the ICDD in the AECO domain. Specifically, ICDD is being adopted more frequently to manage and exchange data related to semantic digital twins. Current investigations concentrate on containerization methods for heterogeneous data and the integration of BIM with time-series data [26]. Additionally, there are ongoing efforts to advance the standardization of ICDD and to develop tools that support the creation and validation of ICDD containers [27]. A significant research direction involves the alignment of ICDD with established standards such as Industry Foundation Classes (IFC) and Linked Building Data (LBD), as well as the specification of machine-readable Exchange Information Requirements (EIR) tailored for ICDD-based workflows [28]. Additionally, dynamic information containers are being investigated, with the objective of enabling the automatic updating of linked data as part of evolving digital ecosystems [29].

3. Semantic framework for Structural Information data

In this section, the semantic framework for data and process management pertaining to inspection projects is presented. The primary goal of the proposed framework is to improve the integration and retrieval of data for structural diagnostics by leveraging a linked-data approach that offers a standard-based storage for heterogeneous data produced from DT and NDT projects.

Inspection tests generate diverse data formats that are frequently vendor-dependent and tailored to specific process types and equipment utilized by contractors. Furthermore, diagnostics projects are often executed through file-based workflows, where variations in data formats and the quantity of information can lead to miscommunication and knowledge gaps within project workflows. In many cases, the data is produced in formats—such as text-based forms that are scanned and digitalized—that require human interpretation by project stakeholders before being transferred to subsequent workflow stages or actors. Collectively, these factors contribute to the complexity and heterogeneity of inspection project data, which may be represented in a standard-based and unified structure via ontologies.

The BIM workflow for inspection projects was previously introduced in an earlier study, which detailed the intervals of data exchange among project stakeholders while assuming that inspection data are stored within proprietary common data environments or databases [30]. In a subsequent study, a data container approach was proposed to address the limitations of isolated file systems by organizing and interlinking files with their associated information. This approach demonstrated improvements in data structure through the addition of a relational layer via links [31]. However, it was determined that the linking of files establishes the relationship between different data but lacks a semantic description, which may comprehensively describe the diagnostic results, project tasks, actors involved, and objectives in the structural assessment and diagnostic processes. To this end, the present study focuses on the semantic enrichment of inspection data by extending and adapting existing Semantic Web ontologies for inspection data and by integrating this ontology into data containers. Figure 1 illustrates the work packages within the proposed framework, which is detailed in the following paragraphs.

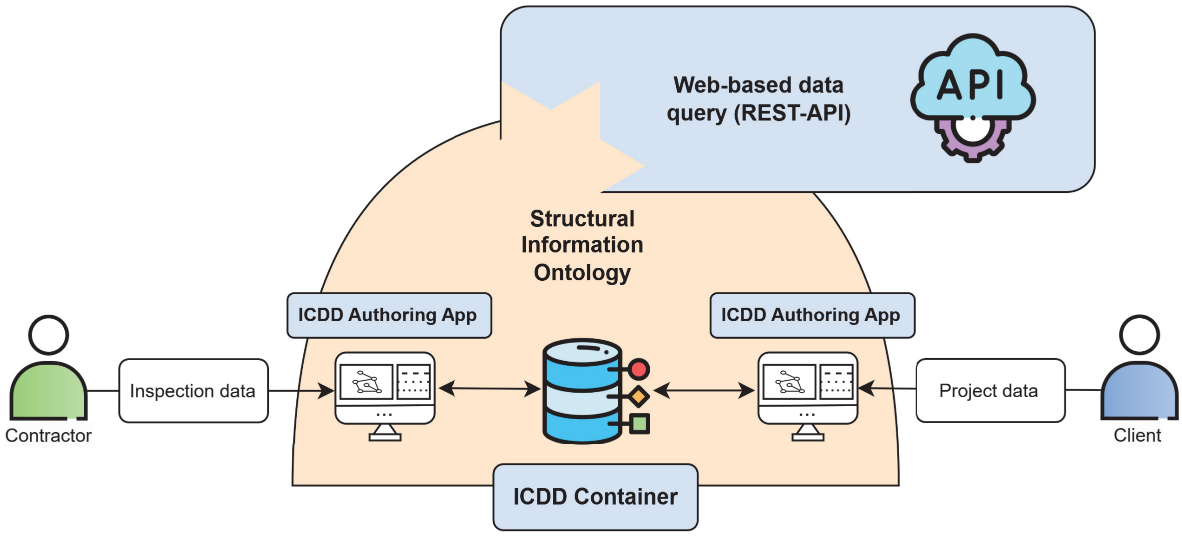

Figure 1: Abstract view of the proposed framework

The ontology *SIO* is following a hierarchical top-down logic that enables a systematical mapping of the entire structural diagnostics process. Aligning with existing standards and best practices in the field, as well as with the ontologies introduced in Section 2.1, the SIO provides a unified framework for managing diagnostic information for seamless implementations within inspection projects, while increasing the interoperability with other data models and systems across various contexts of structural diagnostics. Furthermore, metadata requirements for SIO classes, properties and relationships have been established to uniquely describe each inspection test, facilitating traceability and ensuring precise management of the project information. As illustrated in Figure 2, the hierarchical structure comprises four primary domains. The management domain encompasses project-specific data and contractual documents, such as the type of the diagnostics process to be conducted, test objectives, stakeholders and respective tasks, and data exchange requirements. General information relevant to the structure under inspection is also provided in the management domain.

The Localization domain describes the spatial distribution of the collected information, which is

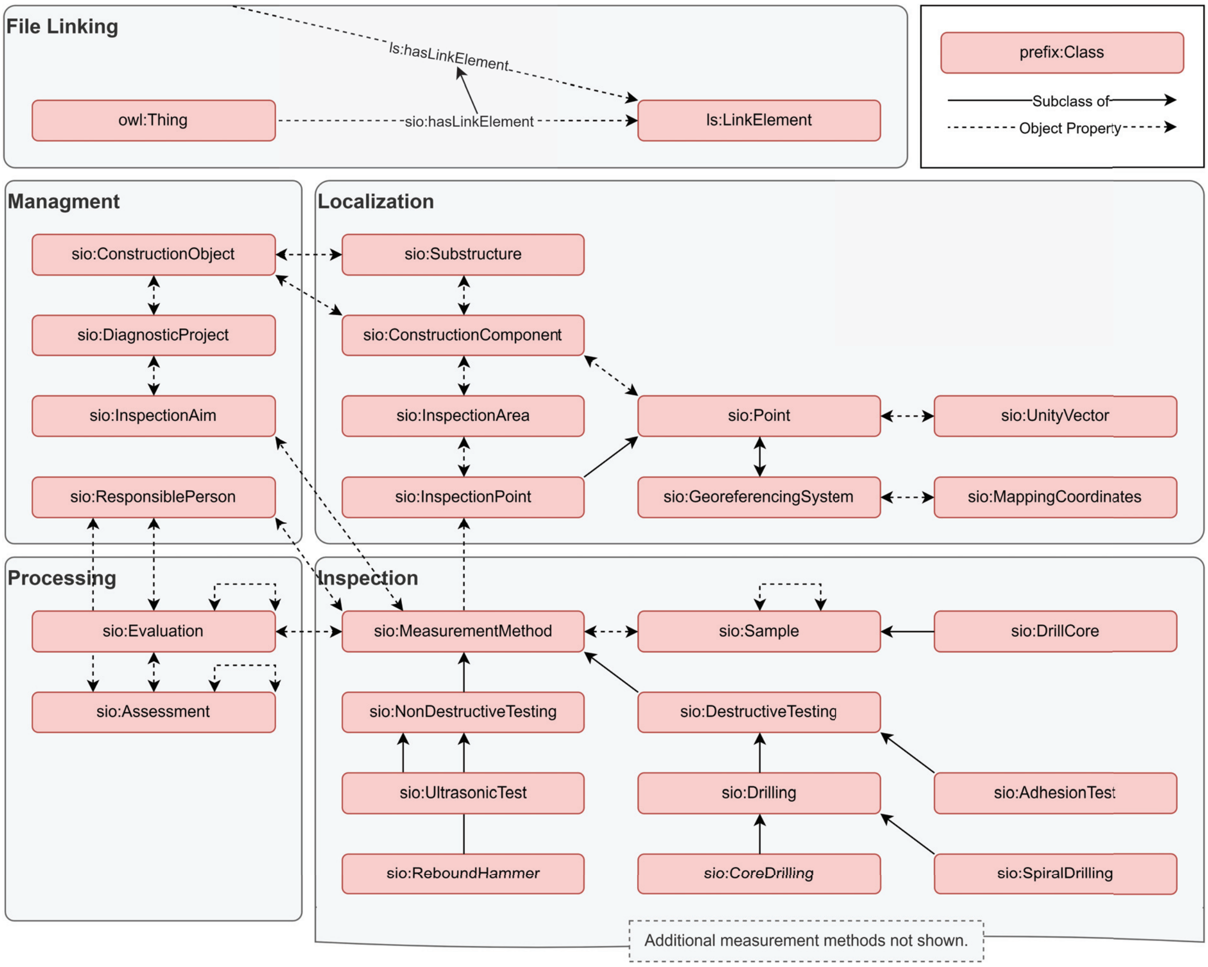

Figure 2: Representation of the classes and object properties within the Structural Information Ontology (SIO).

essentially a semantic representation of inspection areas composed of multiple inspection points and the respective geographic positions. Additionally, geographic localization is enabled by a georeferencing system module, which facilitates the mapping between different coordinate systems if needed. The inspection domain comprises entities that may further define the contractor-specific information, test processes to be conducted, test samples and test results. In this domain, DT and NDT methods are described in detail, including inspection parameters, e.g., diameter of a core sample during drilling, and device-specific information. This domain also includes entities that describe material properties and numerical analyses. The Processing domain focuses on the analysis and representation of the results derived from the inspection area. A distinction is made here between the evaluation and assessment of the results. Evaluation refers to the aggregation and processing of data to gain new, context-specific insights. For instance, raw data from a lateral ultrasonic scan can be processed to create a 2D visualization of the results. Assessment, on the other hand, involves interpreting these results by engineers to determine parameters for structural calculations or to derive recommendations for restoration actions. It is worth noting that the assessment and evaluation classes enable recurring data aggregation and the long-term combination of past and newly acquired data. The clear structure and semantic precision of the ontology make it possible to create a uniform and at the same time detailed modeling of the diagnostics process. The SIO thus offers a base for integrating and interpreting complex structural information data. Characterized by a comprehensive and granular structure, the SIO comprises a total of 1174 axioms, which are distributed across the following elements:

- 34 classes for categorising entities
- 40 object properties for defining relationships between entities

- 140 data properties for describing properties and values

The axioms make it possible to model detailed dependencies and relationships between various information sources within structural diagnostics. This is particularly important when modeling relationships or constraints between inspection processes and the respective outcomes.

The integration of *rdfs:label* and *rdfs:comment* is intended to make the ontology understandable and comprehensible by providing additional descriptions or translations. The SIO provides a terminology that can be adopted by different professionals, such as engineers, IT specialists and building diagnosticians, and can serve as a linguistic basis for collaboration.

Given the diverse range of heterogeneous data generated in diagnostics projects, the SIO is equipped with the linking capabilities of the Linkset (ls) ontology from ISO 21597, as utilized by the ICDD. For this purpose, the SIO contains the object property *sio:hasLinkElement*, which can be linked to all classes within the ontology, implemented by assigning it the domain *owl:Thing*. To further leverage the capabilities of the Linkset ontology, *sio:hasLinkElement* was defined as a subclass of *ls:hasLinkElement*, which facilitates the linking of *sio:Sample* object with a file, for example, with an image of a test sample. Additionally, the *sio:ConstructionObject* class from the Management domain is defined as a subclass of the *dicon:Building* class from the DICON ontology, enabling seamless integration between the two ontological frameworks.

4. Implementations

Building upon the proposed ontology, ICDD data containers may be created for DT and NDT projects. The ICDD provides built-in ontologies for linking information, offering a solid foundation for integrating and managing diverse datasets. While ICDD offers an effective system for centrally storing ontologies, knowledge graphs and files, current tools for creating and editing such containers are limited. To address this gap and to meet the expanded requirements of the methodology, a dedicated authoring software tool has been developed. This tool facilitates the creation and management of knowledge graphs within the container and supports the instantiating and linking of additional ontologies.

The ICDD tool presents structured project data in a standardised knowledge graph. To retrieve information from the container, an Application Programming Interface (API) following design principles of the Representational State Transfer (REST) architecture has been developed. The REST API allows precise information queries through a web-based user interface, enabling users to search for specific values within the knowledge graph. Additionally, the REST API intends to support long-term data utilization and accessibility through comprehensible relationships between various data types and information.

The following subsections describe implementations of the ICDD authoring software tool and the REST API for retrieving data from ICDD containers.

4.1. ICDD Authoring Application

To ensure the creation of an ISO-compliant ICDD, an application has been developed that also includes the instantiation of additional ontologies[3]. The initial prototype has been implemented in Python, with the *rdflib*[4] library playing a pivotal role in reading and writing to *triplestores*. The application aims at facilitating practical implementations of ICDD containers and graph-based data storage for use cases in structural diagnostics. To enhance the usability of the tool, a graphical user interface (GUI) has been designed, as shown in Figure 3, using *PyQt5*[5]. A graph structure is extracted using *rdflib*, structured with *Graphviz*[6], and rendered in a *PyQt5 GraphicsView*.

[3]https://gitlab.uni-weimar.de/professur-intelligentes-technisches-design/opensim-icdd-tool.git
[4]https://rdflib.readthedocs.io/en/stable/
[5]https://wiki.python.org/moin/PyQt
[6]https://graphviz.org/

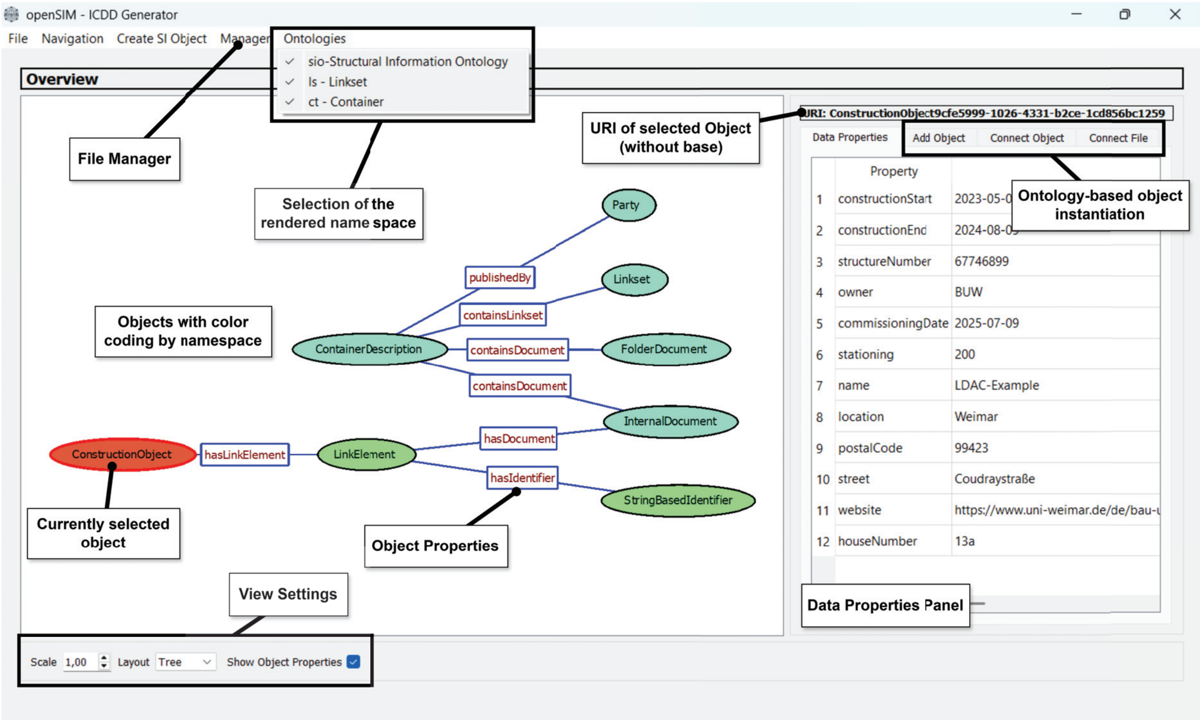

Figure 3: Overview of the main window of the application.

Figure 4 depicts the main components in the application. The first component includes the creation of a new container according to the ISO 21597 standard or loading an existing container. In both cases, the validation process of the container follows. During editing, users can select data that is to be displayed on the simplified graph view. Moreover, users may select a preferred ontology relevant for the creation of specific "areas of interest" according to project goals and deliverables.

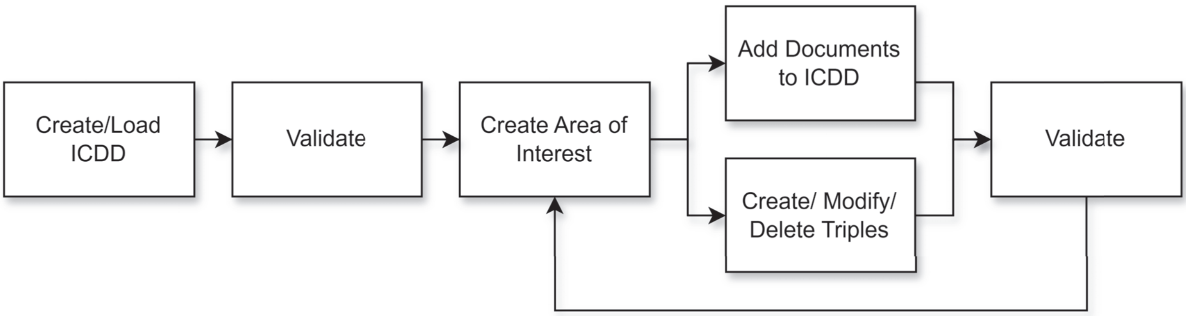

Figure 4: Primary functions of the developed ICDD application.

After initialising, the application includes three primary modules that facilitate the main tasks associated with ICDD editing:

- **Triples management**, allowing users to create, modify, or delete triples within the triplestore;
- **File integration**, enabling users to add files to the ICDD; and
- **Container validation**, verifying the structural and semantic integrity of the container.

The editing of the triplestore adheres to the ontologies available within the container. During the validation process, schemas are parsed that form the foundation for instantiating data. When a user selects an object, ontologies specify available data properties for creating new objects or linking existing objects to the selected object. Similarly, when creating a new object, the schema dictates permissible data properties. Users can also modify or delete existing objects and the associated properties.

Adding files to the ICDD can be performed either manually or via the application interface. The primary requirement is that files must be placed in designated directories as outlined in Section 2.3. Integration of files into the ICDD knowledge graph structure occurs during the validation process.

The validation module plays a twofold role in maintaining the ICDD integrity. First, it verifies the file structure and updates the *Index.rdf* file accordingly. Secondly, all knowledge graph files are validated against the schemas used. Despite the fact that schemas are modeled according to the assumption of an open world and non-unique names, the validation process aims at identifying instances, in which cardinality constraints are specified in the used ontology. This process guarantees compliance with structural requirements and constraints, such as property cardinalities. The Validation process is triggered after every modification to the container, providing immediate feedback to the user about potential errors.

4.2. The REST API for data extraction from ICDD containers

The API for processing and providing data from ICDD containers has been implemented in compliance with the ISO 21597 standard. The API's main purpose is to extract, index, and expose content from ICDD containers through the standardized Open Data Protocol[7] (OData) v4 endpoints. In the prototype implementation, a single ICDD container is utilized and stored within a predefined system directory. The API extracts container content by unpacking the ZIP64 file, then analyzes and classifies the resulting files and directories based on their function.

The API identifies and processes all RDF files contained within the ICDD container. The indexing process is based on namespace definitions, which act as unique identifiers. The RDF files are subsequently loaded into an RDF model set:

- **Index file analysis**: The index file is analyzed first to extract references to other models and to incorporate their data into the RDF model set.
- **Ontology and Linkset integration**: Ontology files are transformed into individual RDF models. Linksets, on the other hand, are processed to integrate the relationships defined within the RDF files into the overall model set.

Depending on system requirements, the RDF model set can be operated in two modes. The *in-memory graph* mode is suitable for small-scale applications, in which query performance is prioritized, whereas the *persistent triplestore* mode is used for scenarios requiring enhanced scalability and long-term data persistence.

The API supports both SPARQL queries and program-controlled access methods, enabling precise and efficient querying of the RDF data model. The extracted information is subsequently mapped into Data Transfer Objects (DTOs) to facilitate a structured processing. The DTOs are converted into domain objects that encapsulate the core business logic of the application. The domain objects form the foundation for exposing data via OData v4 endpoints. To this end, the implementation leverages the *Spring Boot* backend development and deployment framework, as well as the *Apache Olingo Framework*, which creates OData-compliant endpoints supporting advanced features, e.g., filtering, sorting, and pagination.

[7] https://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html

5. Case Study

The proposed method was tested using various DT and NDT project data to assess its applicability and was repeatedly refined. In a case study, a destructive inspection project was conducted on an in-service bridge to determine material strength in bridge piers, namely the compressive strength class of the concrete material. For this purpose, core samples were extracted from selected locations and were analyzed in a laboratory. Data created during the project planning phase, as well as evaluation documents and results (both raw data and laboratory analyses) collected after inspections were gathered in the tripleset. An insight into the structure of the triplestore within the ICDD container is provided in Figure 5. Here, classes and object properties are visualized in a graph structure, while the associated files are represented as rectangles. Data properties listed on the right-hand side describe the asset under investigation.

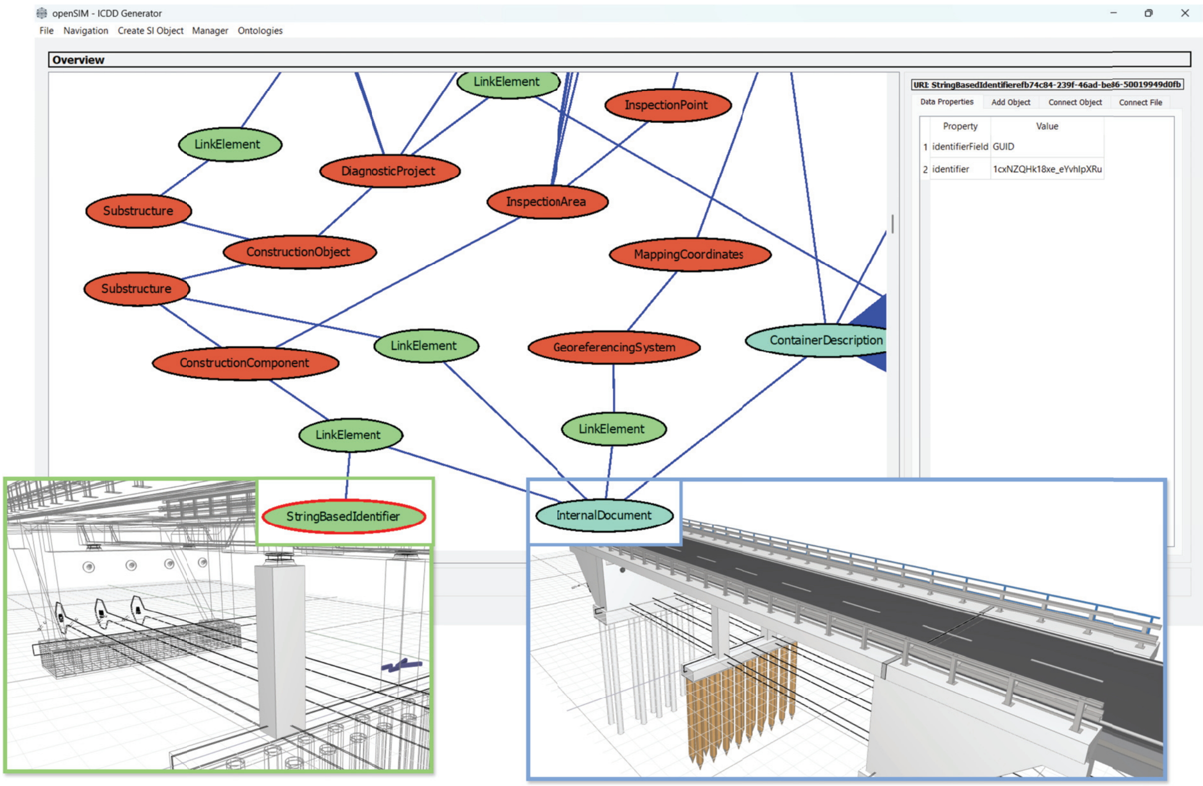

Figure 5: Overview of payload triples visualised in the ICDD application.

Listing 1: Representation of an inspection point in Turtle format.

```
1 @prefix sio: <http://www.openSIM.org/0.10.24/SIO#> .
2 @prefix prj: <http://BUW-ITD.org/openSIM/Diagnostics#> .
3 @prefix idx: <http://BUW-ITD.org/openSIM/index#> .
4 @prefix ls: <https://standards.iso.org/iso/21597/-1/ed-1/en/Linkset#> .
5 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
6 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
7
8 prj:InspectionPoint5436e0e7-b292-4f54-98f7-e08cedc2a3f1
9     a sio:InspectionPoint ;
10    sio:Name "1.3"^^xsd:string ;
11    sio:X "-2.54979361828466"^^xsd:string ;
12    sio:Y "8.78249999999999"^^xsd:string ;
13    sio:Z "5.20020638171526"^^xsd:string ;
14    sio:inGeoreferencingSystem prj:GeoreferencingSystemc668bbce-4c11-4aff-8d9b-3
        f2f4f923350 ;
15    sio:inInspectionArea prj:InspectionArea3b5cd8c3-05a9-4589-991e-e4afcddb59 ;
16    sio:hasLinkElement [
17        a ls:LinkElement ;
18        ls:hasDocument idx:InternalDocumentffeb0194-02b6-47b0-b64c-64a5f4b02a2f
19    ] .
```

An excerpt from the triplestore stored in the Payload triples folder of the ICDD container is presented in Listing 1. Lines 10 and 11 define an inspection point named "1.3." The subsequent lines (11 to 15) describe the position of the inspection point within a defined georeferencing system, which is mapped to the coordinate system of the IFC model of the bridge. The mapping ensures precise localization within the structure. Line 16 specifies the inspection area, in which the inspection point is situated. Lines 17 to 21 demonstrate the definition of a link to a document, with line 19 referencing a document indexed in the index.rdf file located within the ICDD container. As shown in the sample output later in the Listing 2, line 26, the linked document is an image file in PNG format.

Listing 2: Result of Querying for an Inspection Point named 1.3 in OData-JSON format.


```json
{
  "@odata.context": "$metadata#InspectionPoint",
  "value": [
    {
      "name": "1.3",
      "comment": [],
      "description": [],
      "x": "-2.54979361828466",
      "y": "8.78249999999999",
      "z": "5.20020638171526",
      "hasDirection": null,
      "inInspectionArea": [
        "name": "UB1",
        "id": "InspectionArea3b5cd8c3-05a9-4589-991e-e4afcddb59",
        "type": "InspectionArea"
      ]
      "inGeoreferencingSystem": {
        "name": "IFC Local System",
        "id": "GeoreferencingSystemc668bbce-4c11-4aff-8d9b-3f2f4f92330",
        "type": "GeoRefSystem"
      },
      "hasLinkElement": [
        {
          "name": "inspectionpoint.png",
          "filename": "images\inspectionpoint.png",
          "filetype": ".png",
          "identifier": null,
          "identifierField": null,
          "id": "LinkElement3c1131aa-b48b-4dcd-b20f-7fcd60109da0",
          "type": "LinkElement"
        }
      ],
      "id": "InspectionPoint5436e0e7-b292-4f54-98f7-e08cedc2a3f1",
      "type": "InspectionPoint"
    }
  ]
}
```


Once the ICDD container is created, it is uploaded to a server where it is stored as a cache. Container-specific information can be retrieved by using a query through the API. In this example, the query requests information about the inspection point named "1.3". The resulting output, shown in Listing 2, provides the previously described information from Listing 1, along with additional object and data properties.

6. Conclusions & Discussion

This study presented a structured approach for applying the Linked Data framework in the field of structural diagnostics. Utilizing the ICDD Standard, which follows a traditional document-based structure, a workflow was established to enable querying of information from ICDD containers. Beyond the ontologies defined in ISO 21597, an additional ontology, the SIO, was developed and incorporated to represent concepts and elements related to structural diagnostics. The development of SIO involved an analysis of common diagnostic methods to ensure their inclusion within the ontology. Its modular design facilitates the integration of additional methods as required.

It is worth noting that existing ontologies have limited applicability in practical scenarios due to their reliance on the RDFS and the OWL, which allow for a high degree of interpretive flexibility in information modeling. This flexibility results from the open world assumption and the non-unique name assumption. Incorporating constraints that align with the closed world assumption could improve precision by restricting permissible information and enabling formal validations. Schema languages such as Shapes Constraint Language (SHACL), Shape Expressions (ShEx), and SPARQL Inferencing Notation (SPIN) provide mechanisms for implementing various constraints.

The ICDD authoring application described in Section 4.1 and illustrated in Figure 5 has been shown to offer functionalities for creating an ICDD container and enriching it with data. Ensuring the ICDD consistency with both the ISO 21597 standard and the underlying ontologies, the developed tool was proven to provide a robust platform for managing graph-based inspection data. However, the usability of the ICDD application is limited and requires prior knowledge of linked data. Future research should focus on developing user-friendly tools to simplify container enrichment processes, thereby potentially increasing the acceptance and adoption of Linked Data approaches within the AECO industry.

A REST-API has been developed for extracting, processing, and delivering data from ICDD containers. It has been proven that the implemented API facilitates standardized data provisioning, while enabling targeted access to the RDF data models within ICDD containers. By integrating and instantiating the SIO, the API architecture has been tested for extending ICDD containers beyond individual projects, supporting long-term usability and accessibility, preserving data for future applications, and providing scalability and reliability.

However, the simultaneous processing of multiple containers presents challenges, including dependency management, data consistency, and conflict resolution. Conflicts may occur, when containers comprise redundant resources or objects with identical URIs, potentially resulting in namespace collisions or contradictory information. Additionally, query behavior must be defined to enable both isolated and cross-container data access effectively. Effectively addressing these challenges is essential for advancing and scaling the API architecture. Solutions should ensure a balance between robust conflict resolution mechanisms and efficient data management and querying across multiple containers.

Acknowledgments

This study is undertaken as part of the OpenSIM research project, supported by grant number 19F2217F through the mFUND initiative of the German Federal Ministry for Digital and Transport (BMDV). The contents of this paper,including all statements and findings, are solely the responsibility of the authors and do not represent the views of the funding institution. The authors wish to express their appreciation to the project partners of OpenSIM.

Declaration on Generative AI

The authors declare that generative AI tools, namely Google's Gemini 2.5 Pro and Perplexity AI (Pro subscription), were employed exclusively for linguistic refinement and textual paraphrasing throughout the manuscript preparation period from January 2025 to May 2025. It is worth noting that Perplexity AI (Pro subscription) has primarily relied on several advanced LLMs (including GPT-5, Claude Sonnet 4.5, Gemini 2.5 Pro, and Sonar Large) and has automatically selected the optimal model for the specific task, such as paraphrasing or proofreading academic content. Selected passages underwent revision using these tools to verify grammatical correctness, enhance sentence construction, eliminate typographical inconsistencies, and ensure compliance with standard academic English conventions. Notably, generative AI was not utilized for the generation of scientific content and was not employed in the production of figures, diagrams, or code implementations presented in this work.

References

[1] Deutsches Institut für Normung e.V., Engineering structures in connection with roads - inspection and test (DIN 1076), 1999. URL: https://www.din.de/de/wdc-beuth:din21:23474630.

[2] ISO 21597-1, Information container for linked document delivery: Exchange specification - Part 1: Container, 1 ed., Standard, International Organization for Standardization, Geneva, Switzerland, 2020. URL: https://www.iso.org/standard/74389.html.

[3] ISO 21597-2, Information container for linked document delivery: Exchange specification - Part 2: Link types, 1 ed., Standard, International Organization for Standardization, Geneva, Switzerland, 2020. URL: https://www.iso.org/standard/74390.html.

[4] RDF Primer, 2004. URL: https://www.w3.org/TR/rdf-primer/.

[5] N. Guarino, D. Oberle, S. Staab, What Is an Ontology?, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 1–17. doi:10.1007/978-3-540-92673-3_0.

[6] R. Guha, D. Brickley, RDF Schema 1.1, 2014. URL: https://www.w3.org/TR/rdf-schema/.

[7] D. L. McGuinness, F. van Harmelen, OWL Web Ontology Language Overview, 2004. URL: https://www.w3.org/TR/owl-features/.

[8] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor, Cypher: An evolving query language for property graphs, in: Proceedings of the 2018 international conference on management of data, 2018, pp. 1433–1445.

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

[10] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M. B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick: Towards a unified metadata schema for buildings, in: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, BuildSys '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 41–50. doi:10.1145/2993422.2993577.

[11] K. Hammar, E. O. Wallin, P. Karlberg, D. Hälleberg, The realestatecore ontology, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham, 2019, pp. 130–145. doi:10.1007/978-3-030-30796-7_9.

[12] D. Reynolds, The Organization Ontology, 2014. URL: https://www.w3.org/TR/vocab-org/.

[13] M. Bennett, The financial industry business ontology: Best practice for big data, Journal of Banking Regulation 14 (2013) 255–268. doi:10.1057/jbr.2013.13.

[14] A. Haller, K. Janowicz, S. Cox, D. L. Phuoc, K. Taylor, M. Lefrançois, The Semantic Sensor Network Ontology (2017). URL: https://www.w3.org/TR/vocab-ssn/.

[15] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc, M. Lefrançois, SOSA: A lightweight ontology for sensors, observations, samples, and actuators, Journal of Web Semantics 56 (2019) 1–10. doi:10.1016/j.websem.2018.06.003.

[16] J. Beetz, J. van Leeuwen, B. de Vries, Ifcowl: A case of transforming express schemas into ontologies, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23 (2009) 89–101. doi:10.1017/S0890060409000122.

[17] V. Borges, N. Q. de Oliveira, M. L. M. Campos, A Multi-level Ontology-based Approach for Descriptors of Catalogued Resources, in: Proceedings of the 15th Seminar on Ontology Research in Brazil (ONTOBRAS), 2022, pp. 46–59. URL: https://ceur-ws.org/Vol-3346/Paper4.pdf.

[18] S. Törmä, P. Toivola, M. Kiviniemi, P. Puntila, M. Lampi, T. Mätäsniemi, Ontology-based sharing of structural health monitoring data, in: 20th Congress of IABSE, New York City, 2019, pp. 2214–2221.

[19] R. Li, T. Mo, J. Yang, S. Jiang, T. Li, Y. Liu, Ontologies-based domain knowledge modeling and heterogeneous sensor data integration for bridge health monitoring systems, IEEE Transactions on Industrial Informatics 17 (2020) 321–332.

[20] G. P. Tsialiamanis, D. J. Wagg, I. Antoniadou, K. Worden, An ontological approach to structural health monitoring, in: Topics in Modal Analysis & Testing, Volume 8: Proceedings of the 38th

IMAC, A Conference and Exposition on Structural Dynamics 2020, Springer, 2021, pp. 51–59.
[21] F. Bahreini, A. Hammad, Developing an ontology for concrete surface defects to enhance inspection, diagnosis and repair information modeling, Infrastructures 9 (2024). URL: https://www.mdpi.com/2412-3811/9/12/220. doi:10.3390/infrastructures9120220.
[22] B. Moreno Torres, C. Völker, S. M. Nagel, T. Hanke, S. Kruschwitz, An ontology-based approach to enable data-driven research in the field of ndt in civil engineering, Remote Sensing 13 (2021). URL: https://www.mdpi.com/2072-4292/13/12/2426. doi:10.3390/rs13122426.
[23] C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, X. Wang, Ontological knowledge base for concrete bridge rehabilitation project management, Automation in Construction 121 (2021) 103428. doi:https://doi.org/10.1016/j.autcon.2020.103428.
[24] Y. Zheng, S. Törmä, O. Seppänen, A shared ontology suite for digital construction workflow, Automation in Construction 132 (2021). URL: https://digitalconstruction.github.io/v/0.3/index.html. doi:https://doi.org/10.1016/j.autcon.2021.103930.
[25] T. Heath, C. Bizer, Principles of Linked Data, Springer International Publishing, 2011, pp. 7–27. doi:10.1007/978-3-031-79432-2_2.
[26] Y. Zhang, Implementing information container to link multi-models for semantic digital twins, Technische Universität Hamburg, Institut für Digitales und Autonomes Bauen, 2024. doi:10.15480/882.13486.
[27] P. Hagedorn, Implementation of a validation framework for the information container for data drop, in: Proceedings 30. Forum Bauinformatik, At Bauhaus Universität Weimar, 2018. doi:10.25643/bauhaus-universitaet.
[28] L. Liu, P. Hagedorn, M. König, Definition of a container-based machine-readable IDM integrating level of information needs, in: Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference, volume 4 of *Computing in Construction*, European Council on Computing in Construction, Heraklion, Greece, 2023. doi:10.35490/EC3.2023.221.
[29] N. N. Al-Sadoon, R. Scherer, K. Menzal, From static to dynamic information containers, in: Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference, volume 4 of *Computing in Construction*, European Council on Computing in Construction, Heraklion, Greece, 2023. doi:10.35490/EC3.2023.243.
[30] M. Mirboland, P.-C. Schuler, M. Artus, C. Koch, Integrating inspection data from non-destructive tests on in-service infrastructure into openbim data models, in: Proceedings of the 20th International Conference on Computing in Civil and Building Engineering (ICCCBE)- Volume 3: Construction Management, 2024, p. 410–420. doi:https://doi.org/10.1007/978-3-031-84224-5_33.
[31] P.-C. Schuler, M. Mirboland, C. Voigt, C. Koch, Using linked data containers for bim based structural inspection workflows, in: Proceedings of the 20th International Conference on Computing in Civil and Building Engineering (ICCCBE)- Volume 3: Construction Management, 2024, p. 398–409. doi:https://doi.org/10.1007/978-3-031-84224-5_32.