RWTH AACHEN UNIVERSITY

Linking and Managing Heterogeneous Data using Information Containers

Leveraging Linked Data for BIM Stage 3 CDEs

Madhumitha Senthilvel

Linking and Managing Heterogeneous Data using Information Containers

Leveraging Linked Data for BIM Stage 3 CDEs

Von der Fakultät für Architektur der
Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von:
Madhumitha Senthilvel
geboren in Chennai, Indien

Berichtende:
Univ.-Prof. Dr. Jakob Beetz
Univ.-Prof. Dr.-Ing. Christian Raabe

Tag der mündliche Prüfung: 19.08.2024

Disclaimer

This thesis is a product of the PhD pursued by the author at the Design Computation Chair, Faculty of Architecture, RWTH Aachen University. All opinions and research, with the exception of citations, are the opinions of the author.

Copyright

Changelog

v1.0	2023-08-11	First full draft of thesis submitted
v1.1	2023-09-15	Second draft of thesis submitted
v1.2	2024-03-18	Final version of thesis submitted to the Faculty of Architecture, RWTH Aachen University
v1.3	2025-01-28	Final version of thesis submitted to Universitätsbibliothek, RWTH Aachen University

Declaration of own work

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where stated otherwise by reference or acknowledgement, the work presented is entirely my own.

Hannover, March2024
Madhumitha Senthilvel

Acknowledgements

This thesis would not have been possible without the support of numerous people with whom I have crossed paths over the last few years. Although it was not a journey that I had anticipated that I would undertake, it has been a truly wonderful self-exploratory endeavor, filled with countless learnings.

First, I would like to extend my heartfelt thanks to Prof. Dr. Jakob Beetz for his mentorship over the years and the countless opportunities for exposure to different experiences and collaborations, all of which fostered a genuine interest in exploring the world of information management. His relentless support, especially for my scholarship application, is very much appreciated. Next, I extend my thanks to the Deutscher Akademischer Austauschdienst (DAAD) for funding my research, supporting research stays, and facilitating exposure to conferences and its network. I extend my gratitude to Prof. Dr.-Ing. Christian Raabe for accepting the role of the second referent and for our engaging discussions.

I also would like to extend my gratitude to my colleagues from the BIM4Ren projects, which proved to be a sandbox for exploring and learning new ideas and experimenting with them. Dr. Pieter Pauwels deserves a special mention for accommodating me on a brief, but productive research stay at TU Eindhoven and for his guidance on my thesis. I also thank the TandemDok program for connecting me with Dr. Magdalena Tarkiwicz, whose encouragement and support during the last stretch of this thesis and my career path were invaluable.

I have had the pleasure of meeting and connecting with some brilliant people from all walks of life, whose encounters have resulted in enriching experiences: Noemi Kremer, Oliver Schulz, Jyrki Oraskari, Ying Ying Zhang, Jeroen Werbrouck, Alex Donkers, Patrick, Lisa, Thomas, Amrita and numerous others - perhaps unmentioned here, yet not forgotten. A shout out to Anna Wagner, for encouraging me in my first shaky steps in the Linked Data world. And of course Anja, for ever helping navigate the Deutsche Bürokratie. A big thank you also to Anchal and Vairav for all the late-night calls, blood, sweat, and tears over academics and work. A special mention to my colleagues at TenneT for their understanding and empathy. I am also grateful to have been part of various Linked Data and BIM communities which fostered an environment of open exchanges, networking and interactions with peers. My perspectives were greatly enhanced and moulded by these encounters.

And finally, a heartfelt thank you to my wonderfully supportive parents, Uma and Senthil Vel, without whom I doubt I would have had the endurance to pursue this

path. Throughout the 7000 km between us, through the COVID pandemic, countless curfews, travel restrictions, and personal life events, they have been my inspiration and strength.

I hope that you, the reader, enjoy reading this thesis as much as I enjoyed writing it.

- Madhumitha
Hannover, 2024

Preface

The construction industry is widely known to be extremely fragmented: it involves multiparty, multidisciplinary teams collaborating on the creation, and usage of fragmented, yet interconnected data. These data are generated, stored and processed in diverse tools. As a result, interoperability issues between different tools have been a focal point of research for more than three decades. The introduction of BIM paved the way for improved information management and enhanced information interoperability. Despite significant progress, construction projects continue to face consequential challenges in effectively managing the diverse range of information. In particular, brownfield projects such as the ones for energy retrofitting and renovation deal with an extremely complex mix of legacy and new data that are not always structured.

Information interoperability and management is one of the decisive factors which influence the success of these projects. The availability of the right information, at the right time significantly improves overall collaboration and effective project management. Despite extraordinary advances in data capture and availability in the past two decades, projects struggle with utilising them efficiently for decision-making. In typical projects, infrastructure components are represented in various formats, with varying levels of information, depending on the use case for which they were created. While advanced BIM-based paradigms such as Common Data Environments have proved advantageous, there is a significant gap in managing interconnected data in these environments. Major challenges here include gaps in consensus of standardised schema for non-IFC data and lack of standardised vendor-neutral representation of meta-data and interconnected information.

In this research, three key issues in managing interconnected information are addressed. The first objective is to provide a formal representation of informal relationships within heterogeneous data in construction projects by utilizing Linked Data approaches. This facilitates easy understanding and retrieval of existing knowledge from various, disparate sources. Furthermore, this research proposes the use of Information Containers, which serve as the central repository for all formalized knowledge throughout a project's lifecycle in a Common Data Environment. These containers promote efficient organization and management of interconnected data, making it easily accessible to all stakeholders involved in the project. The Information Containers are designed with a CDE perspective, focusing on the functional elements of the containers and their stored data. Lastly, to maintain data quality and consistency, the proposed approach incorporates SHACL rule language for validation. This ensures that all data and interlinked relationships conform to predefined standards and adhere to integrity constraints, thus

enhancing the overall reliability of the project's information. By addressing these three key issues, this research aims to improve the management of interconnected information in AEC projects.

Through the adaption of existing models for Information Containers, and their integration with the functional aspects of CDEs, it is demostrated that heterogeneous interconnected data can be efficiently managed in all phases in the construction industry. Furthermore, by leveraging linked data principles, the integration of the current document-centric practice with the semantic, data-centric practice is demonstrated. The results of this work can potentially push forward the developments of Common Data Environments beyond their current one-dimensional interoperability. Seamless integration and exchange of data, regardless of how the data is structured, has an enormous potential for practical use cases from projects.

Einleitung

Das Baugewerbe ist bekanntlich extrem fragmentiert: Es umfasst mehrere Parteien und multidisziplinäre Teams, die bei der Erstellung und Nutzung fragmentierter, aber miteinander verbundener Daten zusammenarbeiten. Diese Daten werden mit unterschiedlichen Werkzeugen erzeugt, gespeichert und verarbeitet. Daher sind die Herausforderungen der Interoperabilität zwischen den verschiedenen Werkzeugen seit mehr als drei Jahrzehnten ein Schwerpunkt der Forschung. Die Entwicklung von BIM ermöglichte ein verbessertes Informationsmanagement und eine größere Interoperabilität der Informationen. Trotz signifikanter Fortschritte stehen Bauprojekte weiterhin vor großen Schwierigkeiten bei der effektiven Steuerung des vielfältigen Datenmaterials. Insbesondere Altbauprojekte wie die energetische Sanierung und Renovierung haben mit einer äußerst komplexen Mischung aus alten und neuen Daten zu tun, die nicht immer strukturiert sind.

Die Interoperabilität und das Management von Informationen ist einer der entscheidenden Faktoren, die den Erfolg dieser Projekte beeinflussen. Die Verfügbarkeit der richtigen Informationen zum richtigen Zeitpunkt verbessert die allgemeine Zusammenarbeit und das effektive Projektmanagement erheblich. Trotz außerordentlicher Fortschritte bei der Datensammlung und -verfügbarkeit in den letzten zwei Jahrzehnten ist es bei Projekten schwierig, diese effizient für die Decision-making zu nutzen. In typischen Projekten werden Infrastrukturkomponenten in verschiedenen Formaten und mit unterschiedlichem Informationsgehalt dargestellt, je nach dem Anwendungsbereich, für den sie erstellt wurden. Obwohl sich fortschrittliche BIM-basierte Paradigmen wie Common Data Environments als vorteilhaft erwiesen haben, gibt es eine deutliche Lücke bei der Verwaltung miteinander verbundener Daten in diesen Umgebungen. Zu den größten Herausforderungen in diesem Bereich gehören der mangelnde Konsens über standardisierte Schemata für Non-IFC-Daten und das Fehlen einer standardisierten, vendor-neutral Darstellung von Metadaten und vernetzten Informationen.

In dieser Untersuchung werden drei zentrale Fragen der Verwaltung verknüpfter Informationen behandelt. Das erste Ziel besteht darin, eine formale Darstellung informeller Beziehungen innerhalb heterogener Daten in Bauprojekten durch die Verwendung von Linked-Data-Ansätzen zu ermöglichen. Dies erleichtert das Verständnis und den Abruf von vorhandenem Wissen aus verschiedenen, disparaten Quellen. Außerdem wird die Verwendung von Informations Containers als Kernstück einer gemeinsamen Datenumgebung vorgeschlagen, in der das gesamte formalisierte Wissen über den Projektablauf gesammelt wird. Diese Container unterstützen die effiziente Organisation und Verwaltung miteinander verbundener Daten und machen sie für alle am Pro-

jekt Beteiligten leicht zugänglich. Die Informationscontainer werden aus einer CDE-Perspektive entwickelt, mit Fokus auf die funktionalen Elemente der Container und die gespeicherten Daten. Um die Datenqualität und -konsistenz zu gewährleisten, verwendet der vorgeschlagene Approach die SHACL-Regelsprache zur Validierung. Damit ist sichergestellt, dass alle Daten und verknüpften Beziehungen den vordefinierten Standards entsprechen und die Integritätsbedingungen einhalten, was die allgemeine Zuverlässigkeit der Projektinformationen erhöht. Mit diesen drei Schlüsselfragen zielt diese Forschungsarbeit darauf ab, das Management von verknüpften Informationen in Bauprojekten zu verbessern.

Die Adaption der bestehenden Modelle für Informations Container und deren Integration mit den funktionalen Aspekten von CDEs zeigt, dass heterogene, verknüpfte Daten in allen Phasen der Bauindustrie effizient verwaltet werden können. Außerdem wird durch die Nutzung von Linked-Data-Prinzipien die Integration der derzeitigen dokumentenzentrierten Praxis mit der semantischen, datenzentrierten Praxis demonstriert. Die Ergebnisse dieser Arbeit können die Entwicklung gemeinsamer Datenumgebungen über ihre derzeitige eindimensionale Interoperabilität hinaus potenziell vorantreiben. Die drahtlose Integration und der Austausch von Daten, unabhängig davon, wie die Daten strukturiert sind, hat ein enormes Potenzial für praktische Projektanwendungen.

Abbreviations

AEC Architecture, Engineering, and Construction

OAuth2.0 Open Authorization version 2.0

API Application Programming Interface

BEO Building Element Ontology

BIM Building Information Model/Modelling

BIM4Ren BIM for Renovation

BOT Building Topology Ontology

BoQ Bill of Quantities

BPMN Business Process Model and Notation

COINS Construction Objects and the INtegration of processes and Systems

CWA Closed World Assumption

CDE Common Data Environment

CRUD Create, Read, Update, Delete

EIR Exchange Information Requirements

GAEB Gemeinsamer Ausschuss Elektronik im Bauwesen

gbXML Green Building XML

HTTP Hypertext Transfer Protocol

IC Information Container

ICDD Information Containers for linked Document Delivery

IFC Industry Foundation Classes

IRI Internationalized Resource Identifier

JSON-RPC JSON Remote Procedure Call

LD Linked Data

LBD Linked Building Data

LDP Linked Data Platform

LOI Level of Information

MEP Mechanical, Electrical, Plumbing

MMC Multi-Model Container

MSA Micro Services Architecture

NLP Natural Language Processing

OCL Object Constraint Language

OCR Optical Character Recognition

OMG Ontology for Managing Geometry

OSAP One Stop Access Platform

OWA Open World Assumption

OWL Web Ontology Language

QUDT Quantities, Units, Dimensions, and Types

RDF Resource Description Framework

RDFS Resource Description Framework Schema

RFI Request for Information

REST Representational State Transfer

RML RDF Mapping Language

SKOS Simple Knowledge Organization System

SOAP Simple Object Access Protocol

SSoT Single Source of Truth

SW Semantic Web

HVAC Heating, Ventilation, Air-Conditioning

SPIN SPARQL Inference Notation

SHACL SHape Constraint Language

ShEx Shape Expressions

SPARQL SPARQL Protocol and RDF Query Language

SSOT Single Source

SSO Single Sign On

SVoT Single Viewpoint of Truth

SWRL Semantic Web Rule Language

UML Unified Modelling Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

W3C World Wide Consortium

WKT Well-known Text

WWW World Wide Web

XSD XML Schema Definition

Contents

Research Outputs

Research Publications

First author publications in Conference proceedings:

- **Senthilvel, M.**, and Beetz, J. (2020). A visual programming approach for validating linked building data. *Proceedings of EG-ICE 2020. International Workshop on Intelligent Computing in Engineering*, Technische Universität Berlin. https://publications.rwth-aachen.de/record/795561/files/795561.pdf

- **Senthilvel, M.**, Oraskari, J. and Beetz, J. (2020). Common Data Environments for the Information Container for linked Document Delivery. *In Proceedings of the 8th Linked Data in Architecture and Construction Workshop-LDAC*, 132-145. https://ceur-ws.org/Vol-2636/10paper.pdf

- **Senthilvel, M.** and Beetz, J. (2021). Conceptualizing Decentralized Information Containers for Common Data Environments using Linked Data. *In Proc. of the Conference CIB W78* (Vol. 2021, pp. 11-15).https://publications.rwth-aachen.de/record/834345

- **Senthilvel, M.**, Oraskari, J. and Beetz, J. (2021). Implementing Information Container for linked Document Delivery (ICDD) as a micro-service. *Proceedings of EG-ICE 2021. International Workshop on Intelligent Computing in Engineering*, 66-72. Universitätsverlag der TU Berlin. https://publications.rwth-aachen.de/record/826311

Collaborative publications in Conference proceedings:

- Werbrouck, J., **Senthilvel, M.**, Beetz, J., Bourreau, P. and Van Berlo, L. (2019). Semantic query languages for knowledge-based web services in a construction context. *Proceedings of the 26th International Workshop on Intelligent Computing in Engineering, EG-ICE 2019, 2394.* https://ceur-ws.org/Vol-2394/paper03.pdf

- Werbrouck, J., **Senthilvel, M.**, Beetz, J., and Pauwels, P. (2019). Querying heterogeneous linked building datasets with context-expanded GraphQL queries. *Proceedings of the 7th Linked Data in Architecture and Construction Workshop, LDAC 2019, 2389*, 21–34. https://ceur-ws.org/Vol-2389/02paper.pdf

- Werbrouck, J., **Senthilvel, M.**, Beetz, J. and Pauwels, P. (2019), September. A checking approach for distributed building data. *Proceedings of 31st forum*

bauinformatik, Berlin: Universitätsverlag der TU Berlin 173-181. https://d-nb.info/1227054572/34

- Bourreau, P., Charbel, N., Werbrouck, J., **Senthilvel, M.**, Pauwels, P. and Beetz, J. (2020). Multiple inheritance for a modular BIM. *Le BIM et l'évolution des pratiques: Ingénierie et architecture, enseignement et recherche*, 63-82. https://publications.rwth-aachen.de/record/812201
- Oraskari, J., **Senthilvel, M.** and Beetz, J.(2021). SHACL is for LBD what mvdXML is for IFC. *Proceedings of the CIB W78 Conference* 2021. 11-19. https://itc.scix.net/paper/w78-2021-paper-069
- Hagedorn, P., **Senthilvel, M.**, Schevers, H. and Verhelst, L.B., 2023. Towards usable ICDD containers for ontology-driven data linking and link validation. In Proc. of the 11th Linked Data in Architecture and Construction Workshop (LDAC2023) CEUR Workshop Proceedings, Matera, Italy (pp. 71 - 84) https://ceur-ws.org/Vol-3633/paper3.pdf

Contribution to Book Chapter:

- Werbrouck, J., **Senthilvel, M.** and Rasmussen, M.H. (2022). Federated data storage for the AEC industry. *In Buildings and Semantics*. CRC Press. pp. 139 - 164. CRC Press. https://doi.org/10.1201/9781003204381

Ontologies and Mappings

- BuildLinks: An Ontology for capturing metadata for links within AEC projects - https://github.com/SemanticHub/BuildLinks
- ContainerStates: An Ontology for defining container states as per ISO 19650 part 1 - https://github.com/SemanticHub/CS-Ontology
- Mappings between ICDD and BuildLinks https://github.com/SemanticHub/CDEOntologyMappings

Datasets & Prototypes

- SHACL shapes repository: List of all SHACL shapes developed for conformance checks in this thesis https://github.com/SemanticHub/SHACLDB
- Validator bot: A microservice which performs SHACL-based data conformance checks and displays output in a 3D model, and also a SHACL validation report https://github.com/SemanticHub/EpicSHACLVisualizer

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Information & its management in AEC

Construction projects are notorious for being resource, time, and cost-intensive. A typical construction project handles immense blocks of heterogeneous information (i.e. each piece of data is structured differently), flowing in loops and linearly through various project participants. Though estimates for the quantum of information vary widely due to dependencies on project scale, scope and involved participants, conservative estimates from literature range from 6000-30000 documents for just one phase for a project (Craig & Sommerville, 2006; Schroepfer, 2006). When scaled up for the entire lifecycle of a project, these pieces of data can span thousands of documents, models, drawings, images, point clouds, etc. These pieces of data are inherently heterogeneous, i.e. each piece of information can be created using different sources and stored in various formats. A 2022 report initiated by Autodesk quantifies that unusable data or the mismanagement of it cost projects worldwide, a whopping 1.84 Trillion USD (Autodesk, 2022).

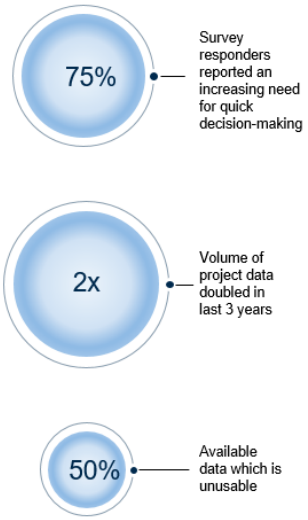

Figure 1.1: Explosion of data in AEC projects (Autodesk, 2022)

Besides the enormous quantum of information generated within a project, these information are frequently exchanged, versioned, and tracked across diverse project participants. During the course of a project, these participants interact in complex ways, making it challenging to establish a Single Source of Truth[1] at any point in time. The construction industry has long drawn inspiration, adoption and adaption of concepts and approaches like lean approach, digital modelling, resource planning, etc. Yet, despite these similarities, fundamental differences in the participants, information flow

[1] A Single Source (SSOT) aggregates all the data of a project into a single location, which is accessible by all members of the project team.

directions, organisational culture etc., make the construction industry more complex[2].

[2]The notion of BIM was inspired by Computer-Integrated Manufacturing, an approach in the manufacturing sector, where information is modelled digitally prior to the production of the physical component

Traditionally, the AEC domain has heavily borrowed concepts and approaches from other domains for both information and project management (Crowley, 1998; Delgado Camacho et al., 2018). However, AEC projects are inherently unique from their counterparts in other domains, like automobile manufacturing, chip manufacturing, software development, etc. For instance, in the case of automobile manufacturing, the broad product lifecycle are:

- Plan and Define program,
- Product Design and Development,
- Process Design and Development,
- Product and Process Validation,
- Production Launch,
- Feedback Assessment and Corrective Action

A typical construction project consists of the following phases which are broadly structured as per ISO 19650:

- Assessment and Need
- Invitation to tender
- Tender response
- Appointment
- Mobilisation
- Collaborative production of information
  - Planning
  - Design
  - Procurement
  - Manufacture/Construction
  - Comissioning
- Information model delivery - Handover
- Project close-out

Although these phases overlap significantly with the construction project phases, the key difference lies in the iterative and highly variable nature of construction projects, where data can change even during the execution phase or

at the point of handover to the operation and maintenance phase. This indicates that in construction projects in the AEC domain, data is highly dynamic, with the potential for variation at any given point by any participant involved.

Figure 1.2 illustrates this highly dynamic nature of the AEC industry by comparing it to the manufacturing industry. Noticeably in the latter, heavy collaboration exists between teams/users within a company, while external collaboration is minimal. However, in the AEC industry, there are distinctly more external collaborators involved and the frequency of collaboration between both external and internal users is sporadic[3]. Combining the above two key features with the relatively long lifespan (ranging from years to decades) of construction projects, information management becomes crucial for a successful project completion and obtains far more complex characteristics than other industries.

[3]Historically, the AEC industry has long reported its complexity: the number of potential permutations and combinations for design, plan, construct each element in a building(Dubois & Gadde, 2002)

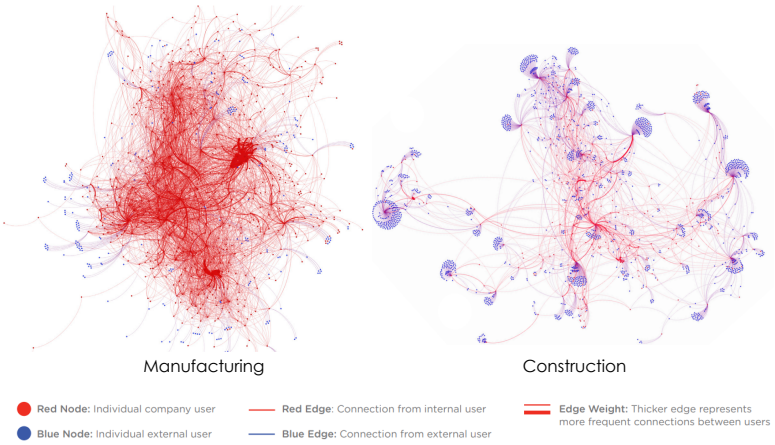

Figure 1.2: Comparing information exchanges between participants in the manufacturing industry and the construction industry (Box, 2014)

In the current age where "information is oil", companies' ability to be agile i.e. responding to new emerging information by leveraging its accessibility, its speed of updates, and resulting leverage value have become key winning points. Historically, information flows are heavily restricted between participants in organisational silos and internal hierarchies and have long impeded personnel from quick access, thus impacting effective decision-making. Designers, contractors, and asset owners were estimated to face significant challenges in managing data: with approximately 96% of captured data going unused (Autodesk, 2022). Furthermore, around 30% of the initial data created by project participants is lost by the time the project reaches the handover phase.

In brownfield projects like renovations, data management assumes a dual perspective: both existing 'as-is' information and new information generated during the execution of renovation (e.g. new redesigns, etc.) have to be considered. Until recently, the core challenge revolved around the collection and aggregation of heterogeneous information. In particular, renovation projects focussing on retrofitting existing building stock, have been reported to face challenges in being able to gather the as-is state of buildings (D'Oca et al., 2018; Volk et al., 2014).

Brownfield projects tend to not have any prior existing BIM data, and rely on the quality of captured/-collected data for further usage. This is especially true for old and historic buildings

With the advancements in technology, laser scans, photographs, sensor data, etc. have been effectively used to capture the accurate 'as-is' state of building stock. However, the problem of managing information not only remains but is exacerbated by the availability and complexity of 3D information which does not follow 2D conventions and paradigms.

The "holy grail" of successful project delivery critically relies on efficient data collaboration between project participants so that the data can be accessed on time. The advent of Building Information Model/Modelling (BIM) and digital information creation and management has eased and paved the way for traceable data and increased the availability of the information required to efficiently manage a project (Eastman, 2011). Continuous research in the past two decades has shown the benefits of using BIM and advanced digitalised information management (Bryde et al., 2013; Kumar et al., 2017; Migilinskas et al., 2013; Shen & Chua, 2011). BIM has brought about massive changes in tools, processes and working culture too. In particular, the tools for authoring and managing BIM-based information have embraced a data-centric approach, thus reinforcing the need to structure these data.

Nevertheless, BIM is not a one-stop solution, which can be blindly applied for all projects, as challenges in its adoption became apparent. These challenges can be broadly grouped into technical (and functional requirements) and non-technical (strategic) needs (Gu & London, 2010). Within the context of the technical needs, Common Data Environment (CDE)s were identified as solutions to facilitate collaboration and access to data. These environments are cloud-based repositories where multidisciplinary project participants can store data generated during the project, modify and update them, and share them with other project participants.

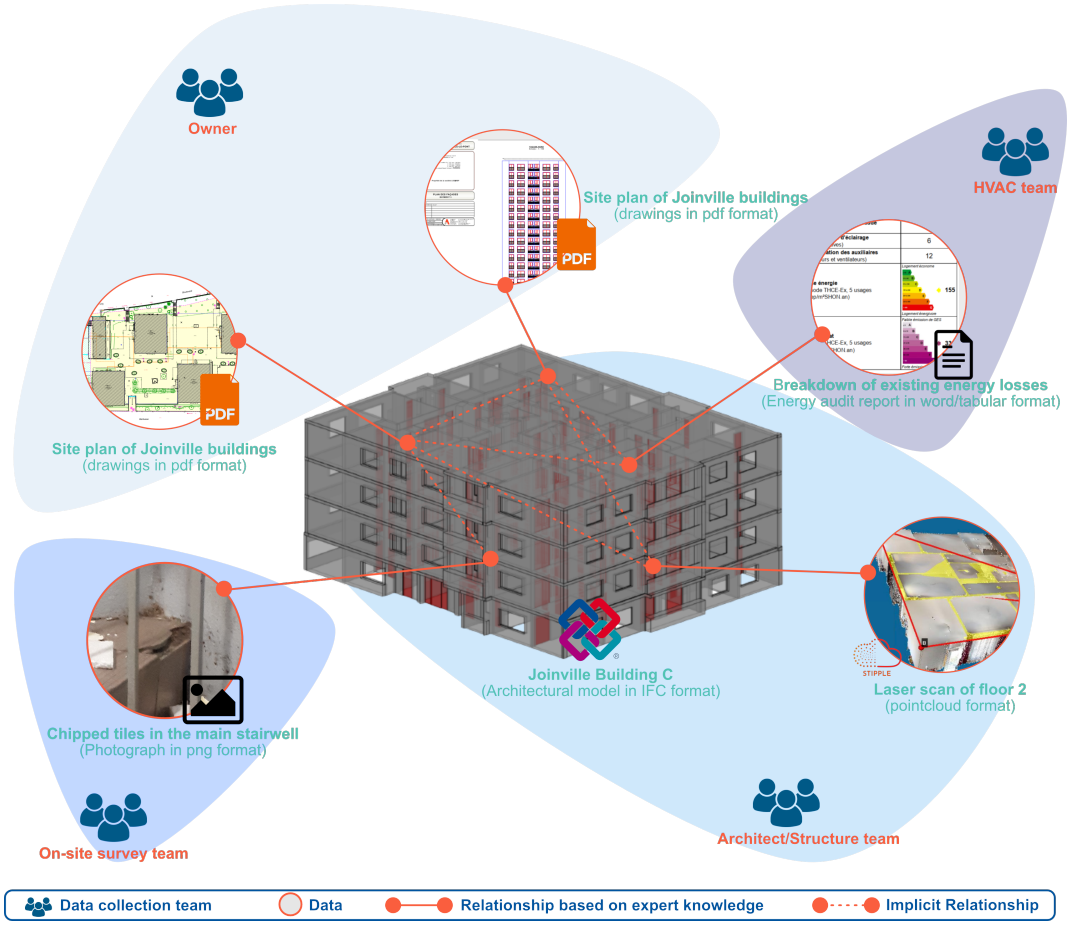

Figure 1.3: Inter-linked heterogeneous information in AEC projects

A CDE was defined to serve as a single source of truth that is accessible to all participants in the project involved and facilitates this access in real-time. However, the complexity of AEC data, participants, their interaction, diverse tools, modelling practices and eventual data usage has necessitated a re-evaluation of the current CDE approaches.

1.2 Challenges

AEC relies heavily on graphical and non-graphical data in various formats for daily communication between project participants (Caldas et al., 2002). Despite the advantages of BIM, this high heterogeneity of data generated and managed in AEC projects has been shown to affect efficiency, leading to coordination and communication problems, and eventually delays in decision making (Beck et al., 2020). Considering that most of these data are potentially interconnected with each other, they are often not explicitly recorded in a project. These relationships, which are critical for decision-

making in a project, are often informal domain knowledge which heavily relies on the interpretation and expertise of project participants. Due to the non-recorded and hence non-persistent nature of these relationships, they can be called '*informal links*'[4] These links are at perpetual risk of being lost if the concerned participant moves out of the project.

[4]These types of links are not explicitly codified or recorded in a project. They are often domain-specific knowledge or understanding, which the subject matter expert, e.g. a site engineer or an architect can comprehend, based on prior experience. An example of these kind of knowledge is shown in Fig. 1.3 where the implicit relationships (represented in dotted line) between different representations of the same building is shown.

Most of the BIM tools available in the market support the exchange of models in IFC schema. These include tools which are installed in the the local machine such as Autodesk's Revit, Bentley Microstation, Allplan, ArchiCAD etc. Advanced collaboration platforms also support object-based databases for storing these models (Eastman, 2011). For example solutions like BIM360, BIMCollab, TrimbleConnect do facilitate collaboration of different project participants and support federation of models. However, they struggle to meet the requirements for federation as defined in Stage 2 and 3 of the BIM maturity model (see Fig. 1.4). They also do not support federation of information without interoperability issues, query of sub-models, scalability of connected models to external tools (Bucher & Hall, 2020; Godager et al., 2021).

Bucher and Hall, 2020 makes a case for classifying these CDEs as 1-Dimensional - they facilitate data exchange and collaboration with tools from the providers' tool ecosystem, but do not support linkage to externally hosted information/platform. This results in CDEs with limited capabilities: they can store the data (i.e., files), but the relationship between them are not always storable, especially if they do not conform to the formats that the CDE ecosystem supports. Each of these tools can indeed use its own internal schema for modelling and representing data. At the point of data exchange, these models are transformed from one system (tool) to another, creating high discord between the tools and also the participants involved.

Although the obvious solution of sharing and linking these disparate but related models (instead of transforming them) has been frequently proposed (Dankers et al., 2014; Goedert & Meadati, 2008; Pocobelli et al., 2018; Sadeghi et al., 2019), current CDE solutions still rely on sharing data in proprietary formats, forcing participants to choose between tools. This interoperability issue becomes more complicated, as there are no concrete recommendations or implement solutions that can be used to create or manage these interlinked data in a CDE.

An example of this scenario is illustrated in Fig. 1.3 (based

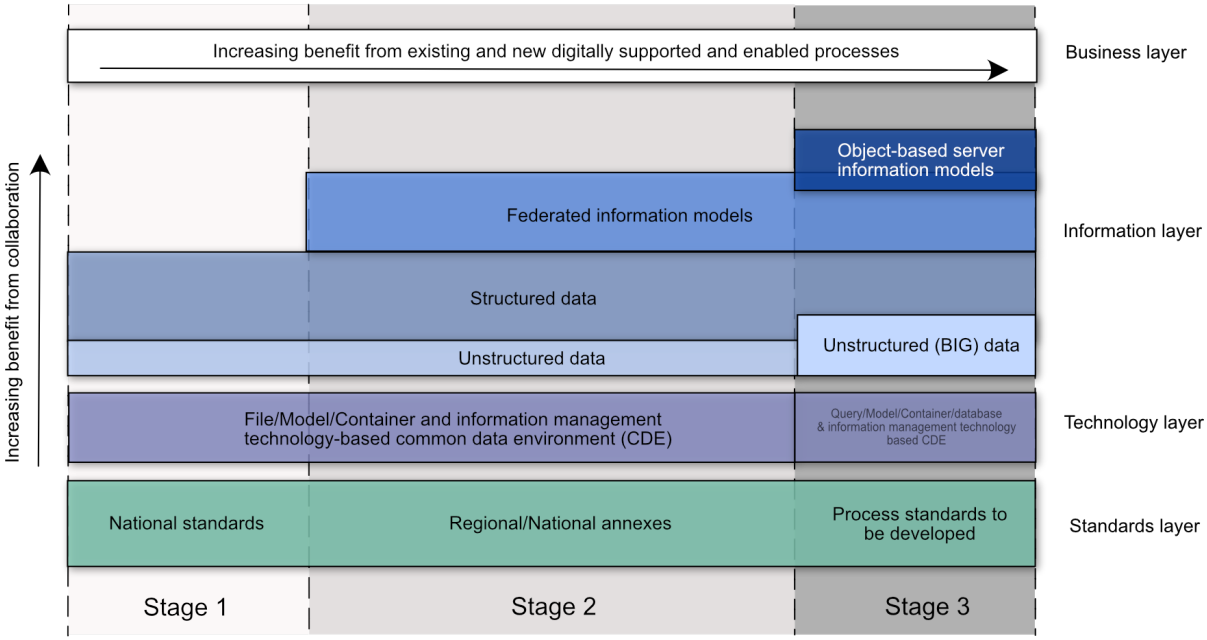

Figure 1.4: BIM Stages as defined by ISO 19650

on BIM4Ren case study, please refer to Chapter 3 for more details). It shows the interaction of different participants owning and exchanging different pieces of data.

Despite the relevance of interconnected heterogeneous data for the successful completion of projects, their integration into CDEs is a neglected field of research. One of the causes for this can be attributed to the lack of advanced CDE functionalities which use database-based information management, data federation, etc. However, a major factor is the considerable ambiguity that exists in the very definition of the functionalities and structures of the CDEs that have to handle the data and interlinking mentioned above.

Given the broad difference in the way AEC industry functions, neither BIM stage 1 nor stage 2 alone is not adequate to ensure better interoperability and collaboration. There is a massive need for ensuring all participants (both internal and external) are able to exchange and collaborate on information models regardless of their authoring software and data structures and these processes happen on the web, so that real-time data is accessible for project steering at all times.

However, it has been observed that the mechanisms to achieve this are vague according to ISO 19650[5] (explained in detail in Section 2.6.1) guidelines (Pauwels & McGlinn, 2022).

Numerous CDEs have been developed to meet the above

[5]ISO 19650 is an international standard which focuses on the management of information across the entire lifecycle of an asset.

needs: from shared local drives over networks to cloud-based storage solutions, all fulfilling the initially defined purpose of a CDE: i.e. a single source of truth. The BIM maturity model proposed originally by BS 1192: 2013, and subsequently expanded by ISO 19650 lays bare the necessities for pushing towards CDE-based data management (refer Fig. 1.4)[6]. However, despite the availability of traditional CDEs, the problems of collaboration and managing interlinked data in Stages 2 and 3 persist.

[6]While BS1192 originally proposed the BIM maturity levels (from 0 to 1), the superseding maturity stages proposed by ISO 19650 is considered in this thesis

1.3 Motivation

In the current context of increasing energy prices, geopolitical conflicts, a prolonged dependence on fossil fuels, and climate change, the focus on reduction in energy consumption has assumed greater importance in national and international policies. Buildings and infrastructure are currently the most dominant mass on the earth, followed by the mass of plants, the mass of human-made material and the mass of animals (Elhacham et al., 2020). Based on this context, the European Union considers improving the efficiency of the building sector to be a key contributor to reducing this mass and energy utilisation. Of these, renovation projects have the potential to contribute to reduction (Agency, 2020, 2021). The previous section identified that efficient information management systems are needed to manage renovation projects.

However, the sector's key problem is the lack of existing technological capability to efficiently convert informal representations of interlinked information into formal, machine-readable, and persistent structured data. Yalcinkaya and Singh, 2015 identified 12 main research areas in AEC based on patterns and trends in BIM research. Of these, *Information exchange and interoperability* ranked as the most researched theme. The topics under this theme included interoperability issues with software and with IFC schema, managing loss of information due to repeated exchanges, query-based retrieval of information, representation of component information using product data and their resulting interoperability etc. These topics stem from needs recognised during project executions, where information is available, but not efficiently usable to make quick rational-backed decision-making.

So, the data are continually generated, stored, and exchanged, yet their overlaps and their relationships to each other are

not explicitly captured. This leads to manual interpretation of information, influenced by variable factors such as experience and knowledge of the project participant. Naturally, there is a need for handling these connected information in a structured way for the AEC industry, i.e. a bespoke linked information architecture. This architecture establishes the baseline interaction of the data and its design - facilitating a minimum requirements model that can be used for building future applications.

The BIM Maturity stages provide the criteria for assessment of information models in a 3-tier paradigm. This maturity model was first presented by the UK BIM Task Group[7]. This was later modified and included in ISO 19650 Part1. Fig. 1.4 shows these modified stages. Stage 1 requires file-based information storage, exchange, and management, with support for partial collaboration between project participants. The information so stored is not required to be federated, and need not necessarily contain structured data. Alternatively, Stage 2 denotes full collaboration, where the CDE facilitates the use of federated models, which are represented in a unified vendor-neutral format (like IFC). This stage also requires partial connection of project processes, technology and access roles. However, the data is still partially represented in files, creating data silos. Stage 3 indicates the usage of object-based server information models which are hosted on the cloud and can represent and manage federated information models. Functionalities such as querying should be feasible in both model and container databases.

[7] https://www.thenbs.com/knowledge/bim-task-group-april-4-mandate-an-internationally-unparalleled-achievement-on-our-bim-journey

This stage requires data modelled and stored in structured databases, differentiating it from stage 2 which is file-centred information management, without the use of object-based information models (like ontologies describing the underlying schema).

Semantic Web technologies have been shown to be of high value in solving some of the requirements for Stage 2 and 3 (Beetz, 2009; Godager et al., 2021; Pauwels et al., 2017; Rasmussen et al., 2021; Svetel & Pejanović, 2010). The approach of using ontologies for representing information, makes the data structured. Multiple ontologies, each describing domain-specific concepts can be connected together; this type of connection at the abstract level (concept level), instead of at the instance data level, fits into the object-based server information model. For example, an architectural model authored in Revit can be converted to the vendor-neutral IFC format. This IFC model can be also represented

in ifcOWL ontology, which is based on Semantic Web concepts. Smart devices installed in the building can be described in a separate model, authored in SAREF Ontology. The connection of ifcOWL and SAREF ontologies, enables the federation of both the instance data.

However, the key challenge is the detection of the connections between the ontologies and the instance data. These connections are not explicitly recorded in the models and the containers being exchanged. Hence, domain-expert knowledge and experience is often relied upon for the understanding and interpretation of the data. Due to varying levels of competencies, knowledge and experiences of project participants, these interpretations can differ, leading to different ambiguous decisions made based on the same data.

Currently, there are multiple other standards and approaches that can be combined to establish the requirements for defining the functioning of model and container federation for Stage 3. DIN SPEC 91391 Part 1 introduces concepts for container and the associated metadata requirements. These evolved from the concept of Multi-Model Container (MMC) which consisted of partial models federated within a container[8] (Fuchs & Scherer, 2017). For example, ISO 21597 [9] on the management of linked information using information containers introduced a structure for exchanging information requirements between project participants during the archiving phase. These containers are called Information Containers for linked Document Delivery (ICDD).

[8] developed from the Mefisto project, more details can be found in Section 12

[9] detailed introduction can be found in Section 17

The container concepts proposed in MMC and ICDD can be used as inspiration for the design of container-based information federation. The design of these containers according to linked data principles, enables information federation, and querying. Functional requirements from a CDE perspective is adapted and formulated based on existing specifications and incorporated in the container design.

The thesis focuses on the research carried out within the context mentioned above and aims to study the feasibility of SW technologies to support the representation and storage of heterogeneous data in CDE using information containers. It also focuses on the identification of the criteria for these containers and their design (technical and functional).

In the next section, the research questions are formulated for achieving the aim of this thesis.

1.4 Research Questions

The literature review performed in Chapter 2 acknowledges the core characteristics of information within AEC and the associated challenges encountered during the lifecycle of a project.

The broad research question based on this research gap is: How can heterogeneous data be linked and managed in web-based information containers in a CDE?

The above question is decomposed into three topic axes with corresponding sub-research questions:

RQ1: *Linking across domains* - How can informal representations of related heterogeneous information be translated into formal link representations which can be used throughout the asset lifecycle?

RQ2: *Information Management* - How can formal link representations be managed and processed within use case-based Information Containers in a CDE through open data standards?

RQ3: *Data Integrity* - How can link relationships, containers and stored data be checked for integrity and conformance requirements?

These three topics are investigated on the basis of hypotheses from the Semantic Web world which are elaborated in the next section.

1.5 Hypotheses

Based on the research questions in Section 1.4 the following corresponding hypotheses are:

H1: Federated model building involving partial heterogeneous models can be effectively addressed by leveraging Semantic Web technological concepts.

H2: Domain-specific functional and conceptual requirements, as specified by ICDD and MMC to facilitate linked data-supported containers in CDEs are feasible by adopting Linked Data paradigms.

H3: Conformance of interlinked data to pre-defined metadata can be verified by leveraging the data shapes as defined

in SHACL.

H4: Concepts proposed by ISO 19650, DIN SPEC 91391 can be combined with ISO 21597 and Linked Data Platform (LDP) to achieve a functional information container. These containers can be used in a web-based decentralised CDE for information management throughout all project phases, rather than just for information exchange during handovers.

H5: The SHape Constraint Language (SHACL) rule-language can be leveraged under the Closed World Assumption. for conformance checks of both incoming data and created links. Some of these checks can be created as standardised constraints in the SHACL Shapes, given existing comprehension of incoming data schema. Standardisation of constraints helps in their reuse, as rules are split into modules, with constraints in reusable shapes, which can be allocated to any target object.

In the field of formal logic, a closed world assumption means a statement is true only if it is known to be true. In this case, the knowledge database is known to be complete, and any missing information is treated as untrue. For example, if a query searches for flights between Berlin and Hannover and the result shows that there are no flights available under the CWA, this result is true. Further elaboration of this topic can be found in 2.1

1.6 Research Outcomes

The research conducted for this thesis illustrates the suitability of combining Semantic Web technologies with BIM for managing heterogeneous interlinked data in a web-based Common Data Environment.

This research presents a concept for the data architecture and process flows for linked building data within information containers, used within a CDE. To accomplish this, the requirements for a container in all phases of a project, and its subsequent architecture in a CDE. It leverages the ICDD container concepts and integrates them with the requirements for CDE. Furthermore, this research also contributes to the application of SHACL for pre-validation checks, thereby using this rule language for filtering incoming data. Additionally, shortcut functions were also developed which standardised some of these rule-sets, for better re-usability.

A proof of concept was developed and demonstrated. The beneficial and problematic aspects of their concepts and applications were also identified and investigated during these explorations.

1.7 Guide to this thesis

This thesis is split into 6 chapters.

1. Chapter 2 establishes the theoretical base for this research by assessing the current state of the art in both research and practice on information containers and stage 3 of BIM maturity stages, and the point of departure (both theoretical and practical) is established. Additionally, it introduces the general concepts in the Semantic Web world which are relevant to this thesis.

2. Chapter 3 focuses on the methodology adopted for addressing the research questions posed in Chapter 1.

3. In Chapter 4, the complexity of interconnected data in AEC industry is explained, and the processes for discovering link relationships between models and images in the context of this thesis are introduced. Additionally, the ontologies necessary to achieve this are also introduced. This chapter is supplemented with the appendix B, which contains the mappings between the ontologies.

4. Based on the the link discovery and linking process, Chapter 5 introduces and formulates the proposed data and process architecture for the information containers, which will store and maintain interconnected heterogeneous data. To support this architecture, an analysis of the functional capabilities, as proposed by existing standards and identified practical challenges/needs from the industry are presented

5. Chapter 6 elaborates on the validation of the interlinked data (which is presented in Chapter 4) achieved using the SHACL rule language. Additionally, checks were formulated for metadata of the incoming models/images/other heterogeneous resources using the same rule language. The requirements listed in this chapter and their corresponding SHACL shapes are documented in Appendix A.

6. The final Chapter 7 presents the discussions based on the research questions. It discusses the corresponding outcomes for these research questions and their implications on broader themes under focus in the research and industry communities. It also includes the current challenges faced in this domain, and the potential ways forward for breakthrough.

Chapter 2

Related Work

This chapter provides a general overview of existing research approaches, existing solutions in the field of BIM-based CDEs, and Semantic Web technologies for interlinked data. This chapter is split into three core sections:

1. The first section provides a concise overview of the concepts and technologies used in this research. Section 2.1 introduces Semantic Web and Linked Data. Furthermore, the following key topics which are relevant in the context of this research are introduced:

    - discovery of links in data,
    - management of links,
    - validation of links and
    - validation of metadata. Due to significant overlap, the terms Semantic Web and Linked Data are used interchangeably in this thesis .

2. Section 2.5 presents the specifications set by current AEC and non-AEC standards, while Section 2.5 reviews the currently available off-the-shelf commercial solutions for their relevance to BIM Stage 3 and linked data-supported CDEs.

3. Section 2.7 presents the analysis of gaps in research and practice, and identifies contributing areas from this research.

This chapter introduces only the SW concepts relevant to this thesis. More comprehensive explanations can be found in (Gayo et al., 2017)

2.1 The Semantic Web: Foundational Concepts

The Semantic Web was conceived as a collective, collaborative W3C standard focused on the conceptual extension of the World Wide Web. Its goal is to transform the current mix of unstructured and semi-structured documents on the Web into a "web of semantically enriched data" ((W3C), 2007). The Semantic Web technology stack provides the architecture to achieve the goals proposed above. Fig. 2.1 is an adapted version of the original stack proposed in ((W3C), 2007).

In this layered architecture of hierarchy of technologies, each layer exploits the capabilities of the layer below it, to provide an extension of the traditional Web[10] which eventually will evolve into a Semantic Web. In Fig.2.1, the components not relevant to this research are greyed out. The relevant parts, which are highlighted in the figure through different colours are briefly explained in the bottom-up direction.

[10]The term 'traditional web' denotes the human-readable content - i.e. a collection of unstructured and linked documents composed of unstructured documents consisting text, images, and hyperlinks. Unlike the Semantic Web, its data is not machine-readable.

Over the years, numerous tweaks and modifications have been proposed for the SW stack. Though these versions address various aspects like IT architecture, business logic, etc. (F. Sowa, 2021)

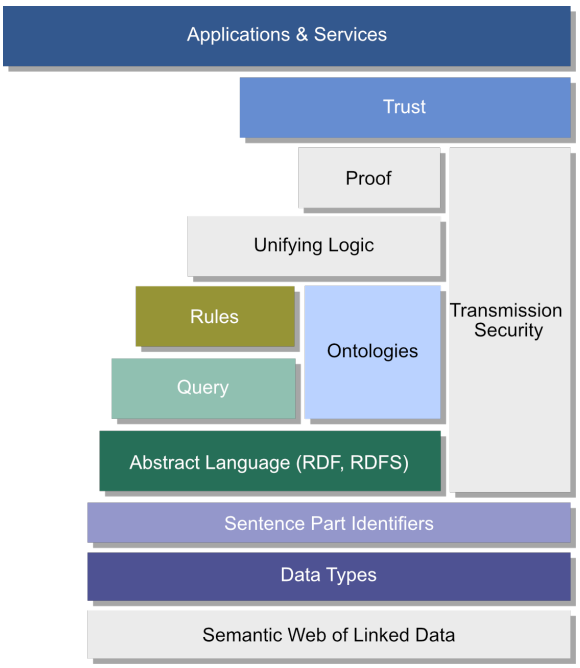

Figure 2.1: The Semantic Web stack, adapted from ((W3C), 2007)

In this conceptualisation, information is handled under the Open World Assumption (OWA), where if data is not present within the currently available data set, it does not result in 'missing' data. This means that the system can neither confirm nor deny the existence of the data, as it assumes

that these non-available data can appear at a later point in time. However, within the real-world AEC ecosystem, data is always handled under the Closed World Assumption (CWA). This means that if data is not available for a particular query, the result is a definitive negation. As this thesis focuses specifically on AEC-related use cases and scope, the research has been carried out under CWA.

Regardless of the version of architecture adopted, the overall layers remain the same. For this thesis, the original stack as proposed by W3C, with minor simplifications has been utilised

Data Types

This layer describes how data is described in a persistent form. In general, Resource Description Framework (RDF) is used for data exchange. RDF uses a triple structure to encode the information. Triples typically contain a 'subject', 'predicate', and an 'object'. Though information can be represented using the RDF notation, its final serialisation can be in any other RDF-supported format (e.g. turtle, JSON-LD, rdfXML, etc).

Part Identifiers

Uniform Resource Identifier (URI) and Internationalized Resource Identifier (IRI) are used to identify the resourced defined in triples. An Uniform Resource Locator (URL) is a special type of URI that is used in the World Wide Web (WWW) to locate resources within the Web. For readability, these URIs are shortened using prefixes, where a base-URI is used as a local term for the identifier.

Abstract Language

As mentioned previously, the RDF language provides the identifiers, syntax, and semantics to encompass information.

Ontologies

The Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL) are schema built on top of the RDF vocabulary. These two are used to describe the data and its associated schemata. Essentially they act as dictionaries consisting of formal definitions of an entity (i.e. its description) and its relation types (i.e. classes and properties).

Query

In the context of the Semantic Web, querying is facilitated using SPARQL Protocol and RDF Query Language (SPARQL).

The language is a recursive acronym developed and published by World Wide Consortium (W3C), it retrieves and manipulates data stored in the RDF format. These functionalities include general querying, updating, deleting, and inserting data into the triples graph.

Rules

Rules in the Semantic Web (SW) world are premises upon which reasoning and inference can be made. This is achieved using the language format which is used for describing the premise. Rule languages like RDF Mapping Language (RML), Semantic Web Rule Language (SWRL), SHACL can be used to define constraints, with varying scope. However, the results of the application of these rules (i.e. if it is false or true) cannot always be used by RML or SHACL to insert new data or remove existing data. However, they can be used to create new statements that add value to the existing data.

Trust

In general understanding, trust is whether a piece of information can be believed and is verifiable. Similarly, in the SW context, trust relates to the reliability of the identity and provenance of the content. This layer is intended to allow users to verify whether the source of information is being judged based on its trustworthiness, providing quality assurance of the data it returns.

Application & Service

This layer consists of end-user-focused applications or services, which allow them to interact and visualise data based on SW principles. It also requires the data to be both machine-readable (as it would have to conform to all the layers below it), and also human-readable. Though these applications can be either declarative or imperative programming, recent trends have shifted towards the former. Declarative programming involves describing the goal of the program without explicitly specifying the control flow needed to achieve it. In imperative programming, the program specification explicitly states each step-by-step instruction for achieving the goal. Due to SW's inherent decoupling of data from the applications, it is feasible to build data-driven software where any differences in the core domain model, its features, or the user interface are encoded as application data

rather than in the imperative data.

2.2 Link Discovery and Management

Information stored in various formats often contain points of reference which allow them to be connected. These connections are termed *links*. Link Discovery is the process of analysing data to find tangible links to each other. Within this thesis context, priority is given to the discovery of potential links between related heterogeneous data generated in AEC projects. So, knowledge mining involves the analysis and discovery of links in the metadata of images, models, and documents. These metadata are usually encoded using a domain-specific ontologies; e.g. IFC models can be encoded as ifcOWL models, using this ontology, or even as other linked building data models. The technical overview, as well as the specific methods that are used for the identification of these links for each type of data, is presented in Chapter 4.

2.3 Data Integrity and Conformance

Data integrity is the overall accuracy, completeness, and consistency of the data. This also includes compliance of data with safety standards. Integrity relies on a set of processes, rules and standards and these serve as the context for the usage of the data. Chapter 5 provides an in-depth introduction of rule checking for AEC-related data, and the use of linked data supported rule languages.

2.4 Information Containers and CDEs

The information that has been generated in a project must be disseminated among all project participants in a time-bound manner to enable subsequent activities. This information transmission also influences the speed and quality of project management decisions. However, significant standardisations of data and processes are necessary to prevent haphazard management of unstructured data and consequent use of ad-hoc processes for using said data.

Information Container (IC) plays a central role during information exchanges. They help bundle relevant files together so that downstream information processing can be performed. Various forms of containers have existed since the

start of digitalisation in the AEC industry. The simplest of these are the folders used on our computers, for cataloguing and ordering documents based on use cases. However, these folders do not contain additional metadata on how the files within them are related. Isolated files can be connected using software that supports them (more details are explained in Section 2.5).

ICs can contain machine-readable data, making them structured ICs, or have texts/drawings/images etc. which are not explicit (at least not without employing some form of Artificial Intelligence like Optical Character Recognition or Machine Learning).

CDEs are utilised throughout the life-cycle of a project, and serve as the platform for mutual loss-less data exchange, thereby becoming an agreed source of truth. ICs that function within these CDEs are created, collected, processed, and distributed/shared with project participants through a controlled process.

This next section evaluates the currently available market implementations of CDEs, based on reported features in published literature. Their shortcomings are identified and used as a basis to establish the practical points of departure for linked data supported CDEs. Additionally, Information Containers and CDE-related standards and specifications are reviewed, along with relevant research studies focused on these themes. This comprehensive analysis is used to establish the current state of the art, and identify the gaps in existing literature. Finally, the challenges encountered in this domain are summarised.

2.5 Practical Points of Departure

In this section, a brief review of the capabilities of commercial CDEs available in the market are assessed. The core criteria used for this evaluation is whether such CDE software: 1) supports BIM Stage 3 functionalities as defined by ISO 19650 or DIN SPEC 91391 and 2) supports information container-based federation of heterogeneous information through facilitating link creation and management. These software are often black-boxes, require licenses and their capabilities do not have full-fledged documentation for their free versions.

Current commercial solutions

CDEs in some formats have existed for some years as off-the-shelf commercial solutions. Autodesk's e-Transmit tool supports the preservation of relationships between different CADD files in a container (Maier & Fair Cape Consulting LLC, 2020). There are numerous solutions which reportedly conform to BIM Stage 2 or 3 like Autodesk's BIM 360, Trimble Connect, Allplan's bim+, Graphisoft's BIMcloud, etc.

Lagazio, 2018 showed in a study on federating BIM models that Autodesk's Navisworks supported the federation of native models generated from other BIM authoring tools in a viewer. These models were merged into a single file - the federated model using BIM360, another tool in the Autodesk ecosystem. Clash detections, cost estimations and construction simulations can also be conducted on federated models. Preidel et al., 2016 assessed various existing CDEs for their seamless integration for collaboration. It assessed the API features of BIMserver, BIM 360, BIMcollab, BIMcloud, bim+ and Trimble Connect. It concluded that these CDEs comply with the traditional Single Source of Truth (SSoT) CDE paradigm, with web-based access and also implement ISO 1960, DIN SPEC 91391 concepts such as versioning, authentication and data management.

However, these tools are not consistent in implementation of all specifications listed in these standards. For example, DIN SPEC 91391 stipulates the feasibility for linking to other resources in containers. It also requires the linkage of heterogeneous, non-BIM information to BIM models. Furthermore, the research also found that these CDEs rely heavily on native proprietary data formats, thereby making them 'closed-BIM' solutions, which lack full access and transparency of data and cause interoperability issues.

A predominant practice in software development is to develop tools that fit in the provider's ecosystem, with little consideration for integration with 3rd party tools. Most often, project owners/clients request for additional developments for these integrations, which are then implemented on a project-by-project basis. A type of these developments are called middleware - interfaces where different tool's API can accessed in one view for viewing/modifying/exchanging data between the tools. However, middleware development is not standardised, and can vary between different organisations based on their internal structure and IT policy.

Bucher and Hall, 2020 conducted a similar study and classi-

fied CDEs in terms of increasing levels of dimensional interoperability: one-dimensional (interoperability within a CDE - limitations in collaboration due to restrictive data formats), two-dimensional (interoperability between different CDEs) and three-dimensional CDEs (distributed CDEs). It concluded that most software are mapped at a 1-dimensional level as they only support data exchange within the software vendor ecosystem. Although these dimensional levels cannot be directly translated to the BIM maturity stages, some of the concepts overlap. For instance, two-dimensional CDEs can be mapped to either Stage 1 or 2 as they support data exchange in vendor-neutral open data formats, while three-dimensional CDEs can be mapped to BIM Stage 3 or beyond, as it supports decentralised interconnected CDEs.

Poinet et al., 2021 noted that BIM's inadequacy in supporting these data structures produces both friction and waste during production, including painful remodelling processes and incomplete or low-fidelity documentation for complex geometries. It is also a time-consuming process that directly contradicts BIM's fundamental utility for project delivery: to simultaneously manage design geometry across multiple, interdependent, multi-media and on-demand representations.

Based on the above analysis, this research establishes that there is a wide gap between the theoretical concepts that a CDE should support, and the current solutions existing in the market. Additionally, most of the current solutions can be classified into either Stage 1 or 2, and do not contain any capabilities for supporting either linked data concepts or heterogeneous interconnected data throughout the asset lifecycle. Consequently, the main points of departure are: 1) existing solutions do not fully implement available specifications for managing interconnected information in CDEs.

2.6 Theoretical Points of Departure

In this research, the theoretical points of departure are used in two contexts: 1) established national/international standards, specifications and recommendation reports by working groups and, 2) research literature. For assessing the relevance of research works for this research, the following criteria in Fig. 2.2 is used.

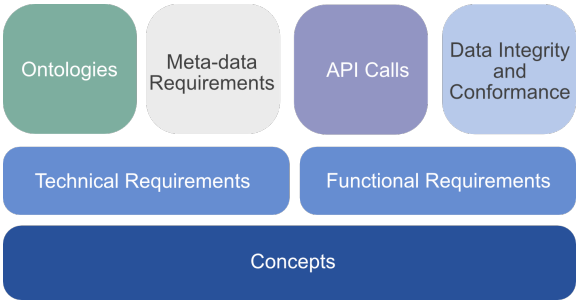

Figure 2.2: Criteria for review of standards and approaches

2.6.1 Current standards and specifications

ISO 19650

The concept of CDE-based information management was first introduced in the now-withdrawn BSI PAS 1192:2013. This standard is now replaced by the ISO 19650 series and serves as the starting point for this research. Part 1 of this standard focuses on information production and management for the entire lifecycle of a built asset, while Part 2 focuses on the information management during the handover phase of an asset. In general, Part 1 offers guidance for establishing an effective information management framework that encompasses data exchange, storage, versioning and organisation.

First, a distinction is made between the terms *'information model'*, *'federated model'*, and *'information container'* and their usage. According to ISO 19650, an information model is conceptualised to be created from a set of either/both structured and instructed information containers, while a *federated model* is defined as a composite *information model*, which itself can be created from distinctly separate information containers. *Information containers* can be a set of persistent data (e.g. a file on the local machine, on the cloud etc.). In the context of this research, a *'federated model'* is defined as a model which stems from combining data present in one or multiple containers. This effectively means a *'federated model'* can be created using just one container's contents.

At its core, this standard defines the maturity model for the use of BIM, according to a 3-tier development. These maturity stages conceptualise how at each stage, information gradually evolves from a file-based, i.e. the lowest level of digitalisation, to storing information in databases to full leveraging of web-based functionalities. Fig. 2.3 shows an adapted version of this in which all the standards assessed

in this section are mapped according to this maturity model. Stage 1 encompasses two-dimensional design work on paper or electronic equivalents without defined collaboration processes, though they are already using a CDE.

Meanwhile, Stage 2 extends the tasks of Stage 1's CDE to the capabilities of organised data exchange between all project participants by enhancing the collaboration process and ensuring lossless information exchange of federated information models using open formats. Stage 3 represents digitalisation where data is linked at the object-level and the attributes are queryable using a container database in the CDE.

ISO 19650's aim for container-based information is rooted in its emphasis on collaborative working. This standard provides guidelines for information production by the author, outlining clear and pre-defined requirements that must be agreed upon by all parties involved. However, it is lack of specificity regarding how a CDE should store and manage shared, federated models. Additionally, while the standard acknowledges the importance of data integrity[11] - including authorised access, and protection against corruption and obsolescence - it does not offer specific methods for achieving this.

[11] in this case - immutability

Nevertheless, it does outline key structural elements for a successful federation. For example, federation must support access to various types of models (like spatial, geometrical and semantical), that can be stored in an Information Container with a defined breakdown structure. These ICs are expected to be dynamic in nature, as they capture continuously evolving project data through the different project phases.

The major gap identified in this standard is its lack of detailed specifications on how these requirements can be achieved. It leaves the task to the project teams, who will have to agree not only on the federation strategy but also on the technology to enable this. It also does not delve into the details of the metadata and the functional requirements for realising these criteria, thereby becoming a mere guideline, and not an approach suitable for implementing CDEs. This standard requires that pre-existing information about the project should be exchanged between concerned parties before and after the project, though it does not tackle how this information should be integrated. It does not explicitly mention heterogeneous federation, and the associated challenges like: the approach for object identification within this data (i.e.

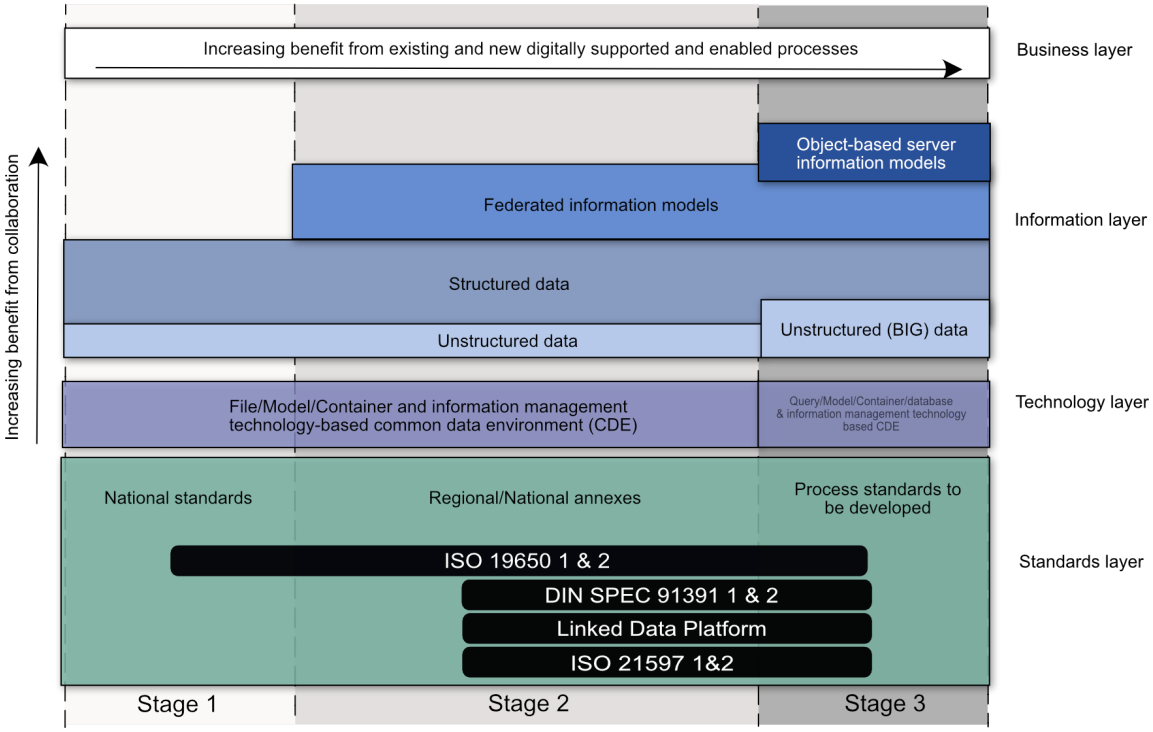

Figure 2.3: BIM maturity stages as per ISO 19650 mapped to standards and approaches(14:00-17:00, 2018-12)

snippets in an image, cell identifiers in spreadsheets, etc.) and the level of the federation feasible. Is it applicable only for the file-level, resulting in a visually superimposed federation, or is it applicable for object-level federation with explicit linking? In its annexes, it provides example illustrations of federation strategies, which focus mainly on visually federated models, where the objects are spatially aligned. In these strategies, discipline-based containers are created; architectural information is stored in the *architectural containers*, MEP information in the *MEP containers* and so on. However, it is unclear whether other heterogeneous data like images, pdf documents can also be federated and if there is additional information on this federation, such as a context/use case for the *federated model*, etc. These kind of heterogeneous information fall under 'data captured' during surveys of the construction/project site. They are documented to assess as-is state of the infrastructure in renovation & retrofitting projects. Often, such images are federated with available BIM models or pdf drawings to provide a complete picture of the infrastructure to project participants.

Although the ISO 19650's scope does not cover the technical aspects of information linking, it establishes the overarching principles for information management in a CDE. These cov-

ered the roles & responsibilities of the project participants in a web-based CDE, and the collaborative strategy to manage federated information. These specifications are relevant to the topic of functional aspects of CDEs, and their further analysis and application can be found in Chapter 5.

Part 2 of this standard focuses on requirements associated with information management, i.e. delivery milestones, information standards, data production methods, data referencing and sharing, and CDE selection during the project's delivery phase. CDEs, according to this standard are required to have containers with unique identifiers (which are to follow a prior agreed-upon convention), and codification of fields (ISO 19650-2:2018, clause 5.1.7).

More importantly, it also requires ICs to have metadata like status, revision, and classification associated with it. It also defines functionalities of the above-stipulated containers: 1) transitioning ability between states, 2) capturing and retaining container author information during these transitions, and 3) access controls. Consequently, due to the elaborateness of these requirements, Part 2 focuses on the information management process itself, roles and responsibilities of the project parties for information delivery, the requirements for information which will be delivered, and the guidelines for selecting the CDE.

The ISO 19650 establishes the minimum requirements for the shift to Stage 3-compliant data structuring using standardised frameworks/object server databases. Hence, this standard can be mapped as mapped as partially complying with Stage 3 (refer to Fig. 2.3). Based on the content review of the scopes of both the ISO 19650 Part 1 and 2, it is clear that ISO 19650 presented the conceptual starting point for the establishment of ICs and CDEs and how the former should function within the latter. However, due to its focus on the high-level overview of information management, it is not detailed enough on structuring and on the minimum requirements for both ICs and CDEs.

Additionally, though it emphasises 'federation', it does not mention the level of federation possible/recommended, i.e. if models are merely superimposed on each other, becoming visually federated, but do not contain actual links to the federated objects themselves. Neither Part 1 nor Part 2 delve deep into the structuring of the ICs, leaving the task to project teams. However, other standards and approaches do address some of these gaps.

DIN SPEC 91391

Where the ISO 19650 fleshed out the broad benefits of using an IC-based CDE for information management, the DIN SPEC sketches the functional level details for ICs and CDEs, the use case contexts that will be supported by them, and the minimum viable function set for an operational CDE. Part 2 of this specification addresses the interfacing of CDEs when multiple CDEs are used by different project participants to exchange and manage information throughout the project lifecycle.

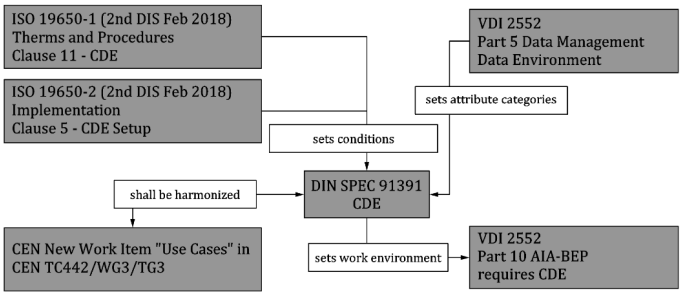

Figure 2.4: DIN SPEC 91391 and its relevant standards (*DIN SPEC 91391-1*, 2019-02-19)

More importantly, the DIN SPEC is proposed to fit in with other standards as shown in Fig. 2.4. While the VDI 2552 also focuses on CDE functionalities, it is a German national standard based on DIN SPEC 913941 and as such does not introduce new concepts.

In Part 1, the technical, organisational and individual principles and functions from ISO 19650 are defined, by splitting them into two categories: obligatory and optional functionalities. However, conceptually the DIN SPEC defines an IC as the smallest storage unit, which can, in turn, contain models, logical constructs etc. or a file alone can be considered as an IC (DIN SPEC 91691:1 section 3.5). This means that ICs are the smallest referable unit (aggregation levels as per DIN SPEC 91391:1 section 3.12) and this can point to a set of files or a single file. This ambiguous conceptualisation conflicts with the notion of containers from other standards and approaches such as the LDP, MMC based on DIN SPEC 91350, ICDD based on ISO 21597, which are introduced in the upcoming sections.

DIN SPEC also extends the Stage 3 conceptualisation of ISO 19650 BIM maturity stages, by requiring that the individual model elements/attributes are the smallest unit of referable

information. This means that elements in an IFC model, e.g. a wall, can be linked to material documentation reports or Bill of Quantities (BoQ) spreadsheets. However, it does not address the feasibility of referring to individual element linking in non-BIM models, e.g. particular cell in a spreadsheet.

DIN SPEC assesses itself that it refers to maturity Stage 2, with partial applicability for Stage 1 and 3 (Introduction chapter, page 7). Principally, it also mentions the ambiguity in the definitions of Stage 3 and the lack of standards that address this stage. However, it states that this stage has to widen the data model of Stage 2, by enabling the storage and processing of element and attribute-level information. Furthermore, it also acknowledges the lack of an ontological base for the metadata requirements that stem from the data and processes needed for ICs in CDEs. This also means that the data structuring for storing this information is not addressed. This conceptual requirement leads to a review of three approaches of interest that can potentially address this: LDP, ISO 21597, ISO 17632.

A comparison of the features reviewed in this specification with respect to the other standards/approaches are mapped and presented in Fig. 2.8.

Linked Data Platform

[12]World Wide Web Consortium https://www.w3.org/

[13]A type of application programming interface which conforms to REST architectural style, and allows interaction with these web services

LDP is a W3C[12] specification which defines the approach for reading-writing linked data, i.e. RESTful[13] Hypertext Transfer Protocol (HTTP) way for creating, read, updating, deleting, and thereby consuming resources (Spiecher et al., 2015). In this context, the term *resources* (i.e. information) includes both RDF and non-RDF resources. For example, IFC models are structured data and can be represented as RDF graphs using the ifcOWL schema or using Building Topology Ontology (BOT) schema. Non-RDF resources include images, and documents since they are represented in their own formats such as `.jpeg` or `.pdf`. So, they are non-RDF resources, as their data is not represented according to RDF schema. However, their meta data like file name, file type, size, date of creation, and file owner can be stored in a RDF graph as an RDF resource.

Essentially, the LDP specification allows the consumption (including read/write functionalities) of RDF and non-RDF resources on the web, through the use of HTTP services. It also defines constructs for `container`[14] - a special type of resource, which consists of collections of RDF/non-RDF re-

[14]Henceforth referred to as `LDP Container`

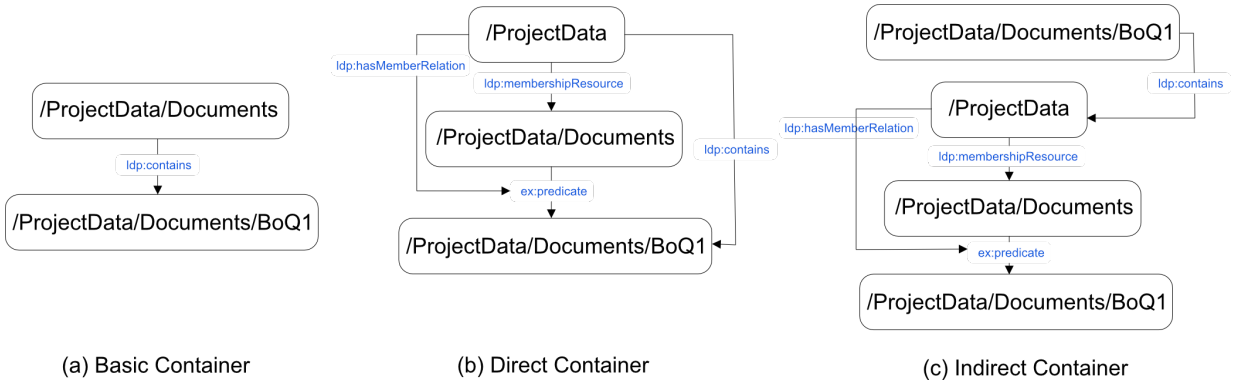

Figure 2.5: Types of LDP containers

sources bundled together. `LDP Containers` are accessible using the container's URI. An `LDP Container` can be of three types: 1) `basic container`, 2) `direct container`, or 3) `indirect container`.

A basic container aggregates its resources it holds, thereby creating a containment hierarchy of its resources. Here, these resources will always have some relationship with the container (the subject of the triple is always the container). Figure 2.5 (a) shows an example of a `basic container` for storing project information using LDP-defined vocabularies. It uses the predicate `ldp:contains` for establishing the relationship of the resource `:BoQ1` (Bill of Quantities) to the container it is residing in.

A `direct container` expands the `basic container`, and can store resources that may not be directly linked to the container (i.e. the subject of the triple can be other resources besides the container itself). This means that although it collects resources stored within this container, it cannot handle relationships to other external containers. Fig. 2.5 (b) shows a new triple with a predicate `hasDocument` is added, due to the presence of `ldp:membershipResource` and `ldp:hasMemberRelation`.

An indirect container, while bearing similarities to `direct container`, allows for storing of resources where all elements of a triple (subject, predicate, and objects) can be explicitly defined. Based on the previous example, Fig. 2.5 (c) depicts an `Indirect Container` that stores a resource named `:ProjectData`. This container stores the membership triples of both basic and direct containers, while providing the flexibility to configure the object of the membership triple (in the Fig. the `ex:predicate` which is related to `ldp:hasMemberRelation`).

LDP offers significant flexibility in terms of defining the metadata that should be included in a container and its contents. This allows software implementors the freedom to design and structure information containers, resources and their metadata. As discussed at the beginning of this chapter, the AEC industry primarily operates within the *closed world* paradigm, with most tasks and activities falling under this category[15]. This means that information is created, stored and analysed to make effective and informed decisions as part of project control.

[15]There are exceptions - For example, digital twins of infrastructure and its maintenance often do not have full-fledged information of the component/building element installed. This can be due to the Level of Information of the BIM model. While the information is absent in the digital twin model, it cannot be assumed an erroneous model. Hence, these can fall under *open world* paradigm.

However, when there is no standardisation for minimal data requirements, it can lead to a multitude of interoperability issues. Project participants can exchange poor quality information, which results in multiple Request for Information (RFI)s and delays in schedule. These issues also apply to ICs, as varying minimum data requirements can lead to non-compliant information exchanges between participants. It is therefore crucial to establish and adhere to standardised data requirements to ensure efficient communication and collaboration of data among project participants.

Beyond the container specifications, LDP also outlines standard techniques for creating clients and servers that handle the storage and management of RDF resources. Since, web-based CDEs also based on Client-Server model, LDP's specification can serve as a guideline for the functional criteria and behaviour. These criteria are mapped and compared with other specifications and standards shown in Fig. 2.8, and discussed in detail in Section 2.7. Additionally, it also provides basic vocabularies for container and resource-related metadata, which can be supplemented with other linked data vocabularies like **Dublin Core**[1], or **schema**[16] for author/ownership metadata, versioning, etc.

[16]A collaborative set of extensible schema/vocabularies facilitating metadata representation in RDF https://schema.org/

Based on the focal point of this research, both the direct and indirect containers are highly relevant with respect to their container definition and structuring. Their definition also contain overlaps with other container definitions from the DIN SPEC 91391 and ISO 21597. These are discussed in Section 2.7.

ISO 21597 ICDD

The ISO 21597 standard on ICDD is a two-part standard, with Part 1 focusing on general specifications for the break-

[1] A set of metadata items for describing the digital and physical resources

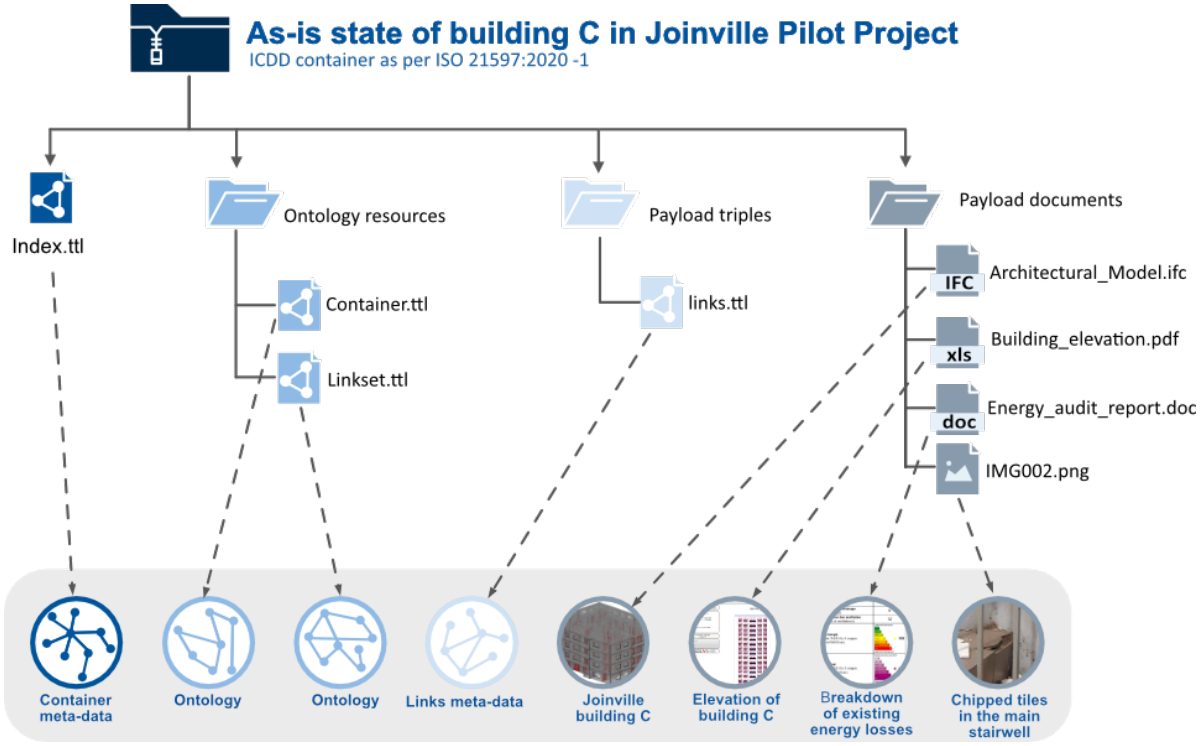

Figure 2.6: Container breakdown structure as per ISO 21597

down of containers for heterogeneous data and their associated metadata, while Part 2 focuses specifically on the various link types as potential relationships for the data within the containers, page (Technical Committee : ISO/TC 59/SC 13, 2020). It is intended for managing delivery of heterogeneous information (e.g. drawings, models, textual documents, spreadsheets, images, point clouds, etc.) during the handover processes between phases and participants, Part 1 of this standard defines the composition of such a container. In this part, the relationships between the encompassed data are defined using the ICDD's *linktypes* structuring and vocabulary.

The ICDD standard was built on collective previous experiences in designing containers for storing multiple partial models and resources. The Construction Objects and the INtegration of processes and Systems (COINS) project, which started in 2003 laid the groundwork for the development of ICDD later. The COINS project had a varied representation of participants from the civil engineering domain in the Netherlands. They included municipal corporations, research institutions, universities, software companies, and construction companies.

The main goal of this effort was to improve content-based

communication between a typical project. All types of information created and transmitted in project are included in this communication, i.e. 3D geometry models, object-specific information like materials specifications, planning schedules, costing estimates, etc. The effort resulted in the development of a container system which facilitated the storage of the content(i.e. resources) and the links between them. Despite the impact of the initiative, it struggled to be implemented in practice (van Nederveen et al., 2010). However, the successive COINS2.0 contained improvements for the container structuring. This version of the project led to the development of ICDD.

Another development which also had an impact on container design is the MMC. Originally developed as part of the Mefisto project[17], these containers were designed to aggregate the partially distributed information models exchanged within a project and their link relationships. These relationships were captured in a `link model`, which was bundled within the container. The `link models` explicitly specified the type of interdependencies and the object(s) concerned. This was achieved by using identifiers for reference objects. Standardised model schemata like the Industry Foundation Classes (IFC) and the German Gemeinsamer Ausschuss Elektronik im Bauwesen (GAEB)[18] were used to represent partial distributed information models.

[17] https://tu-dresden.de/bu/bauingenieurwesen/cib/forschung/researchareas/bim-technologien/index

[18] The GAEB XML format is a German initiative, that serves as an uniform exchange standard for building information.

MMCs also contain metadata about the resources bound within the container. The MMC served as the inspiration for the container concepts introduced in DIN SPEC 91391 parts 1 and 2 and also for ICDD.

An ICDD container is a zip file that has a meta-file called `index` and a three-part folder structure dedicated for storing:

1. Ontology resources,
2. Payload triples, and
3. Payload documents

Fig. 2.6 shows the image of a conventional ICDD container. The `index` file consists of metadata about the container itself, its purpose, and the author information. Additionally, it also specifies which documents are being bundled within this container and their own metadata e.g. file names, revision history, authorship, etc. The example shown in Fig. 2.6 contains metadata about the 4 documents contained within the `Payloads documents` folder.

In the ontology folder, all relevant ontologies related to the container and the data can be stored; in this case, it stores the ICDD container and linkset ontology. The Payload triples folder contains the `links` graph, which declares which files are "link-able" elements, and which elements within these files are "link-able". Additionally, it specifies the structure for how each of the declared linkable elements can be connected using the specialised vocabulary from Part 2 of the standard, like `els:IsPartOf`, or `els:HasMember`, etc.

Unlike both ISO 19650 and the DIN SPEC 91391, the ICDD standard specifies an explicit ontology for both the container metadata, the file metadata and the links. An advantage of these containers is that they can be used to link at both the file level and also at the individual object level (within these files). This object-level link is denoted as *deep linking* (Borrmann et al., 2021). However, ICDD does not contain any metadata requirements nor dedicated vocabulary for links themselves, e.g. link creation history, authorship, modification history, algorithm used, etc.

The container structure proposed by ISO 21597 is one of the few that establishes an explicit foundational-level breakdown structure. However, both parts 1 and 2 do not contain any mention of the functional requirements for the behaviour of the containers, or its usage in a CDE. The original intention of this standard was to facilitate information transfer during the handover phase (between teams and between actual project phases). The default state of the data within these containers is `archived` as per ISO 19650 definitions. Consequently, its definitions were not unforeseen to accommodate the dynamic changes that occur during the design or execution phase (Introduction chapter of ISO 21597-1:2020, page v).

Although these containers have `archived` status, this is neither an ICDD metadata nor explicitly mentioned in the standard. Since this standard was never meant for information management during the entire project lifecycle, it does not contain specifications for its functioning in a CDE, like DIN SPEC 91391, LDP or even broad specifications like the ISO 19650. Furthermore, the ICDD link representation relies on a three-tiered hierarchy of triples. This can result in massive file sizes and a lack of readability.

VDI 2552, DIN SPEC 91350:2016-11

Similar to the ISO 19650, the VDI 2552 Part 5: 2018 defines a set of minimum requirements for CDEs (refer to Fig. 2.7). The workflows are required to cover the CRUD functionalities, with two special additional requirements: "filtering", and "structuring and linking existing data". For structuring the IC, this standard refers to BIM-LV containers proposed by ("MMC Multimodell-Container", 2016; Schiller et al., 2016). BIM-LV is a type of MMC used to combine partial BIM models with other models (e.g. gbXML etc.) for the integration of distributed data.

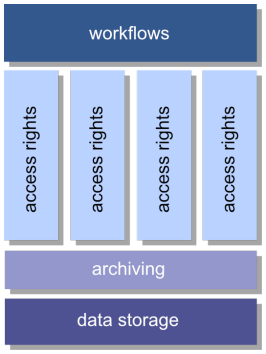

Figure 2.7: Functional block-based requirements of a CDE as defined in VDI 2552(und Gebäudetechnik (GBG), 2018)

The term *LV* denotes "Leistungsverzeichnissen" - bill of quantities or service specification is an exemplary use case container consisting of resources connected together. These include information of one (or more) building model and the data from the specifications. These can be IFC models that contain building components and a GAEB file connected to the models. These information are saved as as per the Multi-Model Container concept (*.mmc format). The service items (or their subsets) are connected to the components of the building model through the *link model*, which is also in the container. These contain facilitate the exchange of both the models and their service specifications in a single process.

Essentially, it consists of three components, which are delivered as a ZIP file (similar to ICDD): 1) a MMC description, 2) Link model containing the links, and 3) the application models (i.e. the payload BIM and other models). The first two components have to conform to the MMC schema ("MMC Multimodell-Container", 2016). The DIN SPEC 91350:2016-11 demonstrates a container for information exchange involving building models and bill of quantities, by adapting it for this particular use case.

```
<BUItems>
    <ItemLink>
        <BUItemID>C163</BUItemID>
        <BUsubItems>
            <ItemLink>
                <BUItemID>I1668</BUItemID>
                <BUsubItems>
                    <ItemLink>
                        <BUItemID>ID_1901_1668_1</
                            BUItemID>
                        <LinkedIFCID>1
                            vhGTRws18zAvKBjeq9YKc</
                            LinkedIFCID>
                    </ItemLink>
                </BUsubItems>
```

```
13              </ItemLink>
14              <ItemLink>
15              <BUItemID>I1669</BUItemID>
16                  <BUsubItems>
17                      <ItemLink>
18                          <BUItemID>ID_1902_1669_1</
                                BUItemID>
19                          <LinkedIFCID>1
                                vhGTRws18zAvKBjeq9YKc</
                                LinkedIFCID>
20                      </ItemLink>
21                  </BUsubItems>
22              </ItemLink>
23          </BUsubItems>
24      </ItemLink>
25  </BUItems>
```

Listing 2.1: Linking as per BIM-LV schema and structure

CEN 17632

Released in 2022, the CEN 17632 - Building information modelling (BIM) - Semantic modelling and linking (SML) - Part 1: Generic modelling patterns focuses on the top-level bundling of semantic information using reusable data-dictionaries approach (*EN 17632-1*, 2022). It focuses on the syntactic and semantic interoperability of asset-descriptive information using the Semantic Web stack. It covers the scope of selected W3C RDF languages like Simple Knowledge Organization System (SKOS), OWL, SHACL, RDF and XML Schema Definition (XSD). At the time of writing this thesis, this standard is yet to be fully published and consequently is not covered in this review.

2.6.2 Current research

The standards and approaches reviewed in the previous section form the basis for research exploration. Due to the broad scope of the CDE and information management within it, various perspectives can be used to explore the available literature.

Given the complexity of the topic, a combination of topics and keywords was used to filter the relevant literature for this thesis. These include **Information Containers**, **CDE**, **BIM level 3 or Stage 3**, **cloud-based platform**, **linked data management**, **Knowledge Graph Management**, **Deep Links**, **Interlinks**, **Decentralised CDE**, **Federated data models**, **Data or BIM Integration**, **Ontolo-**

gies, **Multi-model Containers**, **open BIM collaboration**, etc.

Although research articles were published before 2018 on the topic of linked data-based information management, due to the introduction of ISO 19650 in 2018 and other standards later, articles in both SCI-index journals and peer-reviewed conferences proceedings were considered.

Furthermore, a qualitative evaluation, similar to the approach followed in Pauwels et al., 2017 was adopted for the review. Articles were reviewed based on the problem they address and the magnitude of the resulting contributions based on the criteria listed in Table 2.1.

There are varying tiers of research reported in litarature. For example, the currently defined BIM maturity Stage 1 and 2 are popularly explored. Use-case oriented CDE design, which complies with Stage 3 were also commonly researched. However, aspects essential to Stage 3 namely Information Containers, requirements for querying of federated information were not researched. From this research's standpoint, the reviewed literature was classified into three main categories, where the research addressed either:

- Information Containers, CDEs and heterogeneous data
- Stage 3 of BIM maturity, and
- Implementation of ISO 21597 in use cases

Information Containers, CDEs and heterogeneous data

Fig. 2.2 defined a three-tiered criteria guideline for reviewing standards and approaches relevant to the research questions posed in this thesis. These guidelines encompassed concepts split into two categories:

- Technical Requirements: Ontologies, metadata requirements
- Functional Requirements: API Calls, Rule-checking languages.

The relevant literature on CDEs, Information Containers and Federated information were also reviewed using the same criteria.

Information Containers	CDEs for heterogeneous linked data	BIM Stage 3
federating different information representations	functionalities for heterogeneous data	querying
vocabularies for describing container concepts	processes for data interactions, API calls	integration with databases

Table 2.1: Criteria for evaluation of research articles

Implementing ICDD - ISO 21597

Section 17 introduced ICDD, its basic concepts, namely the structure of the container, its intended usage, and its capabilities for managing interconnected data.

Since the publication of this standard in 2020, there has been a significant increase in research related to it. The reviewed literature can be categorized into two distinct groups: those that utilized the ICDD schemata and approach without any alterations, and those that incorporated modifications such as integrating other ontologies, altering container structures, modifying metadata requirements, and enhancing functionality.

The current ICDD schemata have been proven to be suitable for various applications, including semantic linking of renovation-related data (Karlapudi et al., 2021), a billing system that combines BIM model, BoQ, and Quantity Takeoff for automated payment (Ye et al., 2020), dynamic data exchange for fire simulation utilising the IFC model and its supporting extension model (Al-Sadoon & Scherer, 2021), information integration for energy querying (Hoare et al., 2022), structural engineering approvals and permits application for IFC model, drawings and technical documents (Ciotta et al., 2021).

Different functional aspects of ICDD's capabilities were demonstrated in the above research. For example, Hoare et al., 2022; Karlapudi et al., 2021 utilised SPARQL queries for extracting links specified between resources contained in an ICDD container. Meanwhile, Ye et al., 2020 presented an interesting usecase where ICDD containers were used as evidence of work completed and used to process automated payments to the contractor using blockchain technology. As part of the study, conventional contracts were transformed into smart contract consisting of billing units of work. This was integrated into *BillingModel* which contained IFC models and the BoQ.

Al-Sadoon and Scherer, 2021 on the other hand developed a novel multimodel framework which supported dynamic data integration into the resources stored in an ICDD container. The link extensions developed as part of this work enable scenarios where information is constantly evolving. These include cost monitoring between planned and as-built infrastructure, facility management using as-built data etc.

A similar use-case, albeit for sensor data integration with BIM was presented by Polter et al., 2020. This study also utilised ICDD containers to integrate data like sensor data, and a BIM model for system identifications that were used for the simulation of production processes e.g. deep construction pits, tunnels, culverts, and underground pipes. Hagedorn, 2018 presented an implementation of a version of web-based access of ICDD, with a validator for verifying the conformity of a container to the standard.

Esser et al., 2022 explored the usage of ICDD for dynamic data from the version control perspective. Here, delta differences between monolithic BIM models were captured and links at the object-level using ICDD structure. Borrmann et al., 2021 presented the applicability for linking BIM models with 2D drawings at the file and object level. This usecase directly deals with information exchanged during the Design phases, where information exists in the `Work in Progress` state, with frequent updates and modifications. Such containers can be used within different disciplines, where federated information influences each other's design. Werbrouck, Senthilvel, et al., 2019 compared the use of specialised graph querying approaches like Hyper-GraphQL and GraphQL-LD for querying ICDD containers.

The containers in the above use cases stored both RDF and non-RDF resources - thus demonstrating ICDD's flexibility to accommodate common project information such design plans, images etc. However, they do not demonstrate the use of the specialised links defined in Part 2 of the ICDD standard (i.e. link relationships such as `isIdenticalTo`, `isAlternativeTo`, `isPartOf`). Link relationships that describe the extent of overlap between different sources of data are extremely invaluable to domain experts. They assist in understanding the significance of these overlaps and any resulting consequences arising from the connected information. In Information Containers, the degree of link relationships used is contingent upon the specific usecase; however, container constructs that incorporate both low-level and deep links relationships and structures are essential.

These features ensure that Information Containers can be effectively utilised throughout all stages of a project - from Conceptualisation to Operation & Maintenance.

Collectively in the above use cases, there was no reported need for dynamic containers; ICDD was a mechanism for bundling static data before an exchange, and hence the ICDD schemata could be directly used; sharing it as an archive folder through any means of data transmission, e.g. email attachments, uploads to cloud storage, etc. Inevitably, these implementations are not compliant with "BIM Stage 3", where data should not only be federated, but also live on the Web throughout the project lifecycle. This federated data should be queryable by project participants, who can filter and extract relevant information based on their need. Hagedorn et al., 2022 implemented ICDD in an OpenCDE compliant web interface by developing Application Programming Interface (API)s for seamless information exchange.

Furthermore, Hagedorn, Liu, et al., 2023 demonstrated the use of these container concepts for two infrastructure-project use cases: bundling images containing damages of bridges, IFC model of the bridge and the placement of the damage within the IFC; bundling pavement IFC models with relational database containing road maintenance information. These use cases were demonstrated in a web user interface, where participants can perform Create, Read, Update, Delete (CRUD) operations. It also demonstrates SHACL for verifying these containers.

However, several limitations of the existing form of the ICDD approach have been reported. Senthilvel, Oraskari, et al., 2021 assessed that the ISO 21597 standards did not cover any definitions or adaptations for the functioning of a CDE. Additionally, it also noted that its absolute file-based approach prevented it from being used for other phases and tasks of a project beyond the handover tasks. As noted by both Pauwels et al., 2022, research so far has only demonstrated that the current implementations of ICDD containers can be categorised as an intermediary step for overcoming file-based data, to eventually progress towards a decentral graph storage of information. In other words, ICDD-structured containers should be able to live on web-based CDEs and manage heterogeneous decentralised data.

Another critical point which was not addressed in these standards or the research reviewed in this section is the question: of how to create and manage links between commonly en-

countered heterogeneous data in AEC projects. Pauwels et al., 2017 identified one reason for this gap is the missing conceptual and functional foundation of how these links would work. This thesis argues that for ICDD structured containers to function within decentralised yet connected CDEs, core functionalities and their conceptualisations have to be added.

2.7 Summary and Challenges

Radchuk et al., 2021's report listed that some of the biggest concerns in the existing standards are shown in the Table 2.2.

Topic	**Order of concerns**
Usability issues	23.1
Coherence between different standards	19.2%
Which standard to use	15.4%
Lack of appropriate standards	15.4%
Complexity of standards	11.5%
Too generic	11,5%

Table 2.2: Major concerns reported in study on BIM standards (Radchuk et al., 2021)

Based on the analysis presented in Section 2.6.1, it is apparent that owing to the different scopes of various standards, there is ambiguity regarding which standards should be employed. This stems from the fact that there is no singular comprehensive standard covering all use cases, functionalities and concepts. This is also not realistically feasible. Instead, there is a need for a framework/architecture on how each standard can complement the other, based on the specifications they deliver and use cases they were designed for. Moreover, specifications such as ICDD do not accurately represent the dynamic nature of project information flows, where data predominantly resides in `work in progress`, `shared` or `published` states.

Three approaches focused on detailed container breakdown structure, each following a different method for establishing links between files and objects within them. Where ICDD required a 3-stage linking approach, BIM-LV specified direct links by referring to the GUID of the link elements in

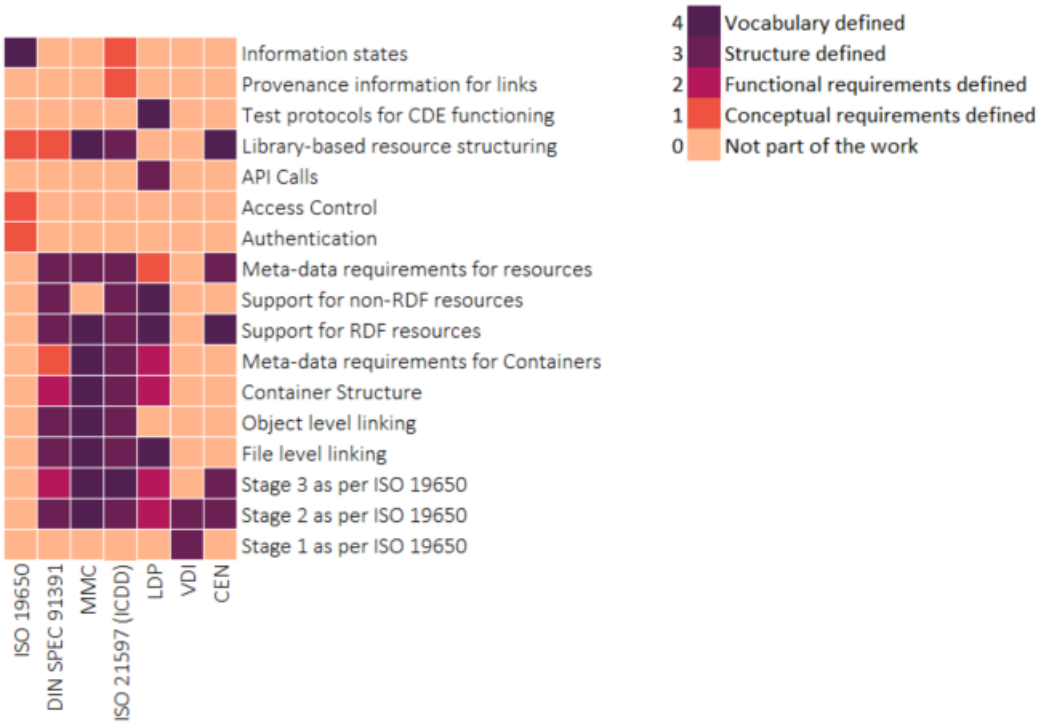

Figure 2.8: Comparison of scope of different standards and approaches reviewed

question. Although ISO 19650 and LDP provide conceptual baselines for establishing a CDE's functions, and behaviour in a CDE respectively, they are too generic. In order to be applied to the use cases within AEC, such as collection of as-build data for progress monitoring, further conceptual developments are necessary. These range from structuring the container, to vocabularies for defining metadata, etc.

However, a significant challenge is that the capabilities of the so-called BIM Stage 3 CDEs are not yet fully defined. In summary, the challenges are twofold: 1) existing commercial solutions do not support the management of heterogeneous linked information throughout the asset lifecycle, and 2) no stand-alone existing standard can be used for addressing the linking of heterogeneous information.

As postulated in Chapter 1 and this chapter, Linked Data (LD) technologies can be effectively employed for managing the linking of heterogeneous information. Since concepts from different standards overlap significantly, they can be stitched together and combined with LD technologies to conceptualise information containers which can comply with the BIM stages 2 and 3.

Stacking ISO 19650 against DIN SPEC 91391, it is evident that the former presents a broad vision of containers and

their role in a CDE. However, the DIN SPEC focuses on the practical elements of metadata for both containers and the information within them, their functionalities for managing linked data, etc. To build a functional container which can function in BIM Stage 3, the following is needed:

- the CDE functionalities defined in the LDP and ISO 19650, and
- the container structure from ISO 21597 and DIN SPEC, MMC and LDP,

Fig. 2.8 shows a map of the features identified in the standards and specifications reviewed in the previous sections 2.6.1 and 2.6.2. This scale employs a qualitative range from 0 to 4, where 0 represents the absence of feature in the reviewed material, and 4 indicates the presence of a well-defined structure for a feature. Further details on these levels can be found in the accompanying figure's legend.

2.8 Conclusions

This chapter focused on the analysis of relevant background for this thesis i.e. the national and international standards and specifications, research literature, and commercial software solutions. As part of this analysis, these sources were compared and the research gaps were summarised and mapped to the research questions and hypotheses defined in the Chapter 1. Additionally, concepts in the Semantic Web domain relevant to the upcoming sections like link discovery in knowledge graphs, and in other heterogeneous data, and management of such discovered links were also introduced. These will serve as the foundation for Chapters 4, 5 and 6.

Chapter 3

Research Design

The previous chapter laid the groundwork by establishing the research gap that is addressed in this thesis. With the research questions already identified in Section 1.4, the corresponding research methodology adopted for answering these questions is presented in this chapter. Additionally, the requirements considered for each development phase for these research questions are also described. These requirements stem from the extensive review conducted in Chapter 2 and also serve as foundations for the upcoming chapters. In addition, the implementation of the proposed concepts is also discussed. The above points are summarised in the conclusion section.

3.1 Research methodology

Chapter 1 formulated and introduced a three-part research question based on the gaps established in the domain of collaboration and managing interlinked information management. These questions broadly attempt to tackle how the information should be structured and linked, how can these linked information be stored in CDEs, and finally how they can be verified and checked.

Each of these topics was the focus of dedicated sections in Chapter 2, delving into the state-of-art research covering existing standards, approaches, literature and commercial solutions - all of which were used for identifying current challenges in addressing these research questions. This is distilled and mapped concisely in tabular format in Table 3.1. Additionally, it also designates the relevant hypothesis applicable to and explored for these questions.

Challenges	Research Question	Hypothesis
Creating links between heterogeneous data, metadata for created links, Linking structure	RQ1	H1, H2
IC Functionalities supporting linking, Data and Process Architecture for this IC and CDE	RQ2	H2, H4
Verification of RDF metadata of incoming data, Links generated in the CDE, Conformance of containers to existing standards	RQ3	H3, H4, H5

Table 3.1: Challenges mapped to the Research Questions and Hypotheses

Due to the cross-disciplinary span of a built asset's lifecycle and this thesis' limited scope, use cases which represent the crucial challenges of typical heterogeneous information management were identified as one of the requirements. For this purpose, the research project BIM4Ren was chosen.

3.1.1 Casestudy: BIM4Ren

[19] https://bim4ren.eu/

BIM for Renovation (BIM4Ren)[19] was a H2020 project that involved a consortium of 23 partners spread across the European Union and focused on the exploitation of BIM for the energy-based renovation of the existing building stock for the entire renovation phase. Within such renovation projects, various heterogeneous data are collected such as partial models of the buildings, legacy drawings of site plans, point clouds and photographs captured during initial explorations before the start of renovation, and its associated energy and material-assessment reports. These data will then be leveraged by different participants as inputs for evaluations, analysis, design tasks, etc. in various stages of the renovation project (Sainz, 2022). Consequently, not only should these data be accessible and processable by different tools of each participant, but they should also be interconnected so that these relationships can be used by participants for rapid decision-making.

Initial research identified crucial issues such as which can be overcome using BIM. These ranged from multidisciplinary collaboration, non-captured knowledge and its resulting unforeseen consequences, and non-precise modelling data (Comission, 2018). During the course of the project, a workflow

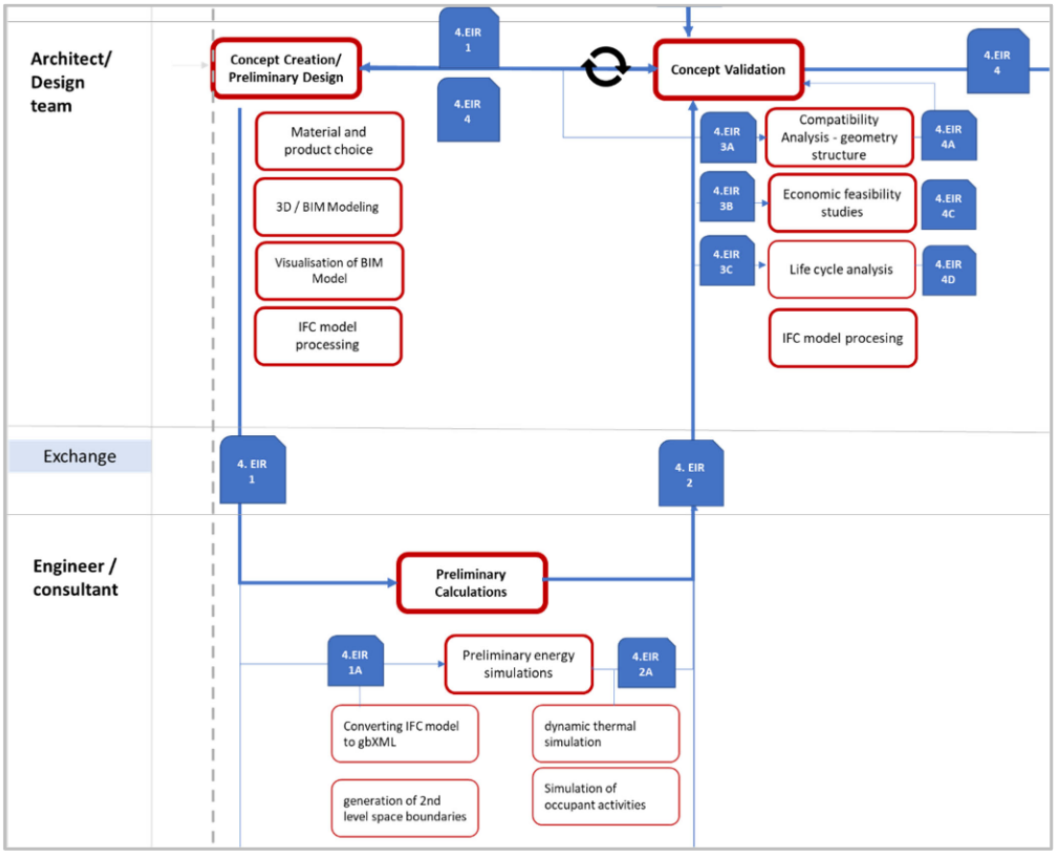

Figure 3.1: Snippet of a BPMN workflow capturing the information flows and their corresponding EIRs for the conceptual design phase (Armijo et al., 2021; Werbrouck, Tarkiewicz, et al., 2019)

was developed to allow various tools to function within a so-called *One Stop Access Platform* One Stop Access Platform (OSAP).

Since the project focused on the improvement of efficiency in renovation projects through digitalisation, it aimed to develop a platform where various tools can be used to streamline data processing based on renovation-specific tasks. The renovation project-specific tasks like compliance checks to national codes, re-evaluations of energy performance, and renovation scenario suggestor, used input data stored within the platform.

The platform thus developed was tested on three pilot renovation project sites (located in France, Spain and Italy), which were used as *Living Labs*. These labs used a series of feedback capture mechanisms to iterate and validate conceptual solutions and prototypes developed within the project. Feedback included surveys, interviews, and workshops conducted among project participants and relevant external par-

ticipants.

The OSAP developed in BIM4Ren was based on the information flow requirements established through grassroots research. participant surveys were used to gather challenges in data collection, processes, and information requirements and developed BPMN[20] workflows which captured both the generic process and EIRs[21] for use cases from the conceptual design phase (Armijo et al., 2021; Werbrouck, Tarkiewicz, et al., 2019). The above workflows served as input to define

[20] A type of flow chart method for modelling the sequential steps of an end-to-end process. The graphical notation facilitates understanding and communication of procedures using a set of standardised diagramming conventions

[21] EIR document(s) provide guidelines and specifications concerning the requirements for information to be delivered at the handover phase of the project. It often includes the acceptance criteria for information models, documents, processes and manuals.

Exchange name	4.ER 4A CONCEPT VALIDATION REPORT: Compatibility Report: geometry structure
BPMN phase involved	Conceptual Design
BPMN tasks involved	# Concept Validation / Model validation, IFC model processing
External Data (ED)	-
Sending Actor	Architect / Engineer
Receiving Actor(s)	Architect
Possible Tools	BIMserver, BIMserver Model Checker, mvdXMLChecker, *Triple Store (red.)*
Description Exchanged Data	• geometric attributes, dimensions • structure, elements, properties • project phases, use cases • validation rules (see Input 1.ER1) • Collisions, errors • LOD / LOG, LOI 200
Exchange Models	Architectural Model, HVAC Model, Environmental model, technical reports, checking rules
Data Exchange	IFC 2x3 and IFC4, XLS, text/csv etc., mvdXML, BCF 2.0

Figure 3.2: Example of EIR defined for the task *Concept Validation* showing the data exchange between an Architect and an Engineer in the conceptual design phase (Werbrouck, Tarkiewicz, et al., 2019)

the toolchain platform. This platform (called the One-Stop Access Platform) functioned as a rudimentary CDE (roughly BIM stage 1 maturity), storing data which is accessible by all participants and also validating it. However, it did not offer functionalities for interconnecting these data at both file and object levels, querying it, assigning container states, etc.

In particular, the OSAP does not conform to the ISO 19650, DIN SPEC 91391 standards for accommodating federated models, or data management using information containers. This thesis also uses the same BPMN workflows as one of the use cases for the conceptualisation of a CDE, which supports data interconnection at the lowermost object level and conforms to applicable standards and to key criteria identified in subsequent chapters.

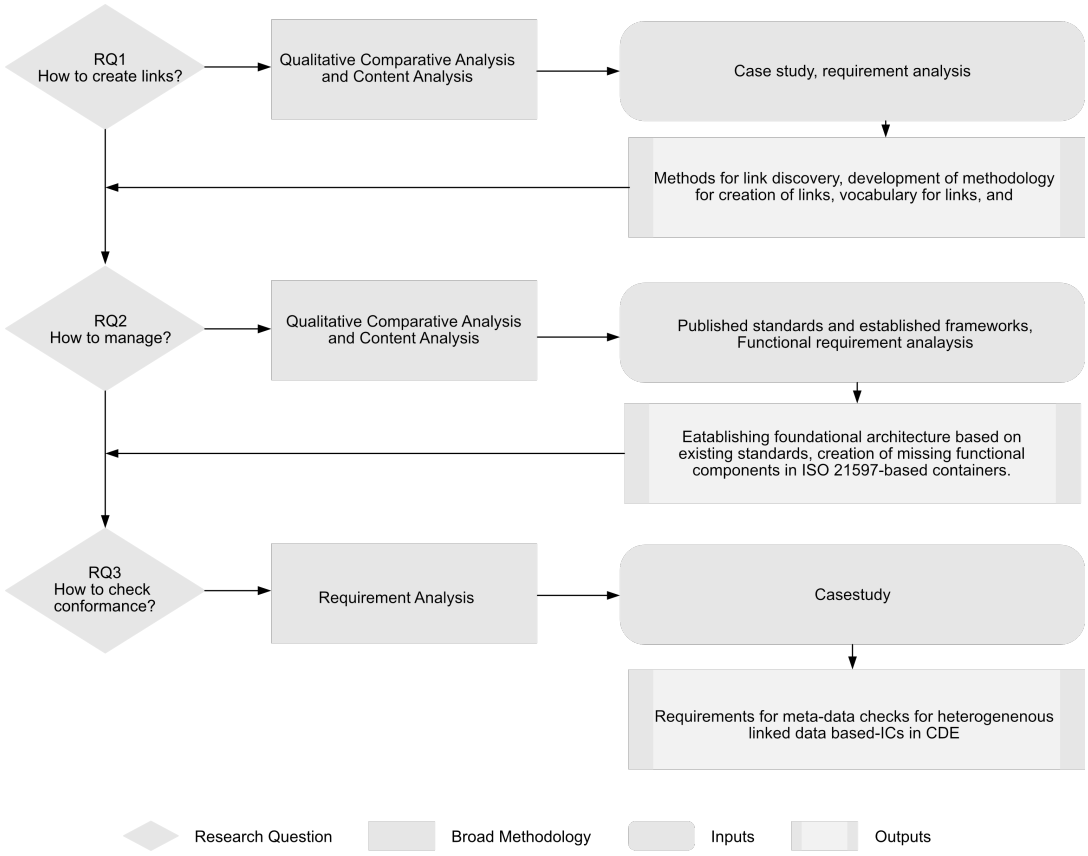

Figure 3.3: Overall research methodology

An example BPMN workflow captured from this research is shown in Fig. 3.1, where three project participants: the *Client*, an *Architect*, and an *Engineer* exchange information, based on agreed EIRs. While these requirement specifications contain how these exchanges should occur during the delivery phase, in real life, project participants exchange data continuously through all phases of a project. For example, the complexity and frequency of these exchanges are shown in Fig. 3.1, where EIRs are sent often multiple times for one task (between preliminary design and concept validation). Consequently, the vast amount of EIRs generated in these situations have to conform to ad-hoc agreements between project partners depending on their own software tools being used, and lessons learned iteratively during the course of the project. It is evident that this setup becomes extremely error-prone exchange and loss of data without establishing data-centric minimum requirements, thereby contributing to delays in project activities.

To extract the essence of the tools and the data they are used to model, further research explored the use of these requirements for all tasks for each phase (Werbrouck, Tarkiewicz, et

al., 2019). An example of a detailed EIR for the task of *Concept Validation* (see Fig. 3.1) is shown in Fig. 3.2. In this exchange between an *Engineer* and an *Architect*, diverse authoring software is used to create and process architectural, HVAC, and energy models together with technical reports, and rule sets for data validation.

It can also be observed that the data being exchanged are modelled in varying underlying schemata. For example, the BIM model is being exchanged in both IFC2x3 and IFC4 schema versions. This is an important detail, since data being modelled (even non-BIM data like images, documents etc.) are often saved in varying versions in the same tool, which potentially leads to incompatibility or interoperability issues for another project participant accessing the data in a different tool version. Hence, it is imperative that data is connectable in a standardised approach for effective project management, regardless of its representation format or unnecessary conversion.

With the backdrop of the BIM4Ren as this research's use case, the research questions can then be split into their own sub-topics: i.e.

- discovery of links in data (Chapter 4),
- management of links and data in a CDE (Chapter 5) and
- conformance of data to required schemata (Chapter 6).

Fig. 3.3 presents an overall mixed-method approach adopted for each topic of this investigation. To answer RQ1, a combination of literature review-based criteria identification, and development of a minimal metadata requirement framework was selected. It includes the development of a dedicated ontology for capturing information about links discovered in the project data. It also defined the structure for serializing these links so they can be queried, retrieved, and modified. On the other hand, for RQ2 a functional requirement analysis is used for qualitative, comparative content analysis and mapping. Finally, an internal requirement analysis methodology derived from the results of RQ1 was chosen to address RQ3. The following chapters explain each of these methodologies, a brief state-of-the-art analysis to determine the most suitable methodology/algorithm, and the conceptual developments which were deemed necessary to fulfill these questions.

3.2 Implementation and prototype

To demonstrate the concepts developed in the following chapters, a prototype was developed for each of the research questions in Fig. 3.3. Due to the complexity of development, a Micro Services Architecture (MSA)[22] was adopted for implementation. These architectures, also called bot (hereafter used interchangeably), were first proposed for the AEC industry by van Berlo et al. in (van Berlo et al., 2016). Bots are bespoke, standalone applications that run on their own servers, independent of another platform with which it works. It takes advantage of the currently fragmented data management approach by being able to use inputs from a particular source, perform a task, and generate an output which another application can use.

[22] A type of architecture style where a large application is split into small, modular and independent parts, each performing its part within the larger scope of work

Bots have an inherent composable architecture; i.e. as long as the input formats match with the bot's configuration, it can function independently of the source software feeding this input to it, allowing the different bots to be combined to form higher-order workflows. Consequently, data flows between bots and associated APIs have to be standardised to facilitate this seamless automated information exchange ecosystem.

As a part of the BIM4Ren project, van Berlo et al., 2020 also explored the possible ways of implementing a BIM bot by suggesting three types of ecosystems based on how the bots interact with an application. Fig. 3.4 sketches the MSA architecture based on the above research, where the link discovery and data validation bots exchange data with CDE, to orchestrate information management.

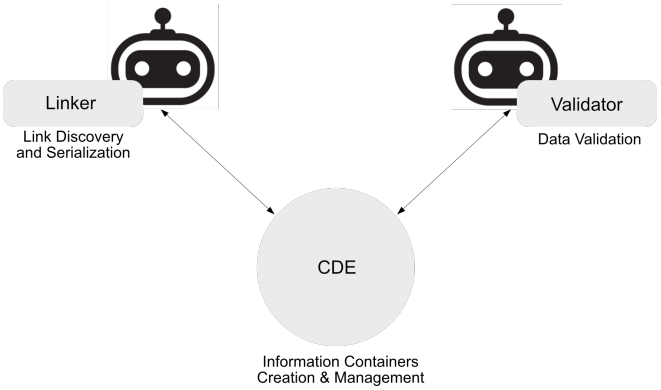

Figure 3.4: Microservices architecture for the implementation of prototypes in this thesis

Due to the separation of each module into its own self-contained unit of software, and the flexibility of calling these

bots anywhere they can be utilised, this microservices approach was also adopted for implementing the three prototypes in this research. Further details, such as the relevant libraries used for the implementation of each part, are explained in the respective chapter.

3.3 Summary and Conclusion

This chapter explained the macro-level research methodology adopted for addressing the research questions posed in Chapter 1. It identified the overarching themes for each of these questions and mapped them to related hypotheses from both the AEC and the SW domains. It also introduced the BIM4Ren project as an exemplary case study and its relevance to this research investigation. It identified points of interest in terms of data and concepts and established the background for pursuing the investigation using this project's data.

For each of the research questions posed earlier, a brief overview of the approach adopted was elaborated. Finally, an implementation architecture was proposed where the solutions developed for these questions individually, were translated into bots based on a microservice architecture. The next chapters address these questions in detail by focusing on the methodology adopted for research, the resulting conceptualisations and the prototypes developed.

Chapter 4

Linking Information

It is widely accepted that AEC projects require extensive collaboration among a constellation of participants from different domains, belonging to various organisations to deliver a physical infrastructure. The ultimate objective of this collaboration is to complete the project within the planned cost and time. While this goal is determined early on, the specific steps, requisite data and processes to achieve it are not always fully developed at the start of the project.

In addition, due to the dynamic nature of these projects, it is crucial that all data are interconnected and maintained throughout all project phases. This enables prompt decision-making to ensure that the project stays on track and remains within its planned time and resources.

Nonetheless, the AEC industry is well recognised for its ingrained reliance on document-centric information models and the inherent complexity that accompanies it. Although there are ongoing debates about the subjective or objective nature of these complexities, it is undeniable that they greatly affect the success of projects (Casti, 1994; Wood et al., 2013).

"The difficult part, the use of the exchanged information in the receiving tasks remains largely unsolved: it requires human interpretation and manual work."

— (Törmä, 2013)

Of the numerous complexities in AEC projects, complexity in information is highly relevant for this research and has been the subject of countless research over the last few decades. These complexities span a wide spectrum - from design, contractual/legal to organisational, technological etc.; and they have been explored and their influence on information management acknowledged (Azhar, 2011; Celoza et al., 2023; Hosamo et al., 2022; Pektaş & Pultar, 2006).

Interestingly, though complexities can be thematically classified based on their use case, their impact bleeds into other domains as well. For example, Pektaş and Pultar, 2006

demonstrated a framework using design structure matrix for managing the level of information modelling during the design phase between different domain experts.

However, in a federated information model, any error in an architectural model can make the entire project information model invalid, since it is used as a reference. In these cases, design complexity creates contractual complexity, resulting in a complicated scenario where the party liable for the error is not easily determinable.

From a project's perspective, it is vital that the heterogeneous data generated through each phase are structured and linked, thereby making them queriable and accessible to all participants. A major challenge here is that building elements/objects often have multiple parallel, non-explicit representations which encompass design intent, geometric information, product features, planning baselines, etc. Though these representations are heterogeneous in their data modelling (and routinely also in their file format), they describe the same physical entity. An example is given below.

> *'In a typical renovation project, the architect recreates a 3D BIM model which is then sent to the structural engineer for analysis and redesign of building elements. The same architect model is also shared with the HVAC engineer, for energy simulation.'*

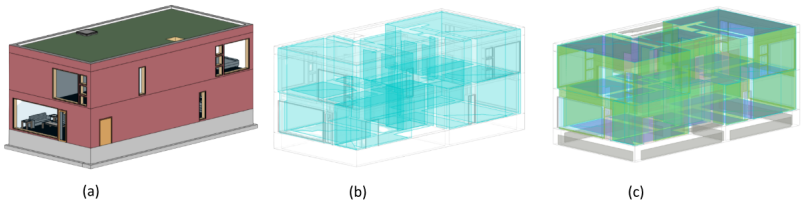

Figure 4.1: Representations of different partial models of the Duplex building: (a) Architectural model; (b) Spaces and Zones; (c) Energy Simulation model

The above example is illustrated in Fig. 4.1 using the Duplex building[23]. Here, the architectural model (a) contains design information like the type of walls, floors, etc. This model is used as an input for first creating the spaces and zones (see (b)) and subsequently calculating the energy simulations of the as-is building elements (see (c)). Subsequently, this energy simulation model will be iterated throughout the project for different materials for doors, windows, insulation, etc., to conform to the project requirements.

[23]The Duplex building is a two-storey apartment model which is well-known in the IFC community and is considered a baseline example dataset and contains disparate partial 3D models from different participants along with documentation like product data, handover information etc.

The same architectural model also serves as the input for the structural design by the structural design team. They use the outlines of the columns, beams, and slab elements for detailing the diameter of the reinforcement bars and stirrups, and their spacing (see Fig. 4.2).

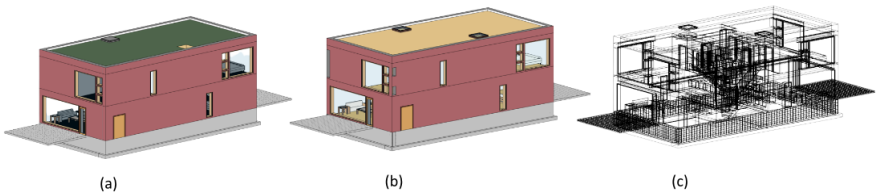

Figure 4.2: Representations of different partial models of the Duplex building: (a) Architectural model; (b) Structural model; (c)Reinforcement Detail

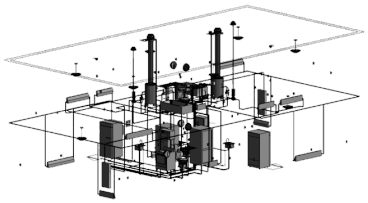

Figure 4.3: The MEP partial model of the Duplex building

Software tools like Navisworks[1] provide options for federation where native models from different BIM authoring tools can be imported and result in an aggregated model. However, though the end-user can access this federated model, upon export, it does not retain explicit links between each partial model. Furthermore, the above set of partial models are often only loosely federated, i.e. they do not contain any explicit link relationships between them.

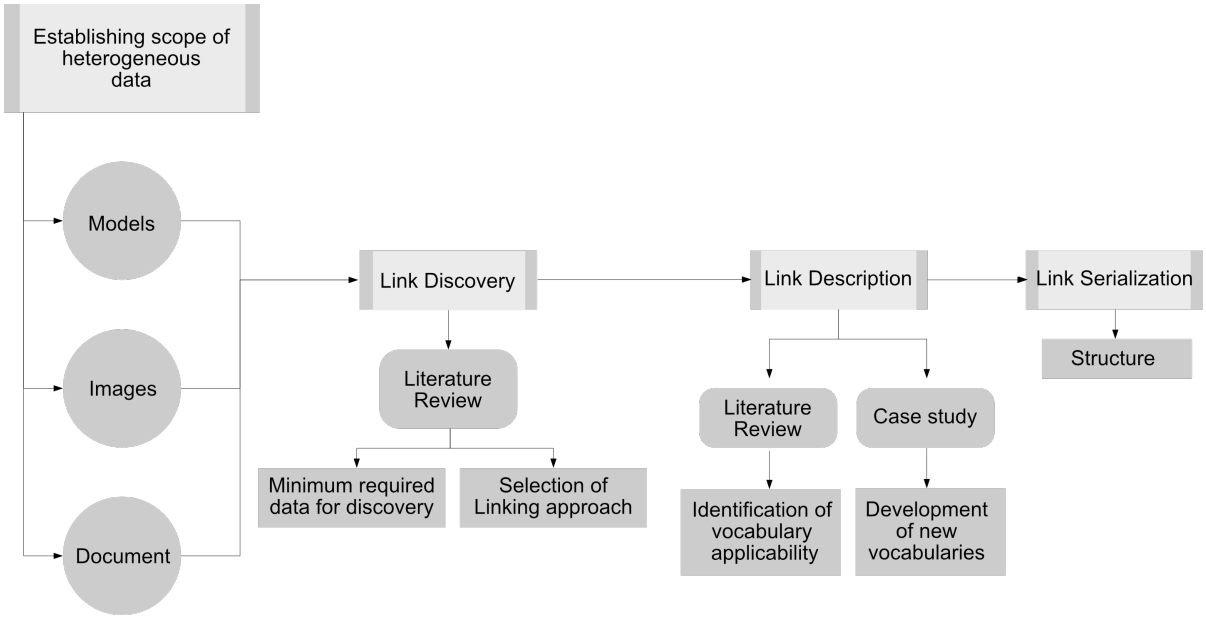

Figure 4.4: Research methodology adopted for answering RQ1

[1] https://www.autodesk.com/support/technical/article/caas/tsarticles/ts/1jiMmPap2x5i2kaHKJSZCz.html

Furthermore, how these parallel representations of information are combined is completely dependent on commercially available software and its features, in addition to the vendor lock-ins that come with it. This affects the reusability of the federated model, particularly when it needs to be linked to heterogeneous non-BIM data like PDF documents or images.

The issue of linking information has previously been investigated previously - the decentralised linked data web which conceptualises interlinked RDF graph (Pauwels, 2014). Before this, a concept by Scherer et al., 2012 and Törmä, 2013 proposed '*linksets*' which are used to represent link relationships between models. However, as Pauwels et al., 2017 adeptly highlights, these 'links' are still managed through human intervention, and include laborious tasks like the creation of links.

From a theoretical standpoint, heterogeneous data representations can be fused together using a combination of methods from ontology matching and integration using concept taxonomies, relations, axioms, etc. However, unlike the ontology and schema-level mapping required for ontology matching, heterogeneous representations will also have to be matched at the instance level (Törmä, 2013). Additionally, requirements for the above mappings, in terms of metadata, their representation, and structuring, will also have to be resolved.

To address the questions posed above, this chapter follows the research approach shown in Fig. 4.4.

Due to the quantum of work involved in considering every piece of information within an AEC project, this thesis focuses only on heterogeneous data limited to only models (based on IFC schema, other Linked Building Data (LBD) schema, Green Building XML (gbXML) schema), raster images, documents (PDF). These three types of data are first assessed to discover potential links between them. The methodology and algorithm for detecting these links is selected from the existing literature through a simplified literature review. This review is also used for establishing the minimum metadata required for the discovery of potential linkable entities.

Albeit variations of definitions for the terminologies *information* and *data* available, within the context of this thesis, they are used synonymously as the semantic interpretations of these are ignored for easy comprehension

Section 4.2 presents a method for identifying interconnected information using linguistic similarities of concepts stored in the metadata of the resources. The discovered links are then described using a set of dedicated ontologies, which were selected from publicly available open-source vocabularies. These are identified based on the metadata requirements identified in Section 4.1. In instances where the existing on-

tologies do not adequately cover all required metadata, supplementary ontologies were developed to bridge any gaps. The linkable elements and their relationships are stored in a container structure framework inspired by the current ICDD structure. This is elaborated in Section 4.3.

4.1 Link discovery

Link discovery is a crucial part of the management of interlinked data. Called by various terms like 'link discovery', 'Knowledge discovery', 'Pattern Structures', etc., this process is central to semantic enrichment.

In principle, there exists no all-encompassing approach within the intersections of the Semantic Web and AEC domains, where every category of project-related heterogeneous information can be effectively linked to each other (Pauwels et al., 2017). In this context, it is emphasised that the linking should incorporate the peripheral (surface-level) file-to-file linking and also the much deeper object-level linking. Object-level links are essential when an actual exchange of partial models (for example requirement model, architectural model, MEP model) takes place between two project participants such as an architect and the HVAC engineer.

Here, the key challenge is to find out which elements in these diverse partial models are actually identical and refer to the same abstract concept, but contain different representations (e.g. diverse representations for the abstraction `Wall` across the disciplines of architectural, structural, and energy modelling).

In the Semantic Web domain, recommender systems have been used to identify commonalities between homogeneous data (Bendouch et al., 2023; Musto et al., 2017; Peis et al., 2008). These systems detect potential relationships between images, knowledge graphs, documents etc. To accomplish this they rely on ontologies which define conceptual similarities and their relationships, thus helping these systems to derive implicit relationships between thematically overlapping resources.

In addition to ontology-based reasoning, these systems also leverage semantic similarity metrics that capture nuanced relationships between terms and contextual information such as time, or location. Although semantic recommender systems are effective in identifying link relationships between different resources, it is important to acknowledge that each

system is tailored to specific types of datasets. For instance, Bendouch et al., 2023's recommendation system focuses on utilising feature recognition for image recommendations. In other words, these systems are not equipped to handle heterogeneous data.

However, mechanisms for the identification of similarities in these systems can also be adopted for heterogeneous data-based use cases in the AEC domain. Ultimately, the linking of heterogeneous information is expected to improve interoperability between participants or even decentralised systems.

According to Musto et al., 2017 *Recommender systems* can be classified into two groups:

- content-based systems - which provide recommendations based on direct similarity and
- graph-based systems - which link user nodes to tailored recommendations.

Petrova et al., 2019 assessed different approaches for these systems and classified them into semantically-aware systems - which use linked data and ontologies for content disambiguation, linked-data powered user-centred recommendations using semantic similarity calculations, and user-profiling-based systems - which leverage previous interactions, social relations, likes, etc. for matching the user's profile with maximum similarity for recommendations. Some of the recommender systems relevant for this research are discussed in Sections 4.1.2 and 4.1.3.

Conversely, inspirations for the creation of links can also be found in the domain of ontology alignment, where similarities in the ontology structures and linguistics (including synonyms, relational aspects, etc.). Creating semantically rich BIM has been explored within the AEC domain in many research works. El-Gohary and El-Diraby, 2010 proposed an ontology integrator that uses a heuristic approach of merging taxonomies, relationships, and axioms for ontology merging. There exists significant published literature of ongoing research on the topic from the Semantic Web domain (Bloem & De Vries, 2014; Paris et al., 2019; Rettinger et al., 2012; Volz et al., 2009).

Within this research's scope, the starting point for matching heterogeneous data are similarity matching aspects. In the next subsections, the metadata required for facilitating the above matching is discussed for each type of data considered in this thesis' scope. They also elaborate on how these meta-

data are created using existing tools and approaches. This is followed by describing how link discovery is employed for metadata of heterogeneous data.

This section gives an in-depth technical overview of the metadata and approaches required for using these recommender systems.

4.1.1 Metadata types and ontologies

Metadata forms an influential part of link discovery. Every piece of information (i.e. images, partial models, point clouds, documents, etc.) can be processed to extract the underlying conceptual themes contained in it. Thematic abstractions can be captured using metadata[24]. The primary goal of incorporating metadata in knowledge graphs is to enable the end-user to query and retrieve contextually relevant information which are connected through common or related metadata. According to the Digital Library Federation (DLF)[25], metadata can be broadly categorised into the following:

- **Descriptive Metadata** which contains descriptions about the object of interest, e.g., what the object consists of or is about.
- **Administrative (or Syntactic) Metadata** which indicates the history and authorship of the object of interest, i.e. the who, what, where and how aspects. This type usually describes non-contextual information, e.g. document size, location, date of creation/modification, copyright permissions, etc.
- **Structural Metadata** links the object of interest to the overarching reference. It describes how an object and the components within it are structured and related.

This thesis focuses on *descriptive*, *administrative*, and partially on *structural* metadata for the discovery of potential links.

Ontologies designed to represent metadata commonly incorporate a shared semantic framework to describe concepts and objects (Greenberg et al., 2003). Their goal is to streamline and structure the complexities of contextual information. Well-known and widely used descriptive metadata languages include annotation constructs in RDF and OWL and dedicated metadata vocabularies like DublinCore[26]and Schema.org

[24]The word "meta" (which in Greek can mean between, after, later), when used in combination with "data" denotes "data about data". Metadata identifies and describes other data

[25]https://www.diglib.org/what-is-metadata-assessment/

[26]https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

[27] https://schema.org/docs/full.html

[28] Ontologies which are applicable across a broad spectrum of domain ontologies

[29] Ontologies containing concepts relevant for a specific domain

[27], etc. These *foundation ontologies*[28] were developed for general applicability, beyond any specific domain.

In the following sections, these are utilised to capture metadata and combined with *domain ontologies*[29] to describe annotations in raster images, documents and links discovered in heterogeneous data.

4.1.2 Raster Images

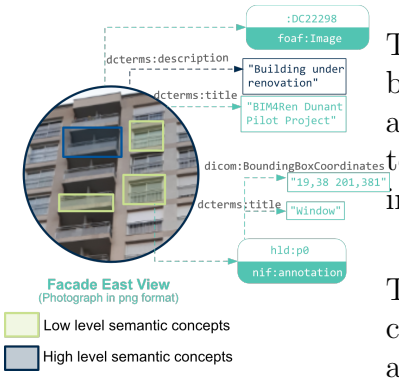

Figure 4.5: Two types of image annotation using an image from the BIM4Ren project

Typically, images captured using any device include specific built-in metadata, e.g. file size, name, format, creator, creation date, etc. However, they lack semantic data pertaining to the objects depicted within. To semantically enrich these images, object detection is commonly utilised for annotation.

These annotations can be created manually, semiautomatically, or automatically using numerous existing algorithms and approaches in the field of image processing (Cheng et al., 2018; Torralba et al., 2010; Yang et al., 2005). In manual annotation, segments of the image are manually labelled based on predefined categories/vocabulary from an ontology. Automated approaches involve varying levels of object recognition.

Regardless of the approach adopted, the resulting annotated images may contain areas of interest, corresponding titles, and descriptions along with author information, e.g. the name, organisation, description, and its provenance information.

Semantic annotations on the Semantic Web are classified into two levels (Khan, 2007):

- **low-level semantic concepts**: which contain conceptual/atomic descriptions of specific objects or image segment descriptions. E.g. a wall, window etc.

- **high-level semantic concepts**: which contain descriptions of the environment along with the specific object. E.g. an image containing walls, windows, floor and a door can be described as a room in a building.

Ontology	Prefix	Namespace
Annotation Ontology	AO	http://purl.org/ao/
Friend of a Friend	FOAF	http://xmlns.com/foaf/0.1/
SKOS Ontology	SKOS	http://www.w3.org/2004/02/skos/core
DBpedia Ontology	DBpedia	http://dbpedia.org/ontology/
Quantities, Units, Dimensions and Data types	qudt	http://qudt.org/schema/qudt#
Schema Ontology	schema	http://schema.org/Product#
Lightweight Image Ontology	lio	https://imagesnippets.com/lio/lio.owl
Exif Ontology	exif	http://www.kanzaki.com/ns/exif#
VRA Core	vra	http://simile.mit.edu/2003/10/ontologies/vraCore3
IMGpedia Ontology	imo	http://imgpedia.dcc.uchile.cl/ontology#
Arpenteur Ontology	arp	http://www.arpenteur.org/ontology/Arpenteur.owl#
DICOM	dicom	https://www.ebi.ac.uk/ols/ontologies/dicom
NIF 2.0 Core Ontology	nif	https://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html#

Table 4.1: Ontologies for image annotation

For example, in Figure 4.5, the image within the blue circle contains an image of a building, with doors, windows, balconies, walls, and the sky as the background. All of these objects are annotated as concepts and they belong to low-level semantics as they describe the lowest atomic level possible. However, if we describe this image as a renovation site, it would fall into high-level semantics, as it describes the environment in which all of the atomic concepts outlined previously are contained.

Currently, there are numerous tools that can be used for manual annotation, like the medical image annotation tool by (Rubin et al., 2009), the platform-independent tool *PhotoStuff*, which allows users to annotate regions of an image

based on the concepts it contains using preloaded ontologies (Halaschek-Wiener et al., 2005). The end goal of annotation is to ensure that semantic information relevant to the project is detected and stored. Irrespective of the techniques/tools used for image annotation and classification, a generic workflow approach for this annotation is shown in Fig. 4.6.

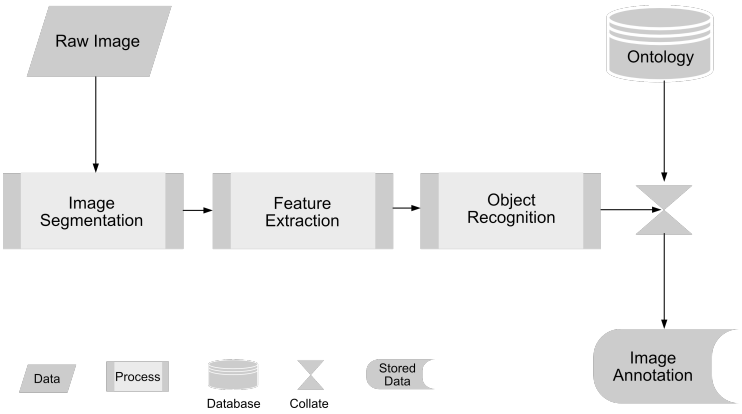

Figure 4.6: Generic image annotation workflow based on (Halaschek-Wiener et al., 2005; Khan, 2007)

In this research, it is assumed that images are already annotated and possess some form of annotation information prior to the process of link discovery. For illustration purposes, the AISO research tool for biological image labelling was used to annotate image segments and assign them appropriate concepts[30]. Despite references to ontologies specifically designed for image annotation[31], many are no longer maintained or not publically accessible. One possible explanation for this could be that many of these ontologies were developed during research projects and were subsequently archived once the projects ended.

[30] https://jaiswallab.cgrb.oregonstate.edu/software/AISO

[31] https://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/#vocabularies

Some of the well-known foundational ontologies which support image annotation and are still publicly available are compiled in Table 4.1. These vocabularies vary significantly in their size, granularity, and formality for describing image-specific annotation concepts. Most of these contain class abstractions and data properties for denoting the type of image, format, contents, spatial location of objects of interest within the image etc. Often, vocabularies tend to define only the properties and refer to another vocabulary for filling in the value definitions. Consequently, for semantically annotating an image, multiple vocabularies are needed.

In brownfield projects such as renovation, images are usually collected with only their inherent timestamps and default image labelling. Contextual information like the image's loca-

tion in the physical site, its orientation, semantic information of the objects within the image etc. are not automatically captured. Survey teams usually collect thousands of images, which are then manually organised into relevant folders and categorised according to ad hoc logic - e.g. the date of image capture, or the broad location of the site. These contextual details can enable the spatial positioning of images within 2D plans or BIM models, providing the survey team with an efficient means to view these images (Schulz & Beetz, 2021).

Explicit links between images and other data sources, e.g. an IFC model, or drawings are not usually recorded temporarily or permanently. Usually, domain experts rely on contextual clues from the images themselves, along with textual descriptions in separate documents to understand their relevance to the task. However, these are often very subjective and never explicitly recorded anywhere.

To explicitly determine the connections between objects, images must include sufficient objective metadata. This data can be utilised by both humans and machines in order to generate accurate annotations. Within the context of AEC projects, two key types of information are crucial for this identification process:

1. image authorship and provenance (i.e. who created the image, when, and who modified it) and,
2. contextual and semantic information about the image (i.e., does it contain objects of commonality with other data sources, how was it identified, when was it identified).

Image authorship and provenance information is classified as Administrative Metadata. Contextual and semantic information can be classified as both Descriptive both Structural Metadata, depending on the level of description (either Low-level semantics or High-level semantics).

To illustrate this concept, Fig. 4.5 depicts the image of a building with multiple building objects of interest like windows, balcony doors, facades, etc. One particular window in this image can be identified in the IFC model (see highlighted section). These two data sources (image and IFC) contain many points of commonality, of which one is this window. Consequently, this window has to be explicitly identified in the image, semantically enriched (by labelling it as a window) and stored. The metadata used to semantically enrich images are listed in Table 4.2.

Prefixes for all *RDF graph* listings can be found in Appendix C

The serialisation of the above graph in `turtle` syntax is shown in Listing 4.1. In this example, the foundation ontologies e.g. the Dublin Core (`dcterms`), schema, Friend of a Friend (`foaf`), RDF (`rdf`) are used for generic metadata representation. These include properties like file name, file type etc.

The annotation is described by the predicate `nif:annotation` and it points to the first point of interest `hld:p0`. This is defined as a rectangular annotation having the coordinates `"19.38 201,301"` using `dicom:BoundingBoxCoordinates`. Furthermore, this region has been labelled as `Window`, thus indicating the semantic interpretation of the pixels in the image.

```
hld:DC22298
    a foaf:Image;
    dcterms:createdOn "2019-11-10";
    dcterms:creator hld:Person134;
    nif:annotation hld:p0;      #Annotation type 1
    omg:hasGeometry hld:p1;     #Annotation type 2
    dcterms:description "BIM4Ren Dunant Pilot
       Project";
    dcterms:title "Photograph of a building".

hld:Person134
    a schema:Person;
    schema:givenName "Aude";
    schema:lastName "DeBresson";
    schema:Organisation "Nobatek".

hld:p0                #Annotation type 1 using
   bounding box
    dicom:BoundingBoxCoordinates "9,20 17,110";
    dcterms:title "Window";
    nif:confidence 0.7;
    prov:generatedAtTime "2022-12-12T18:28:43Z"^^
       xsd:dateTime.

hld:p1               #Annotation type 2 using OMG
   Level 2
    a omg:Geometry;
    omg:hasSimpleGeometry "19 38,201 301"^^xsd:
       string;
    dcterms:title "Window";
    nif:confidence 0.5;
    prov:generatedAtTime "2022-12-12T18:08:23Z"^^
       xsd:dateTime.
```

Listing 4.1: Snippet of metadata for annotations in the image

The above representation can be scaled to capture all these annotated regions in an image, such that the image is semantically enriched. Two representations of the annotation are

shown here. The first region `p0` is defined using bounding boxes from the `dicom` ontology. [32] and with the coordinates of the region annotated and captured as literals.

The second region `p1` is defined according to the Ontology for Managing Geometry (OMG)[33] Level 2. These two ways of representation have significant consequences for overall data management. In the former, it is feasible to embed any kind of geometry description, though the AEC ontologies that use this feature are limited (e.g. BimSPARQL, RDF*, GeoSPARQL). However, OMG facilitates an agnostic link of any geometry description, and thus is not dependent on a format (Bonduel et al., 2019).

Both of the above approaches can be used in varying contexts: for example, the bounding box approach can be used when only a few annotations are present in images, thus making the metadata graph compact. However, when an image contains regions with varying annotation interpretations (e.g. a door, which was labelled *architectural door* and also *wooden door*, the OMG level 2 or level 3 approach can be adopted. In both of the above examples, the property `dcterms:title` is used to identify the theme of the annotated object, and the `nif:confidence` is used to denote the annotation's level of trustability.

[32] Well-known Text is another popular markup language geared towards representing geometric objects like points, polygons, lines etc. It uses string to represent this geometric information

[33] Originally developed as a part of the Scope Project, it introduces a three-level description for adding geometric metadata

The namespace `hld` heterogeneous linked data, the default namespace for instance data used in this thesis

Table 4.2: Collated metadata for images

Metadata classification	**Metadata**
Administrative	File name
	File type
	Author
	Author organisation
	Creation date
	version ID
Descriptive	Description
	Title
	Annotation regions
	Coordinates
	Annotation title

4.1.3 Documents

Similar to image annotation, the sections within documents can also be annotated, contextualised and referenced. These annotations (also known as classifications) can be related to the themes of both the overall document and/or specific paragraphs and lines within the document. Thematic classification is usually achieved through rule-based systems or by using machine learning models. The predominant inputs for text classification are:

- defining the type, genre or theme of the text using its context, or
- visual classification, where analysis of pixels in the image, image recognition and object recognition are employed.

Textual classification also involves techniques like Natural Language Processing (NLP) for analyzing the context through words and phrases to understand the underlying semantics. In the case of scanned documents, techniques such as Optical Character Recognition (OCR) can be used for identifying

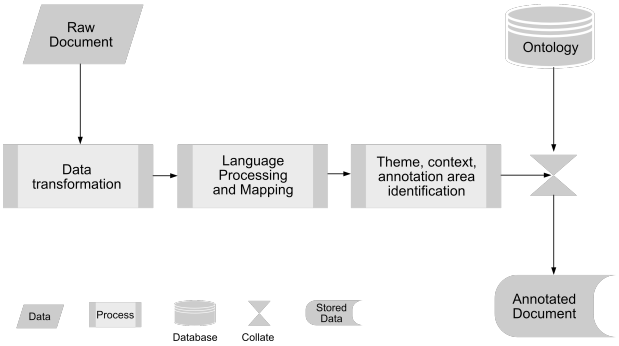

Figure 4.7: Generic document annotation workflow

words, while computer vision can be used for recognising and categorizing objects in images within a document.

Independent of the method used for the detection of themes in the paragraphs or entire documents, the identified themes should be semantically annotated so that they can be used for linking with other related information sources. These metadata are stored separately in a knowledge graph which can then be indexed and queried if they are combined with ontologies, schemata, or metadata of other data sources.

Ontology	**Prefix**	**Namespace**
Friend of a Friend	FOAF	http://xmlns.com/foaf/0.1/
SKOS Ontology	SKOS	http://www.w3.org/2004/02/skos/core
DBpedia Ontology	DBpedia	http://dbpedia.org/ontology/
nsl	nsl	http://purl.org/ontology/
ebucore	ebucore	http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#
mvco	mvco	http://purl.oclc.org/NET/mvco.owl#
Ontology Design Patterns	ontopic	http://www.ontologydesignpatterns.org/ont/dul/ontopic.owl#
Lemon Ontology	lemon	http://www.w3.org/ns/lemon/lexicog/
CSVW Namespace Vocabulary	csvw	http://www.w3.org/ns/csvw#
NIF 2.0 Core Ontology	nif	http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#

Table 4.3: Ontologies for document annotation

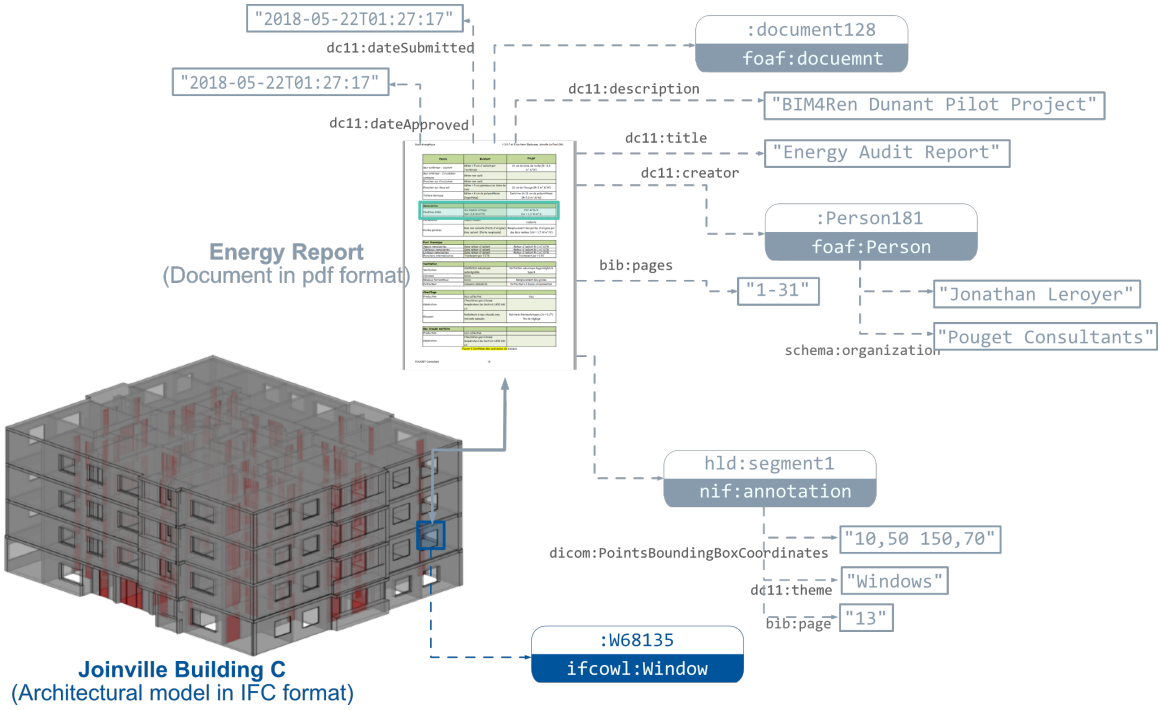

Figure 4.8: Annotated document

There are numerous tools available online commercially and from research that help with the annotation of text-based documents in different formats. These tools can be either manual, semi-automated or fully automated. For example, the PDFTab addon to Adobe Acrobat software allows users to add OWL ontologies to existing PDF documents and its subparts to specific concepts from the ontologies (Eriksson, 2007). Islam, 2015 also developed an annotation tool which performs both automated and manual labelling of tables in a PDF document based on an ontology database. This tool also supported custom annotation and enabled the user to view them in the user interface. Maderlechner et. al explored automatic semantic captioning of images embedded in pdf documents (Maderlechner et al., 2006). A generic workflow for annotation of documents based on these tools is shown in Fig. 4.7.

The format of document storage, i.e. whether it is stored as a `.pdf` or as a rich text document (i.e. `.doc` or `.docx`) does not play a major role in the efficacy of the algorithms. Eriksson, 2007 lists three possible models for adding annotations to documents: 1) adding metadata to the document, though the metadata do not relate to the document contents, 2) relating metadata to specific parts of the document contents, and 3) storage of metadata external to the document. The vocabularies used to describe these can be extracted from

some existing ontologies, like the ones listed in Table 4.3.

Fig. 4.8 illustrates an annotated document consisting of metadata describing the document like title, its description, authorship information, and annotation information. For the latter, it consists of the region of annotation, the overall topic of the text within this region, and the page number, which specifies the location of this annotated region within the document. This annotation example reuses the ontologies listed in Table 4.3.

```
hld:Document
    a foaf:Document;
    dcterms:createdOn "2018-05-22T01:37:17"^^xsd:
        dateTime;
    dcterms:creator hld:Person181;
    bib:pages "1-31";
    nif:annotation hld:segment1;
    dcterms:description "BIM4Ren Dunant Pilot
        Project";
    dcterms:title "Energy Audit Report".

hld:Person181
    a foaf:Person;
    schema:givenName "Jonathan";
    schema:lastName "Leroyer";
    schema:Organisation "Pouget Consultants".

hld:segment1
    a nif:annotation;
    nif:confidence 0.8;
    ao:hasTopic "Window Energy Efficiency";
    prov:generatedAt "2020-10-29T10:55:02"^^ xsd:
        dateTime
```

Listing 4.2: Annotation metadata for document

Note that in order to describe the text sections annotated, this example uses the predicate `nif:annotation`, similar to the image annotation example. The theme of the region annotated is represented using `ao:topic`. Additionally, a `bib:page` is also used to reference the location of the annotated text region within the document. In addition to the authorship details, it also contains two specific timestamp triples with the predicate: `dcterms:datesubmitted` and `dcterms:dateApproved`.

Considering that documents have conventionally relied on the existence of a cover sheet which records the document control process, textual documents use the above two timestamps to capture this information. Additional information such as author, organisation can also be extracted from these

coversheets and supplement the metadata generated for these resources.

4.1.4 3D Models

As stated at the beginning of this chapter, a typical construction project contains numerous partial models from various participants, each describing the design, product, construction, simulation, characteristics of components and associated spaces. One example of these models was introduced in Fig. 4.1 and 4.2, where the architectural, structural and energy models were shown to occupy the same space, yet represent different interpretations of this space through the elements they contain and the concepts they embody.

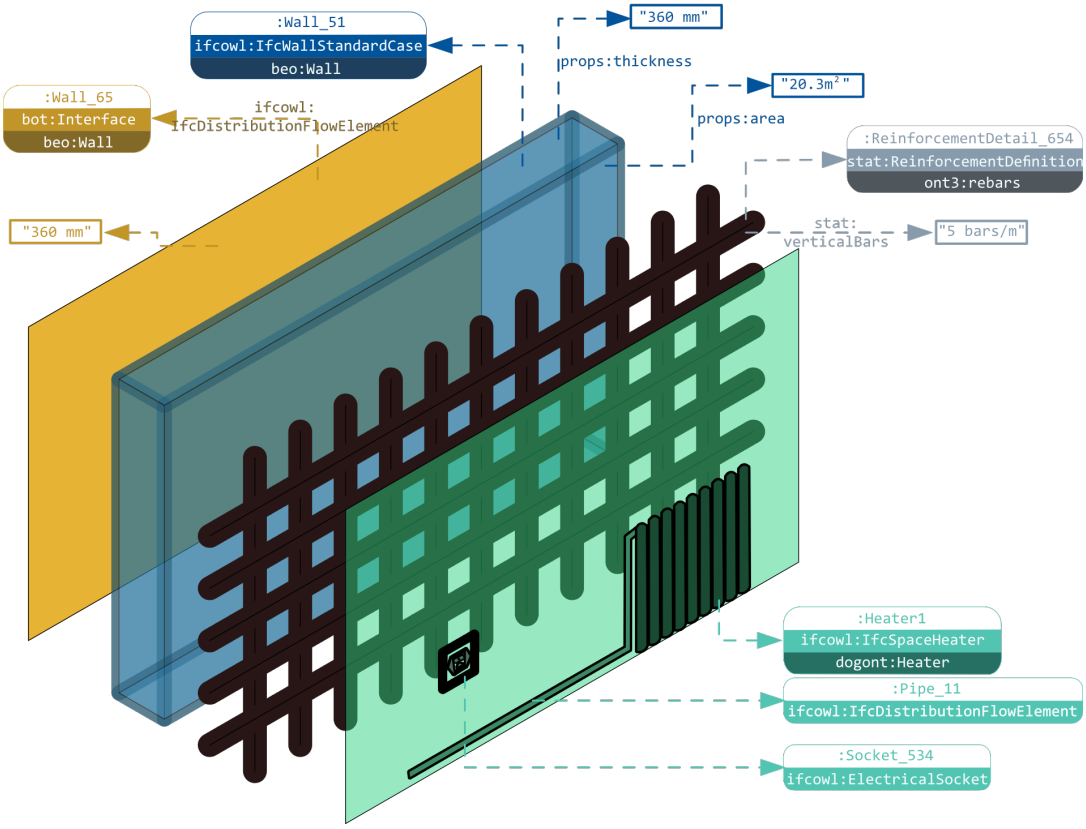

Figure 4.9: An annotated image of one Wall, in 4 different representations

A closer peek for a single building element is provided in Fig. 4.9, where the walls from the above three participants are represented visually superimposed. In this figure, four representations of a wall are shown as per their colour encoding:

- **Blue representation**: Architectural model of a wall with information on its thickness and area

- **Yellow representation**: Energy model which has a wall represented as a boundary space with the wall's thermal transmittance value
- **Black representation**: Structural model of the wall, containing the reinforcement detail
- **Green representation**: MEP and/or Heating, Ventilation, Air-Conditioning (HVAC) model of the wall, containing an electrical socket, a pipe supplying a heating medium and a space heater

The representation definitions of this wall across all these partial models are defined in their respective schemata. For example, the architectural wall is defined as an `ifcowl:IfcWallStandardCase` and as a `beo:Wall`, while the energy model defines this very wall as `bot:Interface` and a generic placeholder [34]`ont2:Element`.

[34]Linked Open Vocabularies is a published search engine endpoint containing index of registered ontologies, their classes and properties. It can be accessed in https://lov.linkeddata.es/dataset/lov/

Note here that, in the structural and the MEP model, the terminology `Wall` is absent, although the elements `reinforcementDetail_654` and `Heater1` can be understood to be present in the same space (or in the vicinity) as the wall.

[34]Placeholder definitions are used to represent classes/properties for which exact domain or foundation ontologies were not found on the LOV endpoint[34], but can potentially exist on the Semantic Web

Notably from Fig. 4.9 and the explanation above, this wall itself is not represented in the same way across their partial models, although all participants design their systems with the representation of a wall as the reference. For instance, an energy model is usually an abstract representation of the building's overall structure and layout, which is eventually used for energy simulations. The representation is structured to contain all core pathways and processes of heat transfer. Consequently, instead of solid building components e.g. floors, walls, etc., these models use spaces, surfaces, and zones. Any door, window, etc. in any of the above components are considered as air openings.

On the other hand, in architectural models, walls are modelled as solid components, with material layers and properties, geometric information including wall width, height and length. However, in structural models, only structural components are modelled. For instance, this will exclude all non-load-bearing masonry walls, glass facades etc. Components like beams, shear walls, structural columns, structural floors, foundations, and their corresponding detailed reinforcement design are also modelled. Each of these partial models are designed by domain experts for their specific activities. For example, the structural model is used by engineers to design

the reinforcement and element properties which are used for structural analysis of loads as per local/national regulations.

MEP and HVAC models tend to have only the mechanical, electrical, piping systems, and duct systems for accommodating them, along with equipment for heating, cooling and air-conditioning which are modelled in specific spaces and zones in the building model. They do not contain explicit building components like walls, doors, beams, etc. However, the above systems and equipment are spatially located and associated with these components.

A brief description of the namespaces and their vocabularies used explained in the sidenote.

- SAREF: Smart Applications ReFerence ontology focused on devices and their functions
- s4bldg: A SAREF ontology extension for buildings
- ifcowl: Based on IFC schema, a vendor-neutral interoperable schema
- stat: temporary library for reinforcement definition
- props: An ontology transforming buildingSMART PSets into OWL-based ones
- bot: An ontology for describing topological concepts about a building
- s4watr: Ontology defining properties related to water flow

```
hld1:Wall_51  #Named Graph inst1
    a ifcowl:IfcWallStandardCase, beo:Wall;
    ifcowl:globalId_IfcRoot hld1:
       GloballyUniqueId_44297;
    props:thickness "22cm";
    props:area "4.5m2".
hld1:GloballyUniqueId_44297
    a ifcowl:IfcGloballyUniqueId;
    express:hasString "2O2Fr$t4X7Zf8NOew3FKRi".

hld2:Wall_65  #Named Graph inst2
    a bot:Interface;
    s4bldg:Wall;
    props:area "4.5m2".

hld3:ReinforcementDetail_654  #Named Graph inst3
    a stat:ReinforcementDefinition, ont3:rebars;
    stat:verticalBars "5 bars/m".

hld4:Heater1  #Named Graph inst4
    a s4bldg:SpaceHeater, dogont:Heater;
```

```
    s4bldg:effectiveCapacity hld4:Value_9562;
hld4:Value_9562
    a schema:value;
    express:hasString "1.2".
hld4:Pipe_11
    a s4watr:Pipe;
    s4bldg:nominalDiameter hld4:Value_9853;
hld4:Value_9853
    a schema:value;
    express:hasString "0.5mm".
hld4:Socket_534
    a ifcowl:IfcOutlet, s4bldg:Outlet,
                        saref:Device;
    saref:hasManufacturer "Siemens DELTA".
```

Listing 4.3: Metadata for architectural, structural, energy and HVAC models

All of the above four representations will have to be linked to each other. Prior to this, the metadata used to describe them need to be assessed to discern if they are adequate for creating links.

Unlike images and documents, information within models are already based on structured data representations, usually following a particular ontology/schema. For example, common authoring software like Revit or ArchiCAD usually utilise IFC (or IFC-based) schema for modelling parametric 3D elements. Since this schema is machine-readable and contains an inherent hierarchy of objects and its associated properties, models do not need further annotations.

Listing 4.3 presents a snippet of the RDF representation of the example shown in Fig. 4.9. For brevity, the snippets of each graph are listed together above, but in reality, they can indeed be stored separately. The goal is to connect each of the components `Wall_51`, `Wall_65`, `reinforcementDetail_654`, `Heater1`, `Pipe1`,
`Socket_534` to each other, so that when a project participant queries information about `Wall_51`, all the connected information across each of the partial models are displayed.

In the next section, the listings described here are used as inputs for discovering potential links, which are then structured, and serialised in a persistent format. The composition and construction of these links in a retrievable format for other downstream tasks are also examined.

4.2 Link discovery for heterogeneous data

In the previous sections, several types of annotation for objects of interest within data sources such as images, models, and documents were presented. As mentioned in Chapter 1, AEC projects have information spread across heterogeneous data sources. While conventional link discovery methods can detect relationships, they are limited to discoveries within a particular type of data, for example, object detection in images.

To detect relationships between heterogeneous data, a combination approach must be adopted that considers all relevant features for each of the data sources. For example, Hessel et al., 2019 presented unsupervised algorithms to discover relationships between images and the corresponding sentences in documents. This approach considers both the images and the sentences to be part of the document and is not treated as separate data sources, despite their heterogeneity.

Classification	**Feature**	**Description**
Linguistic Feature	• Entity Name • Entity Documentation	• The name of the ontology entity • Short textual description of entity
Structural Feature	• Entity Hierarchy • Relations • Attributes	• Information about the entity in the hierarchy • Relations of the entity to other entities • Attributes of the entity
Constraint-based feature	• Data Type	• Data type of the entity
Integration Knowledge-based feature	• Technical Names • Default Values • Identifiers • Code Lists • Instance Data	• The technical annotation of the entity • Default value for the entity • Local or global identifiers for the entity • Possible values for the attributes of the entity • Instances associated with the entity

Table 4.4: Features for matching terminologies of partial models, based on (Martinez-Gil & Aldana-Montes, 2012)

Similar works using supervised and unsupervised approaches have also been presented in many works (Datta et al., 2019; Hu et al., 2016; Plummer et al., 2015). Most of these are based on the identification of objects in single images with descriptions in the natural language, which also has a rich

research history (Hu et al., 2016; Margffoy-Tuay et al., 2018; Plummer et al., 2015).

In the case of link discovery for partial models, the overall discovery concepts remain the same, with variations in their application. Partial models in AEC, as mentioned in Section 4.1.4 often contain elements that occupy the same spatial location, although these elements may not necessarily have the same definition.

For instance, reinforcement bars are contained within concrete walls, and hence both these elements occupy the same spatial space in a federated model. This federated model consists of the structural model and the architectural model; the former represents its elements defined by the Building Element Ontology (BEO)[36] and the latter by the IFC schema[37] or BOT Ontology[38]. Here, neither the GUIDs nor the terminology used for defining the elements are the same and thereby cannot be directly used for establishing links between the two element representations. In these cases, approaches from the domain of ontology matching can be utilised.

The annotations used in this figure follow the ontology template by (Donkers, 2022).

[36] https://pi.pauwel.be/voc/buildingelement/index-en.html

[37] https://standards.buildingsmart.org/IFC/DEV/IFC4_1/OWL/ontology.ttl

[38] https://w3c-lbd-cg.github.io/bot/

In the ontology matching domain, the alignment between two different ontologies is achieved by finding the semantic relationships between the classes contained in each of them. The resulting *mapping* is arrived at after the computation of the *similarity measure*[39], i.e., the alikeness between two class definitions. *Matchers* (or algorithms) that are used to compute this measure can be used both at the structural level and at the element level.

At the structural level, class concepts are assessed using only the internal knowledge contained in the ontology itself or prior alignments. The element level uses relationships of concepts to other external concepts or instances, which are not part of either the ontology or the alignments. When the arbitrary threshold set for this measure is met, the definitions from both these ontologies are mapped to each other, thereby establishing a mapping between them.

Matchers can exploit numerous features of an ontology (or an instance graph) for creating a mapping. Euzenat and Shvaiko, 2013 classifies these features based on their overarching approach used to detect semantic similarity. Table 4.4 provides an overview of the overarching matching techniques commonly used in general ontology matching.

Of the many types of *matchers* available based on the classification and characteristics above, the following were selected

for the above task of ontology matching, based on ease of use and availability of tools to implement them:

- *Language-based (Linguistic Feature)*: these are lexical matchers which use literal name matches from lexicons (which are dictionaries) of the partial models
- *String-based(Linguistic Feature)*: these are called mediating matches, which are similar to the lexical matcher, though it uses an additional external source (either an ontology or a mapping between ontologies)

It is essential to acknowledge that the aforementioned were selected solely for the purpose of demonstrating the fundamental concept of generating link relationships. The algorithms were not assessed for their efficiency in link discovery. Here, the focus is on the structuring of the links thus detected. They do not depend on the accuracy, precision, or recall values in the above link discovery process.

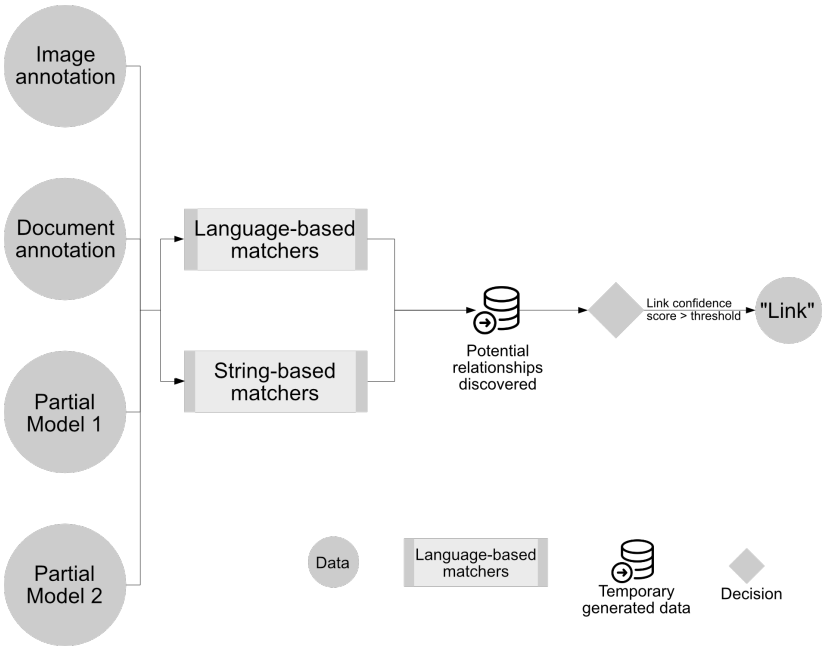

Figure 4.10: Approach adopted for discovering links in heterogeneous data

Both of the above features were implemented using dedicated Python libraries supported by WordNet(Bird et al., 2009). WordNet is a lexical database consisting of semantic relations between words that connect similar words into relations including synonyms, hyponyms[40], and meronyms[41]. The synonyms are grouped into synsets with short definitions and usage examples. Thus, it can be seen as a combination and extension of a dictionary and thesaurus.

[39] Similarity measures are often used in ontology matching domains

[40] A word which has a relationship between a generic term and a specific instance of it. E.g. brick wall, stone wall, concrete wall are all hyponyms of wal

[41] A semantic relation between a meronym term denoting a party, and a holonym term which denotes the whole. E.g. brick is a meronym of wall

Fig. 4.10 shows the approach adopted for the discovery of links using this algorithm. The *WordNet*[42] library is used for both 'Language-based matcher' and 'String-based matcher'. The potential relationships between the three data sources (partial models, image, document) discovered by these algorithms are temporarily stored and verified against an arbitrary threshold. If this threshold is met, these links are serialised as persistent metadata for further downstream applications/tasks.

Using the BIM4Ren example again, Fig. 4.11 illustrates how the building element 'window' is defined in an IFC-based architectural model, a Building Element Ontology (BEO) schema-based structural model, an image, and a PDF document. The PDF and the raster image leverage the additional annotation data created as described in Sections 4.1.2 and 4.1.3.

To express the generic linkage of the commonly related concept, a link types ontology was developed. It focuses on a two-pronged link definition for representing the commonalities in links between different representations found in the AEC domain. This ontology is elaborated in Section 4.3.1. It defines a `basicLink` which consists of only generic link types which designate that an object is `partOf/isLinkedTo/relatedTo` another object, without specifying the exact nature of the above relationship. A more exacting relationship between links is provided by the overarching main class `complexLink`, which consists of spatial and semantic link types (refer to Fig. 4.14).

In this research, the algorithm developed uses only the basic link type `isLinkedTo` to capture the links discovered. It does not differentiate between the technicalities nor identify them. To illustrate the usage of the complex link types, the discovered links are manually extended and detailed as shown in Fig. 4.11. Here, the BIM4Ren project data: image, an energy report, and two partial models were taken as input and the resulting links identified were described using the predicate `isLinkedTo`. However, a deeper classification where the specific relationship between them was identified is presented in Listing 4.4.

```
1 hld1:W68135 #Named Graph: architectural model
2     a ifcowl:Window.
3
4 hld2:O654SD #Named Graph: structural model
5     a beo:Window.
6
7 hld:DC22298 #Named Graph: Image Annotation
```

```
    a foaf:Image;
    geo:hasGeometry hld:p0.

hld:p0
    a sf:Polygon.

instD:document128  #Named Graph: Document
   Annotation
    a foaf:document;
    nif:hasAnnotation instD:segment1.

instD:segment1
    a nif:annotation.

hld1:W68135 #Level 1 link type
    blink:isLinkedTo hld2:O654SD, instD:segment1.

hld1:W68135
    blink:isConceptuallyIdenticalTo hld:p0.#Level 2
        link type
```

Listing 4.4: Usage of the BuildLinks ontology for describing types of links between heterogeneous data

In this figure, the predicate `blink:isLinkedTo` is encapsulated as it is used to describe the detected links. This means that it is used to illustrate the overarching linking approach. In reality, `blink:isLinkedTo` has to be defined and structured based on the granularity of the metadata for the link itself. This includes metadata like the algorithm used for detecting this link, the link's author, the link's confidence score, provenance information, e.g. time of creation, etc. These topics are covered in the next section.

4.3 Describing linkage semantics

In the previous section, the link discovery process was described in detail. This section elaborates on the metadata and its associated vocabulary for describing the links detected in the above processes along with the structure for storing these in an RDF graph which will eventually be used by information containers. This research considers four key aspects for the description of links: type of links, provenance, confidence score, and structured serialisation.

4.3.1 Types of links

Currently, there are no concrete vocabularies from the AEC domain which describe the types and complexity of links between the different data sources. Some of the generic foun-

dational ontologies like LinkLion, can be used, however, they are now defunct and not actively maintained (Nentwig et al., 2014). Three ontologies were identified as most relevant: the VoID ontology, LDP ontology, and ICDD ontology. It was noted that they contained classes describing links that are generic enough that they occurred in all of these vocabularies and had similar definitions.

However, the comparison yielded the observation that both VoID and LDP possess limitations on the degree of expressibility of the link relationships, especially the ones stemming from the AEC domain. For example, they do not explicitly describe the relationship between the heater in an HVAC model to the architectural model.

Of the above three, the ICDD schema proposed by ISO 21597 is the closest to defining detailed links (part 2 describing specialised link types) and focuses on link definitions and their types, but it is a domain ontology. Furthermore, the major issue with using the link types defined in ICDD is that the resources are not directly linked. Rather, they are linked based on a 3 hierarchical level, thereby making readability and graph extension cumbersome.

ICDD defines an overarching structure for links (in part 1) and is explained in Fig. 4.17. Additionally, there are certain specific specialised link types defined in part 2, which can be used in conjunction with the above structure. However, the ICDD standard is very restrictive in its usage. The standard explicitly prohibits modification of these link types and structures and their usage outside the ICDD container structure.

Hence, they were used as inspiration for the creation of a link types ontology. Due to the relevance of ICDD's link types defined in parts 1 and 2 of ISO 21597, they are also incorporated where feasible in the ontology. Fig. 4.14 shows the overview of the classes in the BuildLinks Ontology, along with their hierarchies.

```
1 hld1:W68135                                    #Level 1
      link type
2     blink:isLinkedTo instD:segment1.
3
4 hld1:W68135
5     blink:isConceptuallyIdenticalTo hld:p0.#Level 2
          link type
```

Listing 4.5: Link types

[42] https://wordnet.princeton.edu/

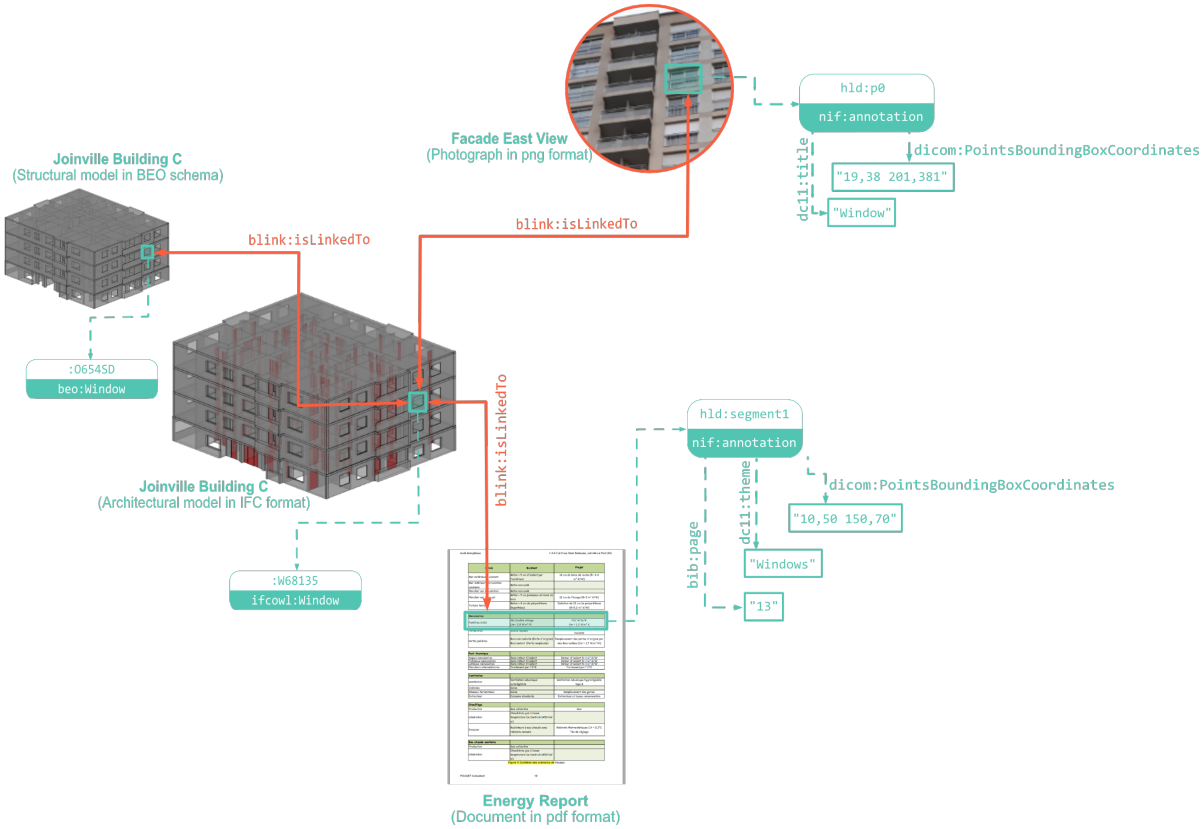

Figure 4.11: Discovering link relationships amongst heterogeneous AEC data, illustrated using BIM4Ren project

Fig. 4.12 shows a simple example of two types of link for Window ID W68135: the first is to segment1 in the PDF document, which is identified with a simple basic link sub-class `blink:isLinkedTo`. This same window also contains another link `blink:isConceptuallyIdenticalTo`, which is a semantic link, used to denote the region annotated in the image is conceptually related. Listing 4.5 shows the corresponding serialisation of this example in turtle.

Another example is shown in Fig. 4.13 and Listing 4.6, where the structural model's Wall is identified to contain the reinforcement bars which are `containedIn` (and hence uses the relationship `blink:isContainedIn`) in the architectural model.

```
hld:ReinforcementDetail_657                    #Level 2
    link type
    blink:isContainedIn instA:Wall_51.
```

Listing 4.6: Link types

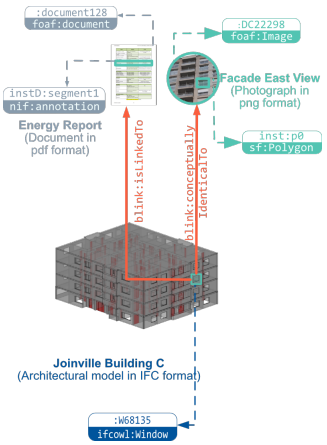

Figure 4.12: Linking of a segment in the energy report with a section of an image

In all of the above examples, a level 1 approach was adopted, where no additional metadata were added such as the author of the link, its creation date/time etc. To add this additional

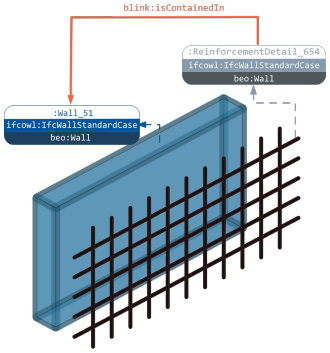

Figure 4.13: Link between the architectural representation of a wall and the reinforcement bars from the structural model

[43] https://github.com/SemanticHub/BuildLinks

information, level 2 of the BuildLinks ontology has to be used, which is discussed in the next section. The structure of the links and their management in the information container are tackled in this section, and also in Chapter 5. The above ontology is published on GitHub[43].

4.3.2 Link provenance

Provenance is defined as the record of an entity's (i.e., a resource) origin. In this chapter's context, this includes the entity's author information e.g. name, organisation, etc., along with who created what kind of links.

To illustrate the above concept, let us take the example introduced in Fig. 4.11, each of the predicate `blink:isLinkedTo` can be assigned its own provenance metadata. But before this can be done, this predicate has to be represented so that statements (or metadata) about it can be added. Listing 4.7 shows the RDF representation capturing the detected link.

```
hld1:wall_51 blink:isConceptuallySimilarTo hld2:
    wall_65.
```

Listing 4.7: Link example

If metadata like the history of this link, author information, etc., are assigned to this triple, then this is captured in additional triples. Essentially, this means another triple in which the subject would be the triple listed in the Listing 4.7. In other words, provenance information in this instance is captured by making statements about other statements. Theoretically, this is defined as **reification** - a statement which expresses an abstract construct using existing statements.

While there are multiple approaches for representing reification like Singleton properties, standard RDF reification, and N-ary relationships, each has its own set of disadvantages. The latter two are considered too verbose, which consequently affects maintainability, while Singleton properties are is not efficient for querying (Orlandi et al., 2021). Instead, provenance data is represented using quads, specifically RDF-star.

```
<<hld1:wall_51 blink:isConceptuallySimilarTo hld2:
    wall_65>>
    schema:dateCreated "2022-11-09T12:50:10Z"^^xsd:
        dateTime.
```

Listing 4.8: Provenance example using RDF*

The type of provenance data for the use cases considered in this research varies from the conventional approach. In this investigation, provenance is required for two cases: 1) provenance for resources (models and images in this case) and 2) provenance for the links detected and created according to the approach in Section 4.1.

For both of the above cases, the metadata remain largely the same, with additional information needed for link creation. For example, information about the creator e.g. the creator's name, creation date, modification date, organisation of the creator etc. However, for link creation, in addition to the metadata listed above, additional information like the confidence score for the link, the algorithm used for detecting these links, and the algorithm author is also needed.

Table 4.5 lists ontologies which specifically focus on describing links and their metadata, with additional support from other generic ontologies. Additionally, dedicated vocabularies such as VoID: Vocabulary of Interlinked Datasets[44] de-

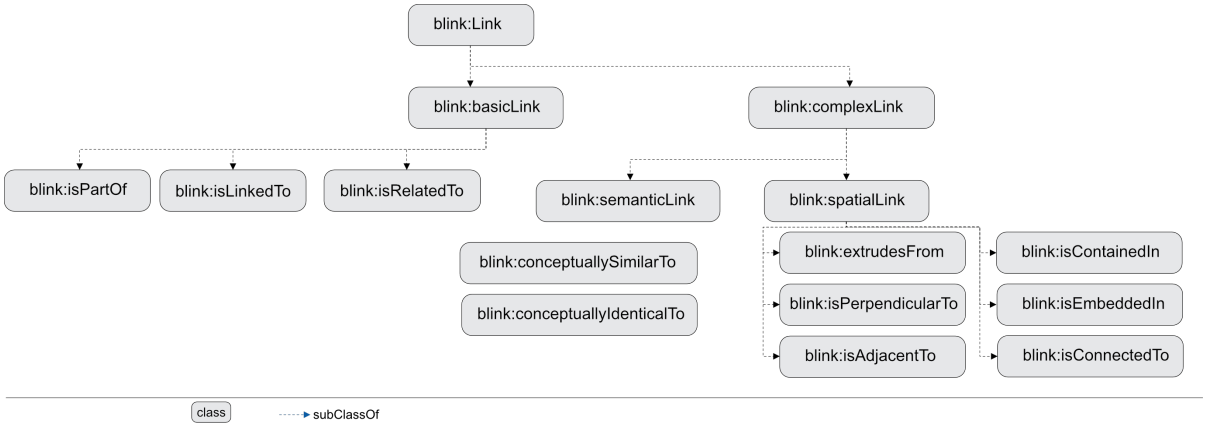

Figure 4.14: Overview of the Classes and their Hierarchy in the BuildLinks Ontology

fine vocabularies for metadata defining link targets, triples, datasets etc. The ICDD standard is the closest near-full-fledged counterpart for this in the AEC domain. A combination of these is used to describe the complete link-based metadata in Listing 4.9.

Ontology	Prefix	Namespace
prov	prov	http://xmlns.com/foaf/0.1/
DC Terms	dcterms	http://www.w3.org/2004/02/skos/core
OSLC	oslc	http://dbpedia.org/ontology/
schema	nsl	http://purl.org/ontology/
LinkLion	llont	http://www.linklion.org/ontology#
Onyx	onyx	http://www.gsi.dit.upm.es/ontologies/onyx/ns#
MARL	marl	http://www.gsi.dit.upm.es/ontologies/marl/ns#

Table 4.5: Ontologies for describing metadata about links

```
<<hld1:wall_51 blink:isConceptuallySimilarTo hld2:
   wall_65>> schema:dateCreated "2022-11-09T12
   :50:10Z"^^xsd:dateTime;
    blink:algorithm :Wordnet;
    blink:confidenceScore "0.80";
    prov:wasGeneratedBy :Madhu.

:Madhu
    a foaf:Person, prov:Agent;
    foaf:givenName "Madhumitha";
    foaf:lastName "Senthilvel";
    foaf:Organisation "RWTH Aachen University".

:Wordnet
    a llont:algorithm;
    schema:dateCreated "2019-12-05"^^xsd:dateTime;
    dcterms:creator "Some Author".
```

Listing 4.9: Full-fledged provenance metadata using RDF*

The confidence score, i.e. the level of trustability is a piece of important information in data provenance. In this research's context, this confidence level applies to the links created in Section 4.1. An illustrative provenance data is shown using the simplified Hello Wall example (Fig. 4.15). In this figure, metadata, which are represented in orange belong to the provenance layer.

[44] http://vocab.deri.ie/void

The above listings use RDF-star which can easily add properties that track both the data sources and the confidence levels[45]. With issues of regulatory compliance and data privacy continuing to grow, it's increasingly important to provide traceability with data provenance. It uses the concept of reification, where triples are used as the subject or object resource.

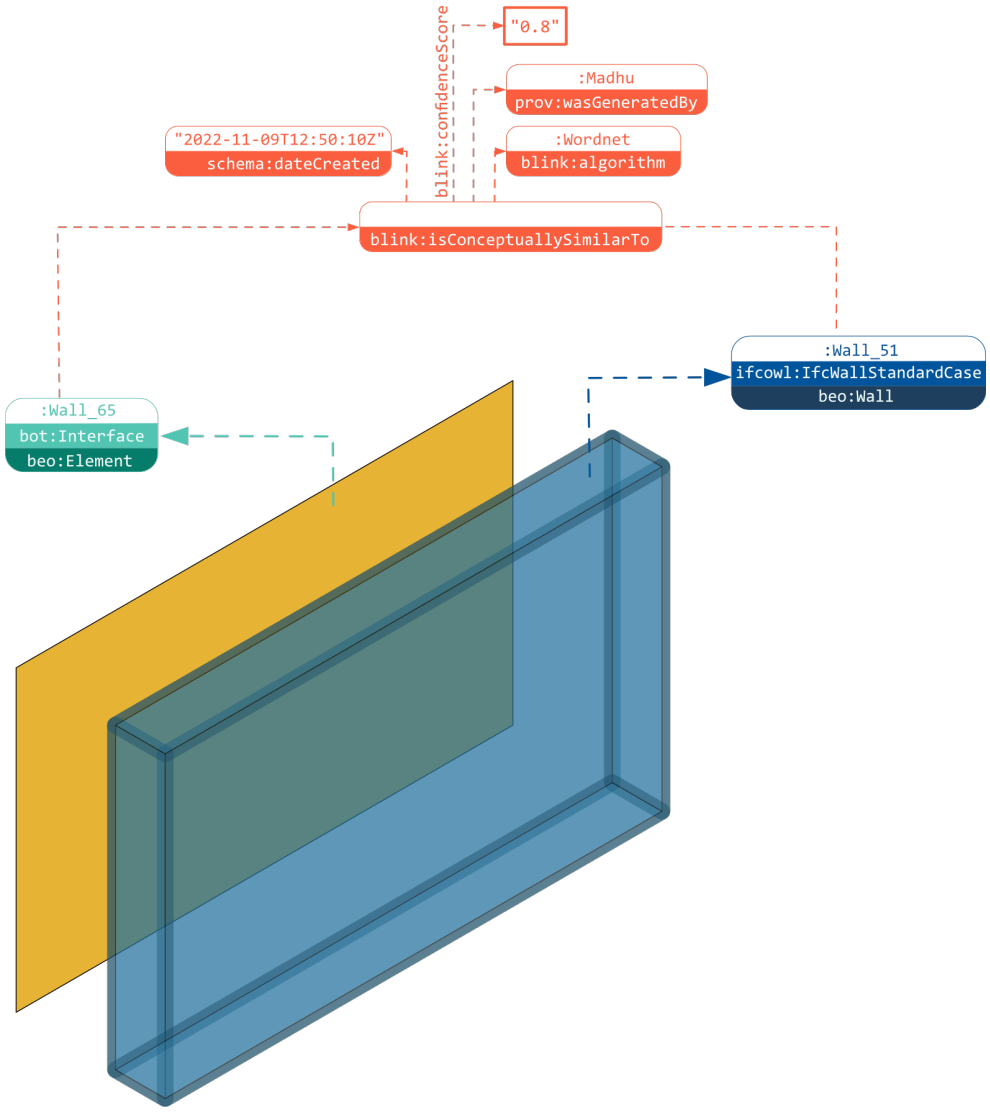

Figure 4.15: An annotated wall in two representations with provenance information

4.3.3 Link storage and retrieval

After a successful link discovery process, all the detected links and their corresponding metadata have to be stored as persistent data in a structured format. Within the scope of this research, the following fall under this task:

- Image metadata
- Document annotation metadata
- Links detected
- Link Provenance metadata

The link relationships identified in Section 4.2 are considered many-to-many, as multiple IFC elements can be associated

with multiple document sections and vice versa. The storage of link relationships can vary depending on the chosen approach. One possible method is to store metadata alongside the original data within the file itself. Alternatively, these metadata can also be stored separately. To ensure ease of data retrieval and maintain data integrity, all types of metadata are stored separately from the original data.

For the storage of the links and the link provenance metadata, approaches from existing standards can also be borrowed. For example, LDP illustrates links declared as *linkable elements* and then directly connects these elements to each other using a binding predicate.

Fig. 4.16 shows two layers: a ***data layer*** that contains heterogeneous data in their original format, and a ***Metadata sub-layer*** which contains the annotations of the documents and images. Additionally, it also contains the link metadata, i.e. all the links detected using these annotations, and their corresponding provenance information. Partial models like the architectural model and the structural model do not have a separate annotation file, as they are machine-readable, and directly used as inputs for link discovery.

The detailed triple structure for storing the links as per ISO 21597 ICDD part 1 and 2 recommendation is shown in Fig. 4.17. This visualisation utilises the example use case introduced in Fig. 4.10 where the detected links for a window in the architectural model, structural model and an image are shown. ICDD uses a three-tier structure to establish relationships between related documents. First, the documents are described on the basis of their title and classification (whether they are internal, i.e. originating within the system, or external, living outside the environment where the ICDD is being leveraged).

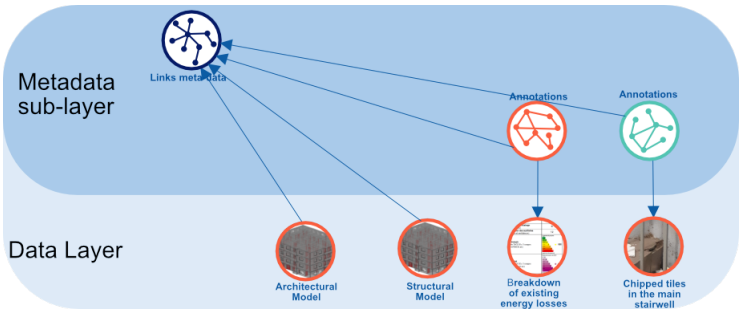

Figure 4.16: Link storage in layers

In the second part, the elements which are *linkable* are de-

clared as such in the `linkset` file (oval highlighted in pale green in Fig. below). Lastly, these *linkable elements* are assigned a link relationship (in this case `els:IsSameAs`). The separation of resources into linkable elements and the links helps to create deep links, i.e., links that supply information beyond the mere definition of a link relationship between elements. Within this thesis context, this denotes the provenance metadata proposed in the above sections.

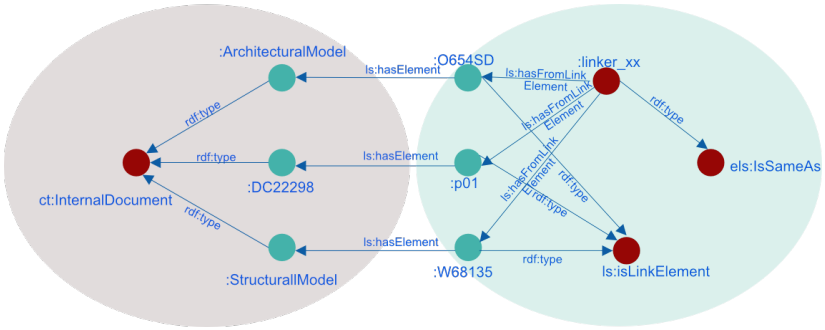

Figure 4.17: Link serialisation

The link metadata is stored in an RDF-compliant format (as a turtle file), and making it retrievable using any RDF-based query language such as SPARQL.

Listing 4.10 shows a SPARQL query to retrieve all information related to the concept `Wall_51` across four different partial models, based on the example shown in Fig. 4.9.

```
SELECT *
WHERE {
    ?s blink:isConceptuallySimilarTo ?o.
    blink:isConceptuallySimilarTo blink:
        confidenceScore ?o2.
}
```

Listing 4.10: SPARQL query for extracting connected information across

As stressed earlier, the original ICDD's structure for storing links requires a three-tier hierarchy of link description. This arduous structuring of link relationship can be simplified by taking advantage of Linked Data principles. These alternative approaches were explored in our paper (Hagedorn, Senthilvel, et al., 2023). Here, two types of modifications were proposed to the existing ICDD structure: 1) extending the current `linkset` ontology by modifying the range and domain of the predicates namely `ls:hasLinkElement`, `ls:hasFromLinkElement` to allow links to other resources

and, 2) assigning a link class as a property value directly, thus making it a two-tier link description.

As part of the above investigation, two prototypes of ICDD using the current specification were presented. Additionally, several recommendations for future development were also presented. These included prior alignment between project participants supplemental agreements for structuring link elements, identifiers and relationships falling outside the boundaries of ICDD; expanding the classes and properties in the current ICDD Schema to accommodate upper domain ontologies; and employing ICDD beyond the ZIP archive approach. This final recommendation is also explored in the following chapter, where Information Containers are functionally elevated to ensure their compatibility within a CDE.

4.4 Summary and conclusion

This chapter presented and elaborated on the adopted approach for the discovery of potential link relationships for three types of data sources: Images, Models and Documents. It also sketched out the minimum metadata required in these sources for enabling link discovery. These metadata were represented using identified existing vocabularies. Examples including graphic and code-listing were provided to demonstrate these ideas. Finally, the chapter culminated in the discussion of how the above information is represented and stored as persistent information, which can be retrieved through querying.

The next chapter deals with how the links discovered and their associated metadata in this chapter can be stored in Information Containers that operate throughout the life cycle of a project.

Chapter 5

Information Containers

5.1 CDE for linked heterogeneous information

Chapter 4 outlined the crucial of CDEs in facilitating conventional requirements for data storage and monitoring. Moreover, to reach Stage 3 of the BI maturity stages, these environments must support pragmatic project needs for creating, storing, and modifying relationships between diverse data sources (both within and outside the CDE).

"The current tendency in the field of CDE is similar to the situation of the interoperability of proprietary authoring file formats. So far, it seems that the emergence of CDE has followed the same dichotomy similar to OpenBIM versus ClosedBIM." — (Bucher & Hall, 2020)

Furthermore, Chapter 4 highlighted the intricate nature of heterogeneous data and its significance in AEC projects. It also proposed an approach for discovering and storing deep links, i.e. object-level links between these diverse data sources.

The links thus discovered can be used across multiple project phases, regardless of the nature of the project itself. Although there are differences, the phases in renovation projects overlap significantly with green-field projects, with one major difference: the availability of data on the existing as-is state of the building/infrastructure. These, as explained through the BIM4Ren use case can consist of, but are not limited to plans, drawings, photographic images, point clouds, as-is BIM generated, reports for energy analysis etc.

The significant phases for any kind of AEC project can be generalised to **Planning, Design, Construction, Operation and Maintenance**. In these phases, planning schedules, documentation reports, images of construction site and construction progress, 3D models for conceptual design, etc. are continuously generated. These heterogeneous data are generated by a multitude of participant teams who repeat-

edly interact, exchange these data during each phase. During the closure of a phase, they also hand over the 'finalised data' to other teams in charge of operating and maintaining the built infrastructure.

These constant interactions and data exchange usually take place in disparate, project-specific, and use case-specific ad-hoc methods, e.g. zip files shared through emails, and cloud storage, and generally organised as a folder structure. These bundles are called **containers**, and they are used to ship related information between teams.

Though conventional containers follow user-defined rules, and scope, there are some specifications from existing standards which elaborate the use of these containers and their requirements. For example, the ISO 21597 ICDD specification is intended for the handover of data for archiving. LDP, a non-AEC approach, does not specify any phase, or task-specific description. DIN SPEC 91391 focuses on data exchange between different participants during all phases of a project through specifications for functional requirements of CDEs. As part of this, information containers that hold connected data are also exchanged throughout the project phases.

However, as noted in Section 2.6.1, the DIN specification has limited addressal for the definition structuring and storage of 'linked' heterogeneous data. Lastly, Multi-Model Containers, as per their conceptualisation, are intended as a data exchange approach throughout a project lifecycle and rely on an XML structure to enable this. However, it too has limitations in the level of deep-linking possible using its specifications.

From above, it is evident that the existing Information Container specifications are required to not just function throughout the project lifecycle for the storage of linked information, but also for the exchange of information. This 'dynamicity' of accommodating ever-changing data, being passed back and forth, also has to function within CDEs, where their contents and relationships change frequently.

This chapter takes a look at how dynamic information containers can be imagined using guidelines derived from ISO 19650 and DIN SPEC 91319 (the two standards that contain adequate requirements of Information Containers for a CDE).

In the upcoming sections, the proposed architecture for dynamic containers to function in a CDE is described, along

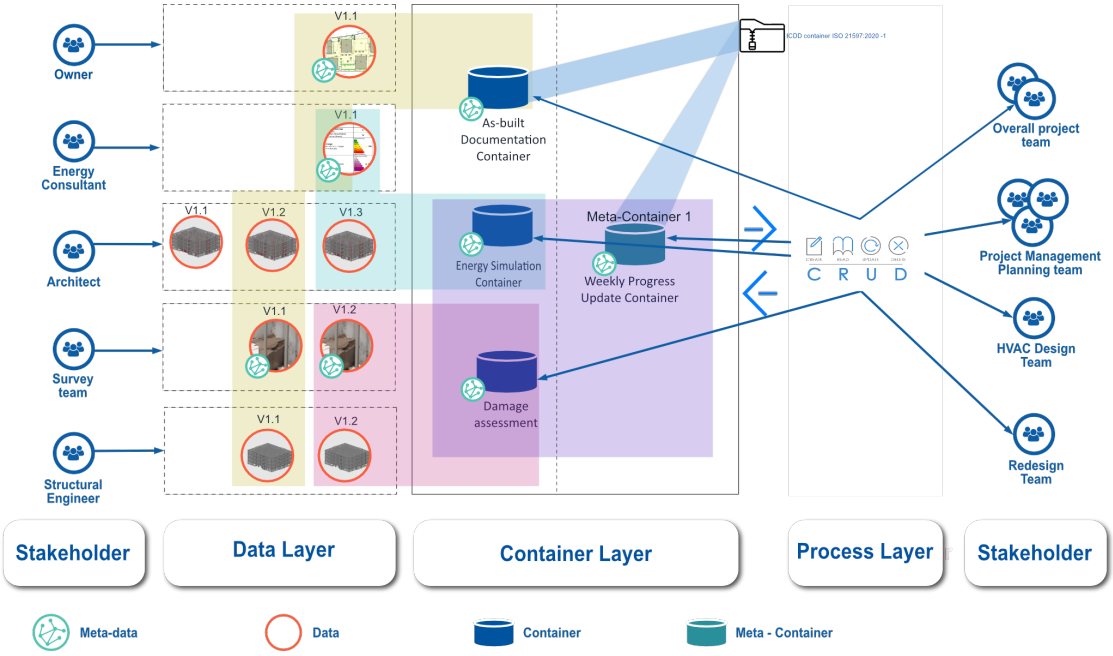

Figure 5.1: Proposed architecture for managing heterogeneous linked information in web-based information containers

with the overarching functionalities they are expected to fulfill. The underlying theoretical concepts such as ontologies and interfaces required to achieve this are also discussed.

5.2 Proposed architecture

To achieve the above, CDEs have to be re-imagined using a layered architecture. The conceptual foundation for the architecture and each layer is elaborated in the following sections.

This thesis proposes a three-layered architecture which facilitates dynamic information containers in a CDE used throughout the project lifecycle:

1. Data layer
2. Container layer
3. Process layer

Each of these layers is envisioned to build on top of each other to realise and deliver a CDE that accommodates the creation and management of linked heterogeneous information. In the following subsections, these layers are elaborated with supporting conceptual diagrams.

5.2.1 Data layer

In the data layer, all resources are stored together, regardless of their file format. Essentially, this layer acts like a data dump, wherein all the documents, BIM models, drawings, images, point clouds etc. can be stored. Each of these data has its own URI[46] which is dereferenceable. This layer also captures the inherent metadata of the resources uploaded to it, e.g., file name, file type, creation date, modification date, author information, and version information (if available). This layer is the fundamental cornerstone of the proposed architecture, as it stores any information in its original form supplied by a project participant.

[46] in the web-based CDE, this is an URL

Metadata	Brief description	Vocabulary used	Equivalents
file name	name of the uploaded file	ct:fileName	dc:title, rdfs:fileName
file type	type of file uploaded	ct:filetype	rdf:fileType,dcterms:format
creation date	date of file creation	ct:creationDate	dcterms:created
Authorship	author information	ct:createdBy	prv:createdBy, pav:createdBy
File classification	Internal or External (to the CDE)	ct:InternalDocument	dot:ExternalResource, dot:InternalResource, nao:externalIdentifier
Description	general file description	ct:description	dcterms:description, rdfs:comment
versionID	Version of file	ct:versionID	dbpedia-owl:version, dcterms:hasVersion

Table 5.1: Inherent metadata for resources in data layer

Fig. 5.2 shows one such example data: the overall site plan of the BIM4Ren in pdf format. Its inherent metadata like filename, filetype, version ID, author, etc. are captured using the ICDD vocabulary (introduced in Section 17 and used in Chapter 4). Additionally, the metadata created as annotations (using methods from Section 4.1) are also illustrated. In this example, although the inherent metadata and the annotations are represented separately, in the data layer, they can both reside in a single RDF graph.

Fig. 5.3 shows a similar example that contains two versions of the image of a damaged tile on a staircase. The image with version ID `"1.1"` contains only the inherent file-based metadata, while version `"1.2"` also contains additional an-

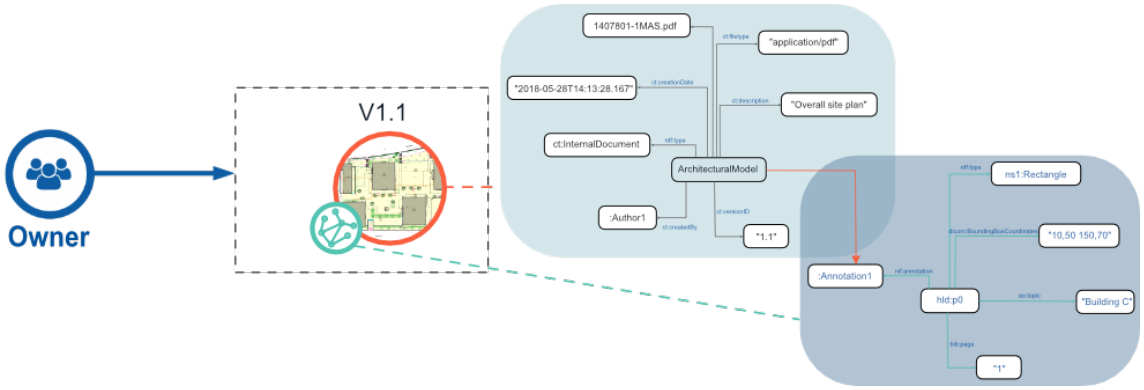

Figure 5.2: Annotation metadata graph for drawings/documents in the Data layer

notation metadata. Both the inherent metadata and the annotation metadata are stored together in an RDF graph.

This layer relies on a mix of many vocabularies like the ICDD schema, DC terms etc. The above metadata are represented in the vocabulary as shown in Table 5.1. Due to the availability of extensive ontologies containing similar, or overlapping terms, only a few equivalents are listed here.

5.2.2 Container layer

The container layer builds on the data layer introduced in the previous section, where the user can create virtual containers for project-specific use cases. Each container consists of some minimum mandatory metadata as recommended by DIN SPEC 91391 and part 1 of ICDD - ISO 21597. These include container name, description, authorship details, and provenance information e.g. version ID, creation date, etc.

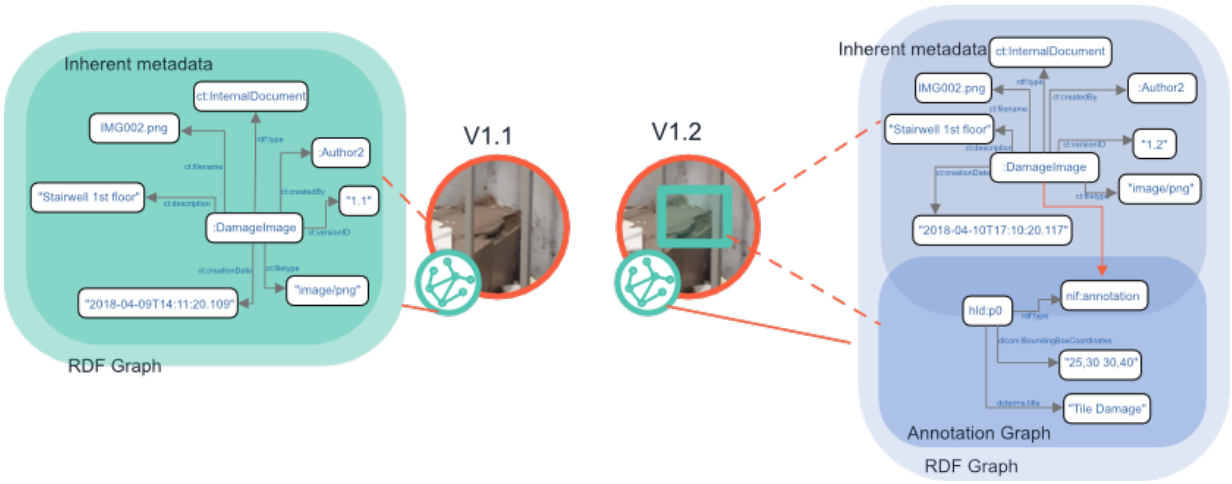

Figure 5.3: Overall metadata (including inherent and annotation metadata) for image versions

In addition to these, each container also requires dedicated metadata for capturing the assigned state as recommended by the ISO 19650 standard. These states are 'Published', 'Shared', 'Work In Progress (WIP)' or Archived'.

A container in this layer is a virtual construct which contains only an RDF graph consisting of the metadata of the container (see Table 5.2) along with the resources it contains from the data layer. These resources are referenced using URIs (or URLs) as defined in the RDF graph of the data (refer to Fig. 5.3). Note that both the container metadata, the resources metadata, and the relationship between these resources are stored in this RDF graph.

This approach differs from the approach adopted by ICDD where container metadata and resource metadata are stored in the index, while the link-able elements of these resources, and their relationships are stored separately in the *linkset graph*.

A typical container decomposes as shown in Listing 5.1. Here, two documents (a schedule spreadsheet and an IFC model) are contained within this container. For brevity, only the file name of these documents is shown in this listing, though typically other information such as creation time stamps, modification time stamps, author information, etc. are also present.

The container layer also facilitates the nesting of containers. These are called '*meta-containers*' and follow the same structure as the generic containers, with one additional component: their RDF graphs contain a container-to-container relationship definition. Fig. 5.5 illustrates two containers: one container for energy simulation which uses an energy assessment report and version ID `1.3` of the structural model and another container for damage assessment. This latter container consists of photographs of the damage and the IFC architectural model.

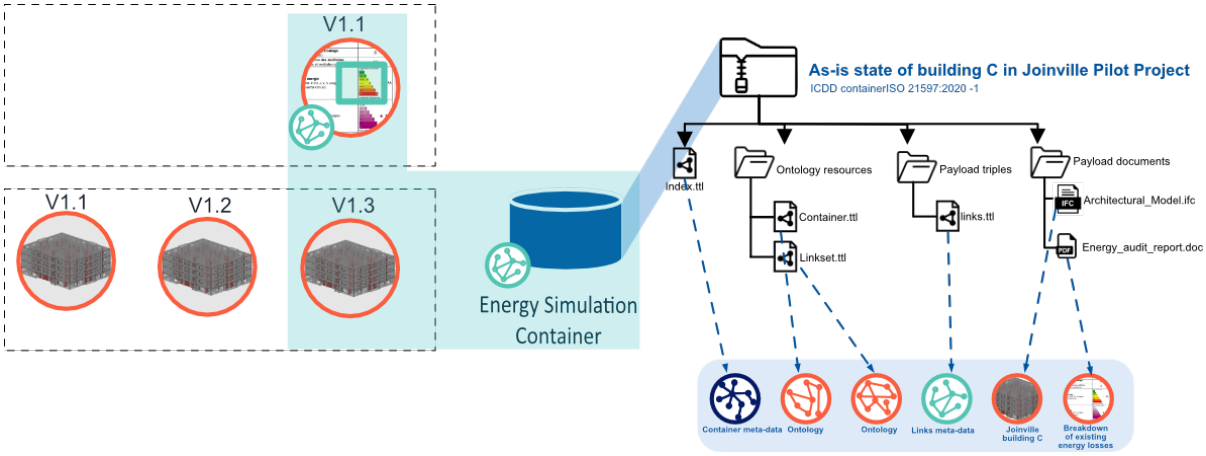

Figure 5.4: An example container for energy simulation from the container layer, which when exported follows the ICDD zip file structure

Both containers are nested within the container for weekly progress updates. The list 5.2 shows the RDF-based turtle serialisation for this type of container using a truncated version of the predicate `els:PartOf`.

The predicate relationship used above can vary depending on the use cases and the data contained within each container. While there are no specific vocabularies which cater to the definition of container-to-container relationships, conventional ontologies which are used to denote file-to-file relationships can be potentially reused here. Some of these vocabularies were identified in Table 4.5.

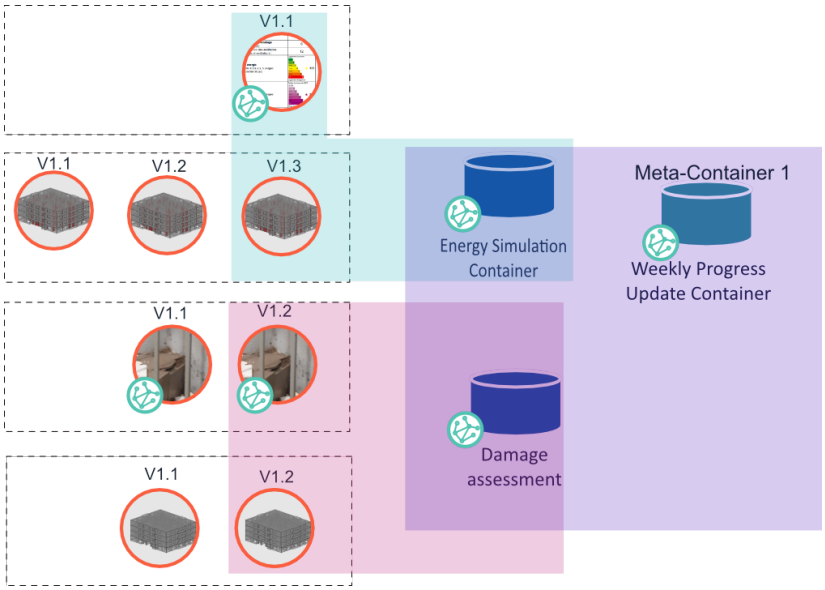

Figure 5.5: A meta-container consisting of two use case-based containers

```
hld:Container001
    a ct:Container;
    ct:description "Scheduling Container";
    ct:conformanceIndicator "ICDD-Part1-Container";
    ct:creationDate "2022-12-11T15:13:28.132"^^xsd:
       dateTime;
    ct:publishedBy hld:Person001;
    ct:versionDescription "first version";
    ct:versionID "1";
    ct:containsdocument hld:document1,
        hld:document2.
hld:document1
    a ct:InternalDocument;
    ct:filename "Architectural model 1".
hld:document2
    a ct:InternalDocument;
    ct:filename "Schedule spreadsheet 1".
```

Listing 5.1: Containers in the container layer

Since there are no existing vocabularies describing the status of a container, a lightweight ontology for describing the data states as recommended by ISO 19650 was also created and published[47]. This ontology complies with FAIR (Findable, Accessible, Interoperable, Reusable) principles generally recommended for ontology creation as espoused by (Wilkinson et al., 2016).

[47] https://github.com/SemanticHub/DataStateOntology

```
hld:Container001
    a ct:Container;
    ct:description "Scheduling Container".
hld:Container002
    a ct:Container;
    ct:description "As-built Dcocumentation
       Container".
hld:Container003
    a ct:Container;
    ct:description "Weekly Progress Update
       Container".
hld:Container001,
hld:Container002
    els:PartOf hld:Container003.
```

Listing 5.2: Nested containers in the container layer

Both of the container examples explained above used different versions of a particular data resource. In the ICDD schema, `ct:versionID` and `ct:versionDescription` can be used to define the versions of the resources, and are used in these container definitions too. For describing explicit relationships between different versions of containers, the predicate `ct:priorVersion` can also be leveraged.

Ontology	Metadata	Class / Property
ICDD	Container description	ct:description
ICDD	Authorship	ct:createdBy
ICDD	Container contents	ct:containsDocuments
ICDD	Provenance	ct:creationDate, ct:versionID, ct:versionDescription, ct:publishedBy
DSO	Container State	dso:State

Table 5.2: Inherent metadata for container resources in Container Layer

In principle, the containers in this layer were envisioned to also contain a metadata layer which stores the provenance information that is created when links are discovered (see Sections 4.1 and 4.3). Every container created here is exportable, and the exports conform to the ICDD folder structure, with the segregation of information between the index graph and the *linkset graph* as specified by ICDD ISO 21597.

However, within the CDE, all of these were modelled to remain in a single RDF graph, in order to improve the trackability. Other functionalities such as exports of containers are accessible through the *Process Layer*, which is elaborated in the upcoming subsection.

5.2.3 Process Layer

The Process Layer forms the final part of the proposed architecture. In this layer, the basic CRUD functionalities (Create, Read, Update and Delete) are available for the project participant. These functionalities enable them to access the heterogeneous files and their annotations in the data layer, and the containers and meta-containers in the container layer.

In addition, all types of project-related tasks and activities like time and cost monitoring, querying of information, etc. can be accommodated here. This layer's flexibility lets it be designed as the final front-end for the user.

In the current research, three functionalities were implemented in the front-end: 1) file uploads 2) container creations and 3) Reading containers (i.e. querying). The former two were already explained in Sections 5.2.1 and 5.2.2 and were implemented using React, Firebase, Express, and

Node.js. It was implemented using Javascript and SPARQL-based libraries, supported by GraphDB triplestores, where the data (which was being queried) was stored.

The processes and functionalities required for accessing and interacting with the above three layers are discussed in the next subsection.

5.3 Overarching functionalities

In the previous section, each layer of the proposed architecture was described in terms of the underlying theoretical concepts developed based on existing literature. These concepts serve to facilitate any service running in the process layer. These services can range from authorisation, CRUD access rights, and functional management roles, to data-centric services including querying, knowledge discovery, etc.

To enable the above-mentioned services, first, the mappings between different ontologies for the metadata descriptions have to be created. These provide the basis for assessing which functions arise from a standard (or approach) and the extent of its vocabulary (if any is available) for reuse.

Additionally, the sequences of operation for each service (e.g. querying) have to be defined. These sequences depict the interactions between a user's actions and the CDE and also determine the resulting behaviour of the CDE for each action.

This section focuses on two major parts of overarching functionalities for Information Containers to function in a CDE: 1) identifying metadata commonalities across different existing approaches, and if they have corresponding dedicated ontologies for representing them, and 2) identifying API commonalities across different approaches and their relevance for the proposed architecture.

5.3.1 Mapping concepts across existing standards

The architecture proposed in the section above was conceived based on the gaps identified in existing literature in Chapter 2. The proposed conceptualisation above builds on these gaps by using them as requirements for the development of both information containers and their functioning in a CDE.

However, it is beneficial to also take a look at how much of

these requirements are met by existing standards and approaches mentioned in Section 2.6.1. Additionally, most of these requirements are reflected in dedicated ontologies or terminologies for representing them.

Abstraction	Concept	ISO 19650	DIN SPEC 91391	ISO 21597	LDP	OpenCDE-API
Container descriptions	Container description	◑	◑	●	●	○
	Membership	○	○	●	●	○
	container types	○	◑	○	●	○
	membership types	○	○	○	●	○
	Container state	◑	○	○	○	○
	Unique identifier	○	◑	●	○	○
Resource/File	File descriptions	○	◑	●	○	●
	name	○	◑	●	○	●
	relationship to container	○	○	●	◑	○
	file type	○	○	●	◑	●
	Unique identifier	○	○	●	◑	○
Versioning	Revision	◑	◑	●	◑	○
	version ID	○	○	●	○	●
	version description	○	○	●	○	●
Authorship	creator ID	○	○	●	○	○
	creator name	○	○	●	○	○
	creator organisation	○	○	●	○	○
Time stamps	Container creation date, time	○	◑	●	○	●
	File creation date, time	○	○	●	○	○
	Container modification date, time	○	○	●	○	○
	File modification date, time	○	○	●	○	○
Link types	types of links	○	◑	●	◑	○
Link authorship	creator ID	○	○	○	○	○
	creator name	○	○	○	○	○
	creator organisation	○	○	○	○	○
Link provenance	Link creation date, time	○	○	○	○	○
	Link modification date, time	○	○	○	○	○
Link meta-data	Link algorithm used	○	○	○	○	○
	Link confidence score	○	○	○	○	○

● Specification + Vocabulary ◑ Only Specification or Partial vocabulary ○ No Specification or Vocabulary

Figure 5.6: Mapping of concepts and corresponding vocabularies across existing resources

Fig. 5.6 lists the concepts extracted from all the standards mentioned in this thesis and their depth of coverage. Abstractions which are mentioned as specifications in standards and also have a dedicated vocabulary are represented in black, while representations in the color grey denote concepts which are covered in the standards without any referenced vocabulary for expressing them, and the ones in color white

contain neither specification nor any corresponding vocabularies for it. From this figure, it is evident that some concepts, though crucial for CDEs are not adequately covered and hence need to be borrowed from other approaches.

Fig. 5.6 presents a set of concepts categorised by overarching meta-concepts for metadata for both the files in the Data Layer and the containers in the Container Layer. It is evident from this figure that ISO 21597 contains the most comprehensive set of specifications for file and container-specific metadata such as versioning, resource descriptions, authorship, provenance, link types etc.

The DIN SPEC 91319 and LDP both contain partial vocabularies for some specifications for Information Containers and resources within them, while OpenCDE-API's Documents API focuses on minimal metadata for versions, time stamps, file author etc. ISO 19650 does not contain any high-level vocabulary for any of the concepts it recommends.

Numerous ontologies which comply with the FAIR principles espoused in the Semantic Web domain are currently available for describing any concept/element and its properties. So mappings between ontologies must be created and maintained. Without these mappings, the processing of any RDF data would be time-consuming, as well as the programming for these processes since no common understanding of these terms is developed. Mappings in this chapter's context refer to both the concepts and terminologies in ontologies in relevant standards.

The mapping presented here is translated to machine-readable RDF format, wherein a simple `owl:equivalentClass` is used for specifying the commonality of concepts across vocabularies. Of the 5 approaches assessed here, only the ICDD, LDP and OpenCDE-API specify their own ontologies, while the DIN SPEC refers to external ontologies for many concepts.

Consequently, in translated mappings, the most commonly used ontologies for these concepts are serialised and available on a dedicated GitHub repository[48]. It is essential to note here that the above figure does not capture the complexity of conceptual mappings. A more detailed analysis containing a one-to-one relational mapping of abstractions across these standards is provided in (Senthilvel et al., 2020).

[48] https://github.com/SemanticHub/CDEOntologyMappings

Similar to the mappings between ontologies discussed above, certain functionalities of CDEs are also mentioned along with

Abstractions	Concept	ISO 19650	DIN SPEC 91391	ISO 21597	LDP	OpenCDE-API
Container descriptions	CRUD descriptions	○	◑	○	●	●
	HTTP-based content management	○	◑	○	●	●
	Client Behaviour	○	○	○	○	○
	Server Behaviour	○	○	○	○	○
	Handling unauthenticated requests	○	○	○	●	●
Testing protocols	CDE conformance tests	○	○	○	●	○
Vocabulary use	Reusing published existing vocabulary	○	○	○	●	○
API	API architecture	○	●	○	●	●
User management	Authentication protocol	◑	●	○	◑	●
	Roles and rights	○	◑	○	○	○
Metadata	Process metadata	○	◑	○	○	○
	Workflow metadata	○	◑	○	○	○
Data Sovereignty and Security	User tracking	○	◑	○	◑	○
	Access controls	○	◑	○	○	○
	Data reliability	◑	◑	○	○	○
Processes	Dashboard	○	◑	○	○	○
	Reporting	○	◑	○	○	○
	Workflows for data exchanges	◑	◑	○	○	○
	Generic data management templates	○	◑	○	○	○
General	Information Structuring	○	◑	○	○	◑

● In depth, high-level specification ◑ Superficial, low-level specifiation ○ No specification

Figure 5.7: Identifying CDE concepts in various standards and approaches

varying levels of specification. To ascertain their relevance and scope for re-utilisation, these functionalities will have to be mapped. Fig. 5.7 illustrates which concepts and functional requirements for CDEs were specified in different standards and approaches along with an indication level - which is used for representing the coverage range of the specifications. It should be noted that most of LDP's specifications for CDE behaviour are non-normative, i.e. it is meant as a guideline.

From the figure above, it can be seen that the DIN SPEC 91391 provides a low-level specification guideline for a functioning web-based CDE. On the other hand, both LDP and OpenCDE-API provide high-level, in-depth specifications for the most expected functions and behaviour of a CDE. In par-

ticular, OpenCDE-API illustrates how a CDE interacts with its sub-parts to achieve a task through the use of web sequence diagrams (buildingSMART, 2021-03-31T07:00:20Z/2022).

These sequence diagrams, which define how systems interact and respond to each other serve as system requirements for further development. In upcoming paragraphs (Section 5.3.2), these web sequence diagrams are outlined and utilised for defining how the proposed CDE interacts within its layers and with external microservices.

Furthermore, the identified requirements are used as a base for orchestrating how various components of the proposed architecture interact with each other to fulfill a CDE's obligation. The next section elaborates on these interactions.

5.3.2 API Orchestration

All types of web-based services require configuration definitions on how they communicate with another piece of software. API is an intermediary interface that defines and enables different parts of software to communicate with each other through a series of exchanges consisting of requests and responses. In short, API orchestration consists of a series of chained call responses which are used by the system to solve a problem[49]. API specifications (like the ones which will be defined in this section) determine how to use a section of an API or implement it.

[49] for example, when a user clicks a button on the system to query, the APIs determine how the query has to be processed, the status codes if the query is successful and unsuccessful

Web APIs are used for exposing both a software's data and functionality through any front-end devices (e.g. laptop, desktop or mobile) to a web server through HTTP. Since APIs essentially define the expected software behaviour for every action, their design is crucial. Though there exist many protocols for facilitating standardisation of API behaviour, the common ones are Simple Object Access Protocol (SOAP), JSON Remote Procedure Call (JSON-RPC) and Representational State Transfer (REST)[50]. Most of the modern APIs are built on REST, which is a set of web architecture principles.

[50] an API protocol based on HTTP to exchange information between different services

A CDE is also a piece of software which can interact with end users and other software. Its expected behaviour can be configured by defining appropriate APIs for the functionalities. Sequence diagrams are usually used to visualise and communicate the requirements for a process or a system implementation. These requirements are transformed to a

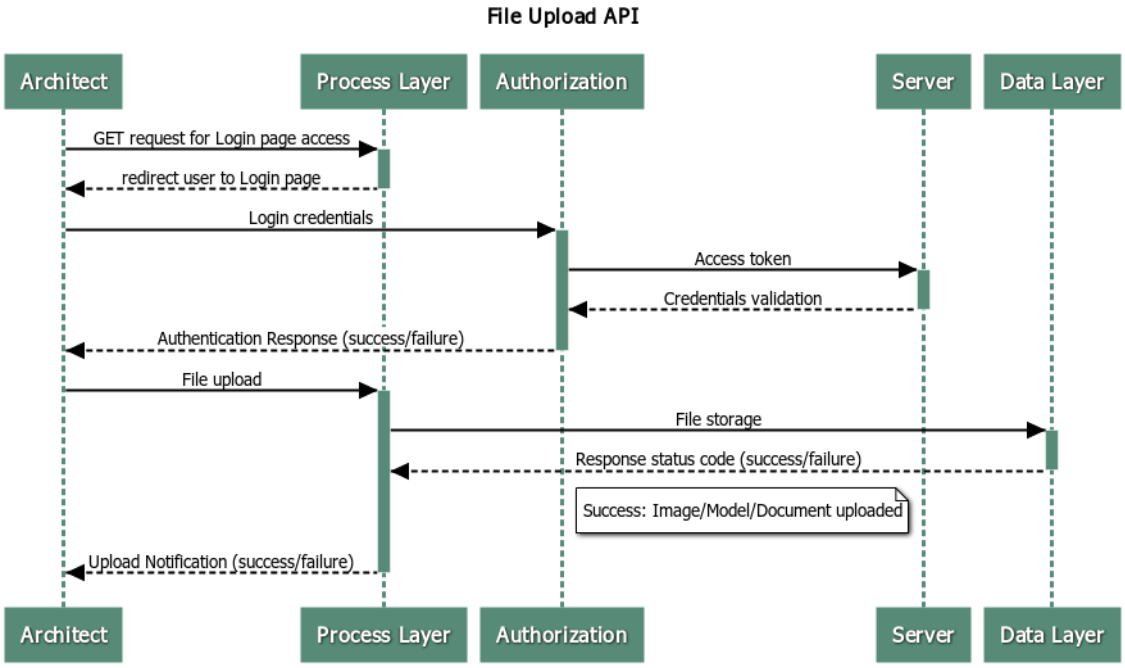

Figure 5.8: API call for a file upload to the Data Layer by an end user

higher formal level of refinement which is used by software developers. Hence, they are applied in API orchestration in combination with Unified Modelling Language (UML)[51].

[51] a general purpose modelling language intended for visualizing a system's design

Guidelines for the development of these APIs can be taken from two approaches: LDP- which was introduced in Section 12 and OpenCDE API - a buildingSMART initiative provides guidelines through a set of API standards primarily for issue reporting, document management and for the buildingSMART Data Dictionary. Both the LDP and Open CDE API provide guidelines on the expected software behaviour. As mentioned in Section 12, this specification defines a Linked Data architecture for a read-write functionality through the use of HTTP. This is specifically meant for CRUD operations for RDF resources.

In general, all CDEs contain a client and a server, with the former acting as the front end through which services (or data) is requested and the latter acting as the back end which supplies the requested service (or data). There are four types of these HTTP requests which are relevant for this research [52]: GET (used for information retrieval from a server through an URI), POST (used to send data to a server), PUT (used to updates resources based on user input), DELETE (used to remove requested resources from the server).

[52] according to HTTP 1.1

Fig. 5.7 stressed the relevance of both LDP and OpenCDE-API for providing guidelines to define minimum CDE func-

tionalities and implementation. For the scope of this research, the CRUD operations are the core CDE functionalities focused in this section.

API protocols are conventionally used to define how these CRUD operations can be delivered. Though LDP provides the basis for some of these API responses, buildingSMART's OpenAPI provides the most comprehensive specifications and is hence used for this research. The behaviour of ICDD-based containers according to these API specifications was investigated previously (Senthilvel et al., 2020) and is referenced and extended here.

Figure 5.9: Notations relevant for reading the sequence diagrams

In this research, API calls are modelled using UML, and one such call for file upload is shown in Fig. 5.8. This figure describes the web sequence of each action performed by the user on the front end and the corresponding backend responses. Sequence diagrams are primarily used to explain the interactions between parts of a system in sequential order and can also be used for representing requirements of system behaviour. Thus they form the base for developers who use these diagrams for fleshing the final system design.

Fig. 5.9 shows the components used in representing the sequence diagrams in this section. *Lifelines* are notation elements used to represent either roles or object instances or systems which take part in a particular sequence. Requests and responses between systems are represented as messages using solid and dotted arrows respectively. Using this as the base, fig/ 5.8 shows the end-user role *Architect* requesting to first login to the CDE using GET requests, which get processed through two systems: the Process Layer and the Authorisation system.

Upon successful authorisation, the *Architect* can upload a file (which can be an image, model or document) to the CDE, which is transmitted to the Data Layer and finally stored in the backend Server. The Data Layer also communicates this response to the *Architect* which marks the end of this flow.

Based on the API outlined above for uploading files to the CDE, Fig. 5.10 shows an API call for creating a container for a specific use case in the *Container Layer*. After successful login and authentication of the user, the 'Architect' can use the POST request for creating a container, which the *Process Layer* transmits as a request to the Container Layer.

Upon receipt of this request, the container layer initialised an RDF graph with basic inherent metadata like container

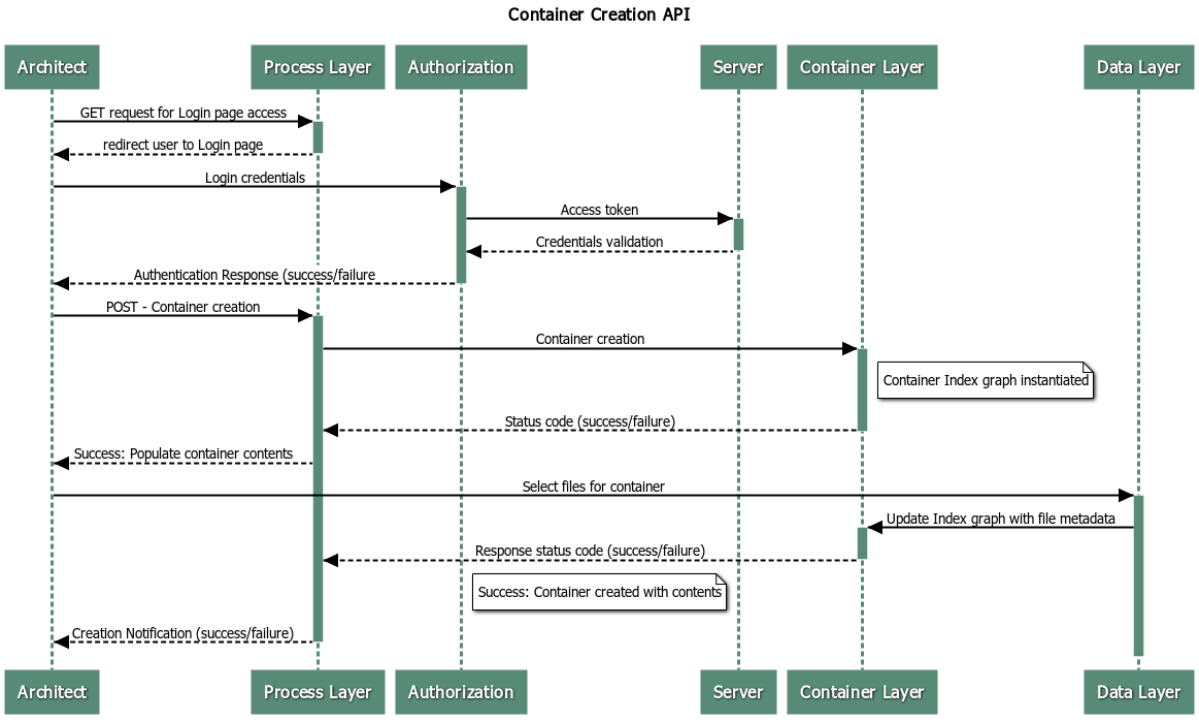

Figure 5.10: API call for creating and populating a Container in the Container Layer by an end user

name, date of creation, authorship details, etc. (as described in Section 5.2.2). This graph serves as the *Index graph* (as per ICDD structuring) and refers to the virtual container requested by the user. Upon successful creation of this container (or index graph) and notification of this to the `Architect`, he/she has to populate the container with relevant files (either internal or external).

If internal files are selected, then the Data Layer is accessed to display the relevant files, and once the 'Architect' selects them, the Data Layer sends a set of metadata associated with the selected files (refer to Section 5.2.1 for detailed metadata) and triggers the update of the *Index RDF graph*. Upon this update, the Container Layer transmits the notification to the 'Architect' through the Process Layer.

Containers, their contents and metadata annotations for files (e.g. images and documents) are routinely updated during a project's execution phase. Fig. 5.11 shows the corresponding API call for updating an existing container. After successful user authentication, the GET call is used for requesting the list of containers from the *Container Layer*, which is shown to the Architect who can select a container for updating.

Modifications to the container by the Architect are executed using the PUT call from the Process Layer through the Con-

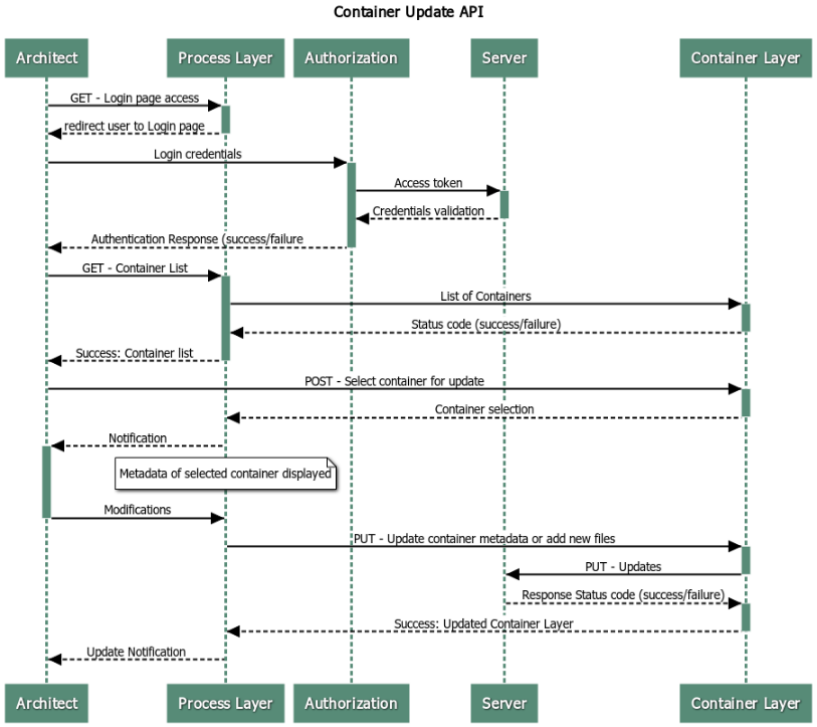

Figure 5.11: API call for updating a Container in the Container Layer by an end user

tainer Layer to the Server. Upon a successful update, a notification response is shown to the Architect.

The last of the CRUD operations is the *READ* functionality. All the information residing in both the Container Layer and the Data Layer can be read or queried by an end-user through the Process Layer. Conventionally, a GET request is used to retrieve information from the Container Layer, though a POST request can also be used if additional parameters or modifications based on queries are needed. Note here that the query is converted into a SPARQL query in the container Layer before transmitting to the Server.

Since container metadata are stored as an RDF graph, here, the dedicated RDF-based query language is used. The query response from the server is then sent through the Container Layer to the end user. Furthermore, if SPARQL queries are used for assertions leading to knowledge creation, the POST request can also be used.

Until now, the CRUD operations in the data layer and the container layer have not been performed. However, as mentioned in Section 3.2, the layers in this proposed architecture interact with external microservices for link discovery and conformance checks of the data. These microservices

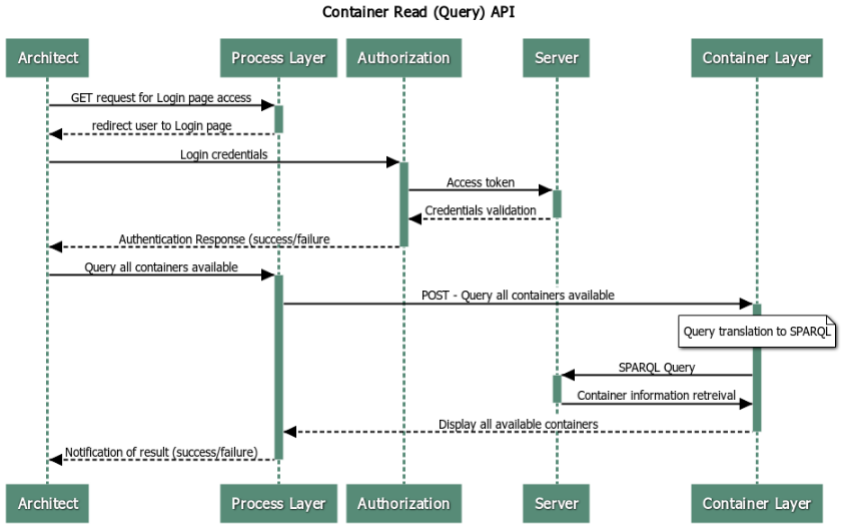

Figure 5.12: API call for querying all containers in the Container Layer by an end user

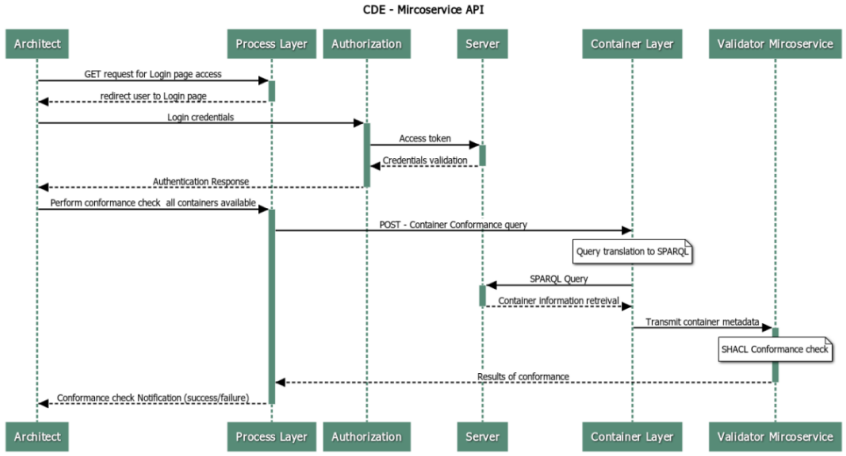

Figure 5.13: API call for the interaction of the Validator microservice's API with the CDE based on a user request for verifying that all containers conform to a set of rules

have their API which, once exposed and accessible, can be interfaced with any of the layers mentioned here.

One such example of the interaction of the validator API with the Container layer is shown in Fig. 5.13. The Validator API is a microservice that extracts container metadata from the container layer and uses it to validate if it conforms to the inbuilt rules (see Chapter 6).

In Fig. 5.13, once the end user sends the request for performing conformance checks for all of the containers in the container layer, a GET query translates this request into a SPARQL query, which is then sent to the Validator microservice together with the metadata of the containers retrieved from the Server. This microservice performs the SHACL conformance check and responds back to the Process Layer

of the CDE with the appropriate result notification for the check.

The microservice architecture proposed in Section 3.2 relies on multiple independent components, which are connected using REST API. Additionally the interaction of the validator service for checking data integrity and conformance (see Chapter 6) for both the data and container layer resources is shown in Fig. 5.13.

5.4 Summary and Conclusion

In this section, this research's core: the conceptualisation of information containers using Linked Data principles was elaborated. The proposed architecture builds on the concepts espoused in ISO 19650, DIN SPEC 91391, ISO 21597, LDP and OPEN CDE-API [53]. The containers thus designed support use case-based information exchange while underpinning federated information representation.

[53] material published and available as of May 2023

Based on the functionalities and the underlying theoretical framework, the proposed CDE architecture supports stages 2 and 3 of the BIM maturity level (14:00-17:00, 2018-12), since it supports both file-based (local machine) storage and web-based storage. Overall, it also supports federated model building, although functionalities such as fully automated link creation are not covered.

This chapter also identified the commonalities of metadata and its vocabularies across different standards and mapped them together to provide a theoretical base for defining information containers. It also identified the functionalities needed for a CDE from different standards and mapped it to the standards/approaches which mention them and contain the specifications or guidelines for it. From the identified functionalities, the CRUD functionality was elaborated by defining specific API calls using web sequence diagrams. These calls defined how a user's action interacted with the different parts of the CDE.

As mentioned previously, the architecture proposed in this chapter considers data blobs as the smallest referable unit, which is a fundamental departure from the conventional school of thought as per (14:00-17:00, 2018-12; *DIN SPEC 91391-1*, 2019-02-19) where containers are considered as the smallest referable unit. In conventional approach(es), due to the container being the smallest referable unit, data duplication tends to occur as the same file can be part of multiple con-

tainers depending on use cases. This conflicts with the data redundancy requirement, a key concept identified across all standards considered in this research.

Furthermore, the proposed architecture also supports only one of the two types of versioning for information within the CDE: namely, the change history of the metadata of all resources (files uploaded by the user, and the index graphs created in the CDE) are captured and maintained. However, iterations to the files themselves (e.g. information deletion, addition, and value changes) are neither detected nor recorded in the CDE. These are passed on to external tools (for example, model authoring software) which can capture the version history in its schema.

Chapter 6

Data Integrity and Conformance

The previous chapters introduced and described an approach for linking and managing information within virtual containers by reusing and building on top of existing standards and approaches. With the adoption of Linked Data approaches for improving and advancing BIM-based data management, each piece of information in the preceding Chapters 4 and 5 have to be efficiently used within CDEs and exchanged seamlessly across various applications (and also other CDEs) residing in the process layer.

To achieve this, the information created and stored in these layers has to be checked for conformance regarding data quality and compliance with specific use cases etc. This chapter addresses how checks can be performed so that they conform to pre-determined rules (both user-defined and as per specifications).

Conformance checks (also termed data validation) within this research's scope cover two broad use cases: a) Incoming data: Checking that all models, documents and images contain the necessary metadata, which is required for the link discovery (see Section 4.1) and b) Generated data: Checking that created links (both by the architecture proposed in this research and externally created ones) contain necessary metadata for establishing conformance of the created links (see Section 4.2 and Section 4.3).

The following subsections will elaborate on how these two types of checks were tackled, with accompanying implementation details of a prototype: a Validator bot, which was developed for demonstrating these checks.

6.1 Why check?

Conformance checks are a crucial part of information management. Throughout a project's lifecycle, information is constantly being streamed through data exchanges, being modified, and reused to serve specific tasks for specific use cases. Before each such use, the data must comply with a set of requirements that ensure its quality. These requirements can be project-specific (e.g. checking that IFC models contain the correct Level of Information (LOI) as defined in the BIM Execution Plan) or general conformance (e.g. version history for documents). Considering the highly fragmented and heterogeneous nature of information in the AEC industry, it becomes imperative that data exchanges between various participants take place with as little data loss as possible. Fulfilling these requirements enhances the completeness of the data, and results in successful data integrations with other kinds of information or in data exchanges with other project participants, thus improving the interoperability within the project environment.

Conventional manual rule checking has been recognised and established as excessively time-consuming, error-prone and labour-intensive (Beach et al., 2013; Dimyadi & Amor, 2013). As part of ongoing efforts toimprove efficiency, the use of rule languages for checking digitally-represented information is a widely researched theme (Beach et al., 2013; Nawari, 2018; Solihin & Eastman, 2015; Zhang et al., 2018). Beyond proprietary rule checking tools, BIM-based approaches also use mvdXML, and IDS[54] for defining and codifying rules for conformance checks of IFC schema-based project information.

[54]Information Delivery Specification - A buildingSMART initiative, in which exchange requirements are defined at the lowermost Level of Information; i.e. at the object, properties, values level

In this research, Linked Data technologies are used to represent semantics (i.e. metadata for files and containers) of heterogeneous data sets (e.g. images and documents); thereby making them RDF resources. As a result, it is necessary to use RDF-supported rule languages to check their conformance against any kind of requirements. These rules also serve to guarantee the quality and interoperability of the data when exchanged between various tools from various project participants. As data quality greatly impacts the usefulness of information for any task, these checks are crucial in ensuring seamless exchange of information.

Chapters 4 and 5 generated two kinds of data: metadata of files and containers (inherent) and metadata created from the link discovery process. The link relationships were created by matching the overlapping concepts identified in partial

models, images and documents. These two data types were either captured by the system or generated by the system. For example, using the user's input for container names or the author name of a link created. Although this data is stored automatically in an RDF graph, it still has to be checked if it contains all necessary data as required by the CDE proposed in this thesis.

Since all of the above metadata are represented in RDF, Linked Data-supported languages would have to be used for defining these rules. Some examples of these languages are SPARQL, OWL, SHACL, Shape Expressions (ShEx), R2RML etc.

One part of the checks carried out for the 3D partial models is "plausibility checks": manual visual inspection for assessing rough inaccuracies and missing geometric model content. However, with the advent of rule languages, these checks can also be codified, so that they can be transferred from "subjective" to "objective" checks - giving a quantifiable result of whether the model passed validation or not.

Model checking for federated data in a CDE requires that we consider that the data is present under the CWA assumption, though this contradicts the principles of "federations", which is based on OWA. This implies that, during rule-checking, any information which is not present will be treated as a violation of the rule, and not be assumed that it can be added at a later time/is present in another graph which is not yet connected.

Knublauch presents an elaborate comparison of Open World vs Closed World Assumptions, in addition to the implications of using OWL and SHACL (Knublacuh, 2022). In this research, the primary focus is the data validation using SHACL constraints; the inferencing capabilities of OWL are not covered.

6.2 SHACL for conformance checks

Various data languages such as SPARQL, OWL, ShEx, Object Constraint Language (OCL), and SHACL have been used to formulate constraints necessary for conducting conformance checks. Some of these like OWL, OCL, SPARQL have been commonly employed for validation purposes for RDF-based data. ShEx and SHACL are dedicated languages specifically designed for this task. For example, SPARQL[55] is an RDF query language to retrieve and manipulate serialised data in

[55] A W3C standard since 2008

RDF formats. It has been used for regulatory compliance checks (Beach et al., 2013; Pauwels & Zhang, 2015; Zhang et al., 2018) and data integrity checks to ensure interoperability during data exchanges (Lee et al., 2015). For regulatory compliance checks, international/national/local rules are codified from natural language to requirement identification. Once these are identified, they are formulated into SPARQL queries.

Similarly, for interoperability-based integrity checks, requirements are expressed as SPARQL queries. Typically, `SPARQL WHERE` clause and `SPARQL CONSTRUCT` are used for querying the RDF resource. The result of these queries can then be used to assess the resource's conformance to the rules. While SPARQL is highly expressive and most RDF products have some form of support for SPARQL-based querying, it was designed for information retrieval.

Furthermore, despite its expressiveness, it can be difficult to thoroughly construct SPARQL queries that can handle all possible variations of data, and reject any wrong structures. As a result, queries for validation can become excessively verbose, and eventually challenging to compose and debug.

[56]SPARQL Protocol and RDF Query Language

SPARQL[56] is a graph-matching language for querying RDF graphs. It can query graphs through matching patterns defined by users, modify solutions, and also construct new triples based on a query result. SPARQL's high expressivity and schema flexibility for querying have been employed in many research works for conformance checks. Bouzidi et al. proposed regulation compliance checking by first representing rules in natural language, and then using SPARQL queries for conformance checks (Bouzidi et al., 2012).

Krijnen and van Berlo utilised SPARQL for quality and regulatory requirements by encoding a query-based formalisation of rules (Krijnen & van Berlo, 2016). Zhang et. al also developed SPARQL functions for requirement-checking scenarios taken from the Dutch Rgd BIM Norm and the Norwegian Statsbygg BIM manual(Zhang et al., 2018). However, SPARQL's schema flexibility and expressivity can also be counterproductive, i.e. it becomes too verbose when used for large and complex constraint definition (Gayo et al., 2017).

SPARQL Inference Notation (SPIN) is a set of vocabularies which are used to express rules as SPARQL queries. It uses SPARQL's `CONSTRUCT` or `UPDATE` constructs to implement these rules. It was designed to provide adequate expressivity for both semantic querying and computation of RDF graphs

by linking class definitions with SPARQL queries. ShEx, another language for describing and validating RDF Graphs was developed by the ShEx Community.

Of the above, SPIN and ShEx are semiofficial W3C submissions, which never became formal specifications. SPIN was initially proposed to improve SPARQL functionalities by mapping classes to SPARQL-based rules and constraints. However, both SPIN and ShEx were also reported to have readability and maintainability issues (Gayo et al., 2017; Knublacuh, 2022).

OWL, an axiomatic Semantic Web language to represent the information on the web, has frequently been used to detect violations in data requirements, leverages inferencing for data validation, and works under the OWA(El-Diraby, 2014; Karan et al., 2016). Although its original scope was logic-based modelling of classes, properties and the resulting relationship between these, along with inferencing, its cardinality constructs have been used under CWA for data validations. However, a detailed analysis showed that OWL does not contain adequate language expressivity for constraint definitions, and its OWA results in significant tractability problems for violations(Knublacuh, 2022).

However, the use of dedicated rule languages like SHACL can mitigate these concerns. SHACL shapes can be divided into reusable segments which can be used for checking various elements in an RDF graph. For example, specific building elements like walls, windows, doors require a unique identifier to be linked to their definition. This modularisation of constraints into reusable blocks makes them concise and versatile. In the next section, this aspect of SHACL is presented and exemplified.

SHACL, a W3C standard language was designed to express constraints for RDF Graphs. Rules defined using SHACL allow RDF Graphs to be validated against them. Additionally, SHACL can integrate SPARQL queries within its shapes, thus allowing more flexibility and freedom in constraint definitions. SHACL specifications contain two parts: SHACL Core, which describes the core RDF vocabulary of commonly used shapes and constraints, and the second one on the extension of SHACL constructs using SPARQL, called SHACL-SPARQL. I

Inspired by SPIN, SPARQL and Resource Shapes[57], it contains high-level vocabulary for constraints like Cardinality, Data Type, Value Range, etc. A typical SHACL shape con-

[57] developed by IBM: https://www.w3.org/Submission/shapes/

tains two parts:

1. data graph consisting of the instance data in RDF format
2. shapes graph consisting of the constraints defined in the SHACL vocabulary

When the Data graph is validated against the Shapes graph, a SHACL validation engine generates an RDF report wherein violations are reported using SHACL's dedicated vocabulary, along with information on the elements/objects/triple nodes which violated the SHACL shapes.

Both Data graphs and Shape graphs can be either a single RDF graph or multiple graphs (federated graphs). A Shapes graph often contains multiple SHACL shapes, each containing the resource being checked, the corresponding Node Shapes for it, and property shapes within which conditions for the targeted node or its properties are defined. Node Shapes are SHACL constructs comprising of the constraints acting on the resources of a particular type of tripe in the data graph while property shapes comprise of constraints which act on the predicates (relationships) in the data graph.

Commonly used SHACL constraint types are:

1. **Cardinality constraints**: Simple constraints used to check any property, class or restriction for the frequency of occurrence, i.e. the repetition of a property in a given instance graph. This type of constraint is also often reused for verifying if a resource or its property exists, by limiting the minimum and maximum values to 1;
2. **Value range constraints**: These restrict the value ranges that a property can take. For instance, the property thermal transmittance (or U-Value) should be within 0.24 Units according to DIN 4108-7 specifications. This restriction is represented in SHACL using the `sh:minInclusive` and `sh:maxInclusive`, (including boundary values). The exclusion of boundary values can be achieved using the `sh:minExclusive` and `sh:maxExclusive`;
3. **Datatype constraints**: This constraint is used to restrict the datatype a specified property can take. For instance, for the example used above, the value has to be a `xsd:Integer`;

4. **Specialised Target Declarations**: Although we can use `sh:targetClass` to specify which class instances we would like the constraints declared as property shapes to act on, at times, specific subjects or objects of certain predicates need to be targeted. For example, all (instance) objects of `beo:Window` can be checked for the violations mentioned above. Where `sh:targetClass` is used to specify that all instances of a class must be validated with a shape, while both `sh:targetObjectsOf` and `sh:targetSubjectsOf` focus on the objects/-subjects of some property.

The value addition that SHACL brings due to its expressivity, especially for defining constraints has increased its appeal for conformance checks. Soman, 2019 explored the use of SHACL rules for automated look-ahead planning. Similar rules were also used to validate the ifcOWL models by Stolk and McGlinn, 2020, while Hagedorn and König studied the feasibility of SHACL-based validation for a fictive tunnel construction project (Hagedorn & König, 2020). Oraskari et al., 2021 framed unit tests (a feature of software development) for using SHACL to test models within the BIM4Ren Project by providing a comparison with mvdXML. Hamdan and Scherer, 2021 utilised SHACL rules for checking and assertions of semantic damage assessment descriptions for bridge maintenance.

```
All containers should have at most one description
   of the container's function
```

Listing 6.1: Example requirement in natural language

Listing 6.1 shows an example of a requirement based on the examples used in Chapter 4, where every container in the CDE Container layer should have at most one metadata value for the predicate `description`. The corresponding formalisation of this rule as a SHACL Shape is shown in Listing 6.2

```
:ContainerDescriptionShape
    a sh:NodeShape;
    sh:targetClass ct:Container;
    sh:property [
        sh:path ct:description ;
        sh:minCount 1;
        sh:maxCount 1;
        sh:datatype xsd:string;
    ].
```

Listing 6.2: SHACL Shape of the requirement described in Listing 6.1

A typical SHACL shape, like the one defined above, contains two main types of shapes: Node shapes - which are used to declare a constraint directly on a particular node, and Property shapes - which declare a constraint on the values with a node by following a specific path. In Listing 6.2, an arbitrary shape `:ContainerDescriptionShape` represents a Node Shape, which targets all resources which are defined as *ICDD Containers* (here represented as `ct:Container`).

Additionally, it defines a Property shape using SHACL's `sh:property` which targets the Container descriptions. Finally, within this shape, constraints are declared which apply restrictions on the components defined in the property shape. The cardinality constraints are defined using `sh:minCount` and `sh:maxCount` which specify only one value for the predicate `ct:description`. Also, it stipulates that the values of these descriptions should be of type "string".

```
ex:containerid13215
    a ct:Container;
    ct:description "container for scheduling".
ex:containerid6432 a ct:Container.
```

Listing 6.3: Instance graph snippet showing two containers: one containing a description metadata and another without this metadata

Validation of a *data graph* against a given *shape graph* results in a conformance report, which contains the list of triples that violate the defined constraints. This report utilises SHACL's dedicated validation terminology to indicate which parts of the triple violate the a SHACL constraint. Given an instance data graph (see Listing 6.3), and the SHACL shape defined above, the resulting RDF conformance report upon validation is shown in Listing 6.4.

```
[
  a sh:ValidationResult ;
  sh:resultSeverity sh:Violation ;
  sh:sourceConstraintComponent sh:
     MinCountConstraintComponent ; #Constraint
     being violated
  sh:sourceShape _:n1355 ;
  sh:focusNode <http://example.org/ns#
     containerid6432> ;
  sh:resultPath ct:description ; #Triple predicate
     causing the error
  sh:resultMessage "Less than 1 values" ; #Error
     description
] .
```

Listing 6.4: Confromance report of validation performed using SHACL Shape defined in Listing 6.2

The report is represented as an RDF graph using `sh:ValidationResult`. According to SHACL vocabulary which identifies that the constraint `sh:minCount` (minimum count) for the predicate `ct:description` has not been met, using `sh:resultPath`. Additionally, the property `sh:Violation` is used to indicate the severity of this non-conformance. If no explicit severity is defined (i.e.`Info` or `Warning`), the default `Violation` is automatically assigned.

Additionally, this result includes a description of the error. In this particular instance, it is an automated message, although customised messages can also be configured using `sh:message`.

In addition to its predefined constraint constructs, SHACL also provides SPARQL-based constraint declarations. The construct `sh:sparql` in conjunction with `sh:SPARQLConstraint` can be used to write a SPARQL constraint component. It can also be used for inferencing through the use of `sh:entailment`. Due to the breadth of SHACL's coverage, only constructs used in this chapter are discussed in the upcoming section as they are introduced. Comprehensive introductions of all available constructions are described very well in Gayo et al., 2017.

Given the differences in rule languages and their applicability for conformance checks, comparisons between SHACL and other languages like SPARQL, OWL, and SPIN have been explored previously by multiple researchers (Oraskari et al., 2021). Due to its status as a W3C rule language intended for data validation and conformance checking, SHACL is adopted for defining rules for checking data in this thesis.

Lastly, a distinction is made between the suitablity of these languages under specific conditions and their relevance to the present checks. Not all of the above languages are meant for data validation and they also operate under different assumptions. Rule languages typically fall within two distinct approaches from the knowledge representation domain: OWA and CWA. The former assumes that any statement (i.e. a triple) which does not necessarily conform to a given set of rules does not result in a *violation*, but rather that these data are inconclusive, i.e. *unknown*. This status can potentially change if and when future data is added. On the other hand, CWA considers statements which do not conform to a rule as *violations*, and conclusively results in non-compliance.

Though within the Semantic Web domain, the OWA is in-

herently present, in the context of the AEC domain and this thesis, CWA is required. If statements do not conform to rules defined by a user, then the instance graph violates the requirements and is hence termed 'non-conformant'. As pointed out by (Pauwels et al., 2017), the usage rules under these two assumptions influence the inferencing, though this is not within the scope of this thesis.

In the next section, the two types of conformance checks required for the proposed CDE architecture are sketched out with the definition of minimum metadata which has to be checked for each type of heterogeneous data (within this thesis' scope), corresponding SHACL rule set and a proposed process workflow for implementing conformance checks.

6.3 Types of conformance checks

As mentioned in the introduction of this chapter, there are two distinct types of validation that must be performed in this research. In the following subsections, each type will be discussed in detail, including the minimum requirements for models, documents and images, the process for conformance checking, and snippets of the SHACL rules to facilitate these checks.

The following sub-sections introduce the proposed usage of SHACL for conformance checks within the scope of this thesis along with relevant main constraint concepts necessary for their usage.

6.3.1 Incoming data check

In the first part of the check, incoming data (in this research, these are models and images) have to be checked if they contain the necessary minimal metadata corresponding to their element properties and annotations. This determination of the minimum requirements in the incoming data's metadata helps to establish which kinds of image/models/documents/information will be allowed in the CDE and highlight any missing data. They also help in drafting the cardinalities of the SHACL shapes.

Images and Documents

Table 6.1 shows the minimal list of requirements derived from the metadata for images introduced in sub-Section 4.1.2 and Section 4.3. These metadata are essential for making deep-link discoveries possible. As shown in the table, the majority

of these metadata undergo validation for both cardinality and data type.

```
:ImageShape1
    a sh:NodeShape;
    sh:targetClass dcterms:Image;
    sh:property [
        sh:path ct:fileName ;
        sh:minCount 1;
        sh:maxCount 1;
    ].
```

Listing 6.5: SHACL Shape for checking the existence and max. 1 value for file name of a resource

A key distinction from model-based checks is that the data types in image annotation are often restricted to literals. This can be partly attributed to the current lack of ontologies containing both typified object and data properties.

```
:ImageShape2
    a sh:NodeShape;
    sh:targetClass dcterms:Image;
    sh:property [
        sh:path dcterms:title ;
        sh:minCount 1;
        sh:datatype xsd:string;
        sh:maxLength 40;
    ].
```

Listing 6.6: SHACL Shape for checking the existence of at least one value for the property annotation title and its data type conforms to "string"

Metadata	**RDF representation**	**SHACL constructs**
file name	ct:fileName	[1,1] cardinality
file type	rdf:type	[1,1] cardinality
creation date	dcterms:date	[1,1] cardinality, data type
author	dcterms:creator	[1,n] cardinality, data type
author organisation	schema:organisation	[1,n] cardinality, data type
description	dcterms:description	[1,1] cardinality, data type
image title	dcterms:title	[1,1] cardinality, data type
annotation regions	nif:annotation	[1,n] cardinality
coordinates	dicom:PointsBoundingBoxCoordinates	[1,n] cardinality, data type
annotation title	dcterms:title	[1,n] cardinality, data type

Table 6.1: Minimum metadata requirements for deep-linking for Image

SHACL shapes for each of the above-listed metadata can be framed in numerous ways, and extended to capture other

requirements (for instance, specific triple patterns). For example, the first requirement: *there should exist only one file name for each incoming image*, which can be translated to a SHACL shape shown in Listing 6.5.

The above shape is very similar to the example shape defined in Listing 6.2, due to the simplicity of the requirement itself. Though most of the metadata listed in above will have similar SHACL shapes, requirements with multiple constraints will have different shape structures. For instance, the requirement that ***all annotations need at least one title, and this value should conform to the data type "string"*** needs to encode separate dependent constructs.

Listing 6.6 shows the shape consisting of these two constraints contained within the `sh:property` construct. Additionally, it is possible to restrict the character limits of these free text string properties by using the `sh:MaxLength`.

Due to similarities in the annotation metadata for images and documents, the same SHACL shapes can be reused for document conformance checks too. The SHACL shape which checks both images and documents for the existence of a creation date associated with it is shown in Listing 6.7.

```
:ImageShape1
    a sh:NodeShape;
    sh:targetClass dcterms:Document
                   ct:InternalDocument
                   ct:ExternalDocument
    sh:property [
        sh:path [
            ct:creationDate
            dcterms:created
            schema:dateCreated
            ];
        sh:minCount 1;
        sh:maxCount 1;
    ].
```

Listing 6.7: SHACL Shape of the requirement: existence and maximum two values for the metadata "file name"

The appendix A contains the comprehensive definitions of the SHACL shapes for image-based requirements in Table 6.1.

Models

Building models, regardless of whether they are based on IFC or non-IFC schemas such as BOT or BEO, have a more complex description of elements and their associated properties. This typically follows a structured hierarchy of the ele-

Metadata	Terminology checked	SHACL constructs
Link type	blink:link	[1,n] cardinality, data type
Link provenance	schema:dateCreated, blink:confidenceScore, prov:generatedBy, blink:algorithm	[1,n] cardinality
Creation date	dcterms:date	[1,1] cardinality, data type (xsd:date)
Author	dcterms:creator	[1,n] cardinality,data type (literal)
Author organisation	schema:organisation	[1,n] cardinality, data type (literal)
Description	dcterms:description	[1,1] cardinality, data type (literal)
Link confidence	blink:confidenceScore	[1,1] cardinality, data type (xsd:integer or xsd:float)
Link algorithm description	blink: algorithm	[1,1] cardinality, data type
Link algorithm author	schema:createdBy	[1,n] cardinality

Table 6.2: Minimum metadata requirements for validating links created by proposed CDE architecture using the ontology *Buildlinks*

ment definitions. As a result, rule-checking for these models becomes complicated and requires numerous. Based on the models introduced in Section 4.1.4 and in Fig. 4.9, the minimum requirements for the models are summarised in Table 6.3.

```
:ModelShape1
    a sh:NodeShape;
    sh:targetClass ifcowl:IfcWallStandardCase bot:
       Wall;
    sh:property [
        sh:path [
                ifcowl:ThermalTransmittance
                bot:thermaltransmittance
                s4bldg:thermaltransmittance
            ] ;
        sh:minCount 1;
        sh:maxCount 1;
        sh:minValue 1.2;
        sh:maxValue 2.5;
        sh:message "Thermal transmittance is not
           within acceptable limits"
    ].
```

Listing 6.8: SHACL Shape of the requirement: existence of a property for wall thickness and its value within per

In the above table, the last metadata `object properties` represents all the object properties which can be potentially checked for their existence. Due to the similarity of metadata with the image requirements such as file name, type

etc., they are not explained again in this section. Instead, the object properties requirements like the thermal transmittance of walls in a model are used as examples.

Listing 6.8 defines the SHACL shapes for all resources which belong to the class IfcWallStandardCase, and contains a property *thermal transmittance* (or U-Value) whose value should be within the range "1.2 - 2.5". A point to keep in mind: the units W/m2K are not explicitly stated. Additionally, the shape also specifies a customised error message for violations. Currently SHACL allows all measurement values to be defined in one unit of measure, i.e. it is not possible to represent properties having multiple units. This means that all values are assumed to be declared in a generic unit measurement, and only their value ranges are checked as part of the constraints. But, the unit of measure is not checked. Alternatively, units can be checked if they are represented as strings, using the `sh:pattern`, which can check the presence of a specific unit.

However, a convenient solution would be to use ontologies like Quantities, Units, Dimensions, and Types (QUDT). The collection of QUDT ontologies quantify units of measurements using class properties in the OWL schema("QUDT; Quantities, Units, Dimensions and Types", 2011). It aims to improve the interoperability of quantity-related RDF representations through its semantic specifications of the measurement units.

SHACL shapes can be formulated in a variety of ways, due to the language's flexibility in the modularisation of constraints. Therefore, a limited set of these shapes were listed in this chapter, and additional shape examples are available in the Annex A.

Validation workflow During the upload/creation of any of the above data resources in the CDE, the shapes defined above and in Annexe A will trigger the data conformance checks, to establish the data quality. Regardless of the nature of the incoming information (i.e. whether they are images or models), they follow the workflow shown in Fig. 6.1 for the initial validation.

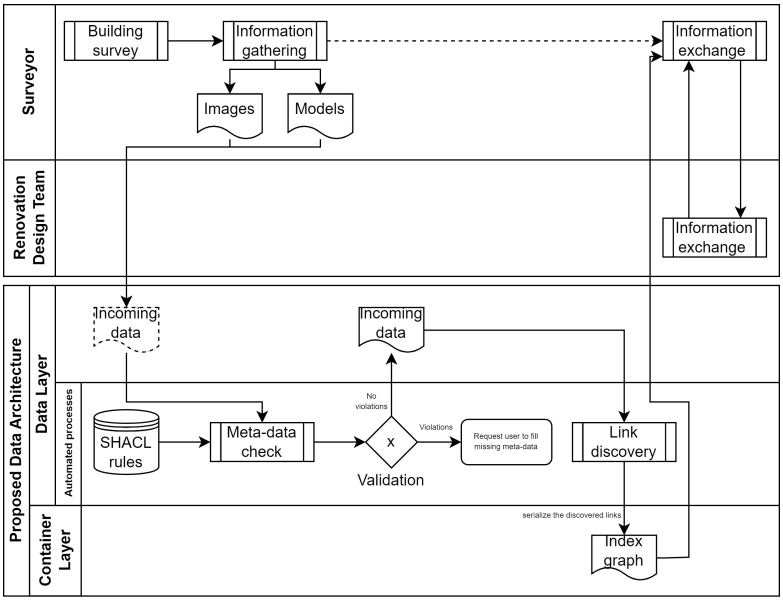

Figure 6.1: Incoming data validation workflow for the proposed architecture

Metadata	Brief description	SHACL constructs
file name	ct:fileName	[1,1] cardinality
file type	rdf:type	[1,1] cardinality
creation date	dcterms:date	[1,1] cardinality, data type
author	dcterms:creator	[1,n] cardinality, data type
author organisation	schema:organisation	[1,n] cardinality, data type
description	dcterms:description	[1,1] cardinality, data type
object properties (thermal transmittance, etc.)	ifcowl:ThermalTransmittance, s4bldg:thermaltransmittance	[1,1] cardinality, data type, pattern

Table 6.3: Minimum metadata requirements for model deep-linking

In this figure, the surveyor captures initial building data including models and images and exchanges it through the proposed architecture to the *Data Layer*. Before being persistent storage in this layer, it passes through the metadata requirement checks as defined in this section. Upon successful validation, it is sent for link discovery and eventually, all discovered links are stored in the container layer. Finally, all the data (including the uploaded data and the generated data) is accessible by the renovation design team through the container layer.

6.3.2 Generated data check

In the second part of the validation, the information generated as a part of the automated processes in the CDE is

checked for data integrity. This entails metadata from Section 4.3 - link types, link authorship, provenance etc. The requirements for this type of validation are listed in Table 6.2. The class definitions/object properties being checked are the ones being used for representing the metadata within this thesis. Hence, the SHACL shapes reflect these and can be adapted to reflect other definitions/vocabularies, should they be used.

```
hld:fileNameShape
    a sh:NodeShape;
    sh:targetClass [dc:Image, dc:Dataset];
    sh:property [
        sh:path [
            dcterms:fileName
            ct:fileName
            ct:fileType
            rdf:type
            ] ;
        sh:minCount 1;
        sh:maxCount 1;
    ].
```

Listing 6.9: SHACL shape for checking the existence of only one file name and file type for Images and Models

Note that unlike the SHACL constructs used in Section 6.3.1, the data types in the above table also have non-literals. This is owing to the machine-readable and inferencable approach used for serializing link information, where ontology-based class definitions and object properties were used for capturing this information.

Almost all of the SHACL shapes for the above constructs require a combination of multiple SHACL expressions. Hence, a few are elaborated in this subsection. In the Listing 6.9 the SHACL shape targets all resources which are declared as either Images or Dataset (here used to represent all types of models), and the property of file name as per the Dublin Core Ontology [58] or the ICDD Ontology[59], and finally the cardinality constraint of [1,n].

In the above example, the file type is also included since it acts on the same target classes and has the same cardinality as the constraint for the file name. However, there are many different ways of combining SHACL constraints within a shape. In the Listing 6.10, all resources which are declared as Dataset (here indicating models) should have only one file type, and this value should be equal to a literal, which is defined using `sh:value`.

More complex data validations require a combination of SHACL

[58] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/#http://purl.org/dc/terms/type

[59] https://standards.iso.org/iso/21597/-1/ed-1/en/

constraint constructs bundled together. The SHACL shapes are more complex for checking the conformance of the index graph as they contain constraints which follow a pattern and are often related to each other. For example, the metadata link type's definition usually varies depending on the structure of the ontology being used. Here, for ease of readability and reduced complexity, link-type terminology from the *Link Ontology* introduced in Chapter 4 is used. Additionally, the metadata *link confidence*is related to the link algorithm description and the algorithm authorship. The SHACL shape for these interconnected requirements is shown in Listing 6.12.

```
hld:fileTypeShape
    a sh:NodeShape;
    sh:targetClass [dc:Dataset];
    sh:property [
    sh:path [
        dcterms:fileType
        ct:format
        rdf:type
        ] ;
    sh:minCount 1;
    sh:maxCount 1;
    sh:value ["application/x-extension-ifc"]
].
```

Listing 6.10: SHACL shape for checking the file types for Models conform to IFC file extensions formats

For the link confidence metadata in Table 6.2, both cardinality and datatype must be present, with the option of also defining a value range for this confidence score. Listing 6.11 shows the corresponding SHACL shape for this constraint.

```
hld:fileTypeShape
    a sh:NodeShape;
    sh:targetClass [dc:Dataset];
    sh:property [
        sh:path [
            dcterms:fileType
            ct:format
            rdf:type
            ] ;
        sh:minCount 1;
    sh:maxCount 1;
    sh:datatype [xsd:integer xsd:float]
    sh:minExclusive 0.6;
    sh:maxExclusive 0.9;
    ].
```

Listing 6.11: SHACL shape for checking the file types for Models conform to IFC file extensions formats

```
:ImageShape1
    a sh:NodeShape;
    sh:targetClass ct:linktype blink:linkType
    sh:property [
    sh:path [
        hld:confidenceScoreshape
        hld:linkAlgorithmshape
        hld:algorithmAuthorshape
        ];
    ].

hld:confidenceScoreshape
    a sh:NodeShape;
    sh:targetClass blink:confidenceScore;
    sh:property [
        sh:path schema:value;
        sh:minValue 1;
        sh:maxValue 1;
        sh:datatype [xsd:integer xsd:float];
    ].
hld:linkAlgorithmshape
    a sh:NodeShape;
    sh:targetClass blink:linkAlgorithm;
    sh:property [
        sh:path schema:value;
        sh:minValue 1;
        sh:maxValue 1;
    ].

hld:linkAuthorshape
    a sh:NodeShape;
    sh:targetClass blink:algorithmAuthor;
    sh:property [
        sh:path schema:value;
        sh:minValue 1;
    ].
```

Listing 6.12: SHACL Shape of the requirement: existence and max. 2 value for the metadata file name

In the above shape, each constraint like *Confidence score* was declared as a separate Node Shape, which was nested within the primary Node Shape focused on **linktype**.

Validation Workflow

Considering that all the generated data reside in the **index graph**, the validation workflow uses the SHACL rules defined above to check this index graph as shown in Fig. 6.2. Once the graph conforms to the set of rules, it is then exposed to the end user through the Container Layer.

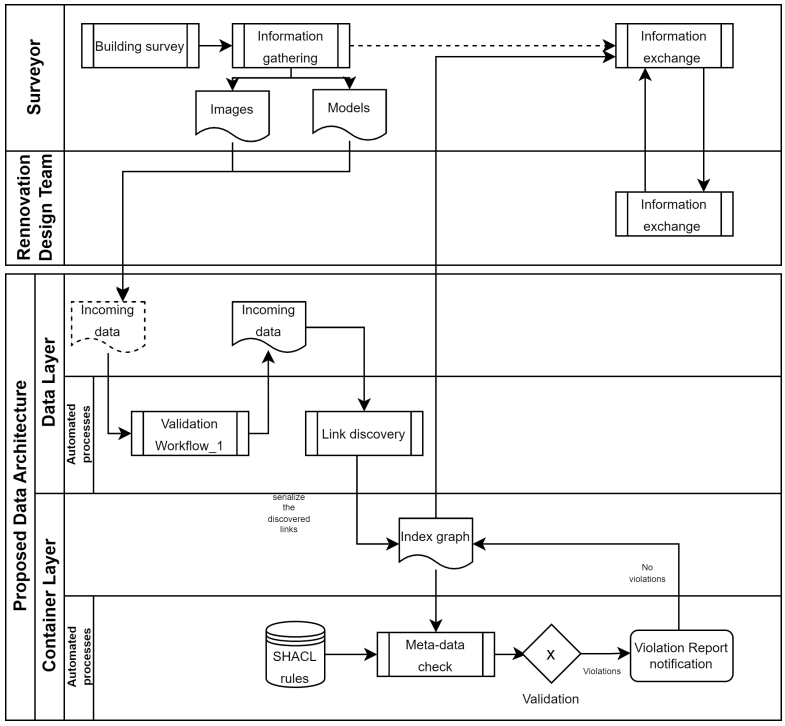

Figure 6.2: Generated data validation workflow for the proposed architecture

The above workflow follows the same process as the incoming data check and defines additional checks for the discovered and serialised links in the index graph. It leverages the SHACL shapes stored in a database and uses them for validation. Upon successful validation, it redirects back to the index graph without any notification, while in the case of violations, it generates a SHACL violation report containing the list of non-conformant triples. Similar to the previous subsection, the SHACL shapes for the requirements defined in this section are available in Appendix A.

6.4 Prototype

A web-based prototype was developed where users can load any data graph, a corresponding gITF data graph and a valid SHACL shapes graph, and outputs a SHACL validation report. During validation, the GUIDs of elements which do not conform to the shapes defined in the shapes graph are queried using SPARQL in the data graph and passed over to the gITF data graph. This prototype was built on libraries, e.g., the rdf-validate-shacl package for SHACL validation, xeokitsdk for visualising models, and React and Node.js for interface and server. Exemplary Data graphs, Shape graphs,

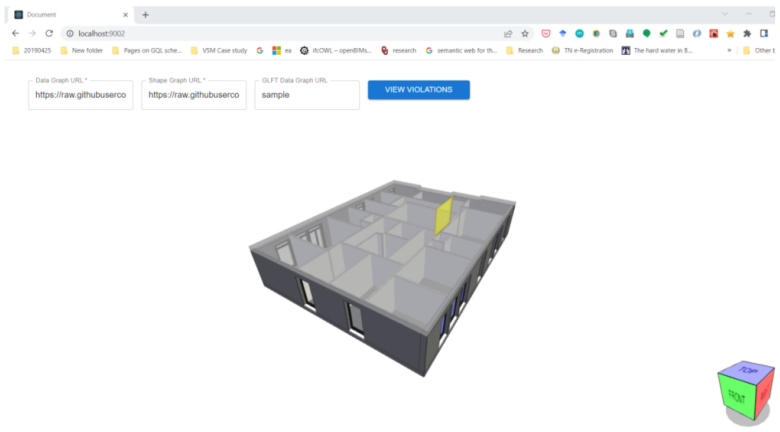

Figure 6.3: Visualizing validation errors in the prototype

and gITF Data graphs are available in GitHub[60]. When these input fields are filled and the user clicks the 'View Violations' button, an alert message with the violations is displayed. Note that the viewer uses the 3D BIM model to display violations in the images/documents too, as it leverages the link relationships to process this.

[60] https://github.com/SemanticHub/SHACLDB

Additionally, all the components which do not conform to the SHACL shapes in the Shapes graph are highlighted in yellow and displayed in the gITF model as shown in Fig. 6.3. The prototype developed is publicly available on GitHub[61].

[61] https://github.com/SemanticHub/EpicSHACLVisualizer

The technology readiness levels (TRLs) can be applied to the proposed SHACL-based conformance approach and the prototype developed for demonstrating its implementation. Conceptually, the SHACL shapes formulation in this chapter establishes the feasibility of using the language for expressing constraints for checking link relationships and container/resources metadata. The prototype development also indicates the presence of existing libraries that support the implementation of this verification approach. Hence, it falls in the range of TRL 3 - 4 (see Fig. 6.4). To evolve to TRL 9, the language has to become more easily exploitable. Since it was designed for linked data-based conformance checks, it requires high expert knowledge to apply it in practice. This knowledge is an implementation barrier, as participants within a project might not necessarily be linked data experts.

So, interfaces which can simplify the process of creating the constraints are needed. One way to achieve this is through visual programming. It utilises visual representations of code blocks for creating programs, and can be used to generate SHACL shapes without the user knowing the language's syntax/structure. This approach was explored in Senthil-

vel and Beetz, 2020, where a visual programming interface 'PyFlow' was integrated into a 3D model authoring software (FreeCAD). The paper explored the development of modules containing reusable SHACL constraints, which can be used as lego blocks for constructing SHACL shapes graph by an end-user (Senthilvel & Beetz, 2020). Due to the visual connection of these modules, no prior coding experience was required.

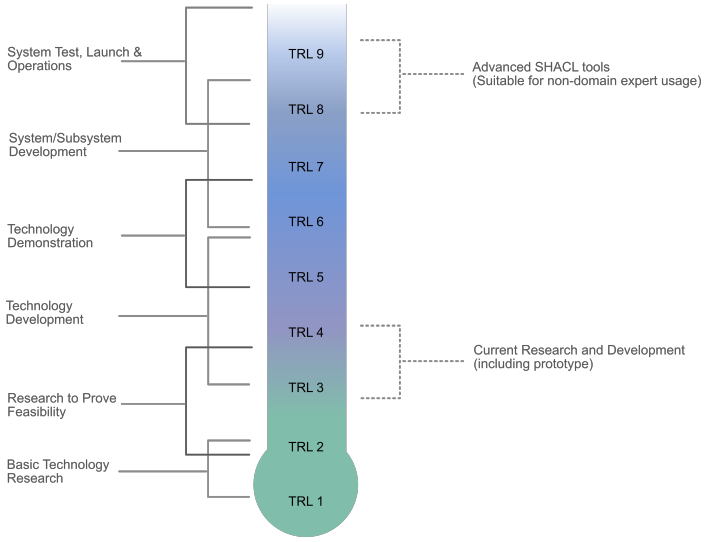

Figure 6.4: Mapping SHACL developments in this thesis to Technology Readiness Levels

Additionally, constraints are usually repetitive, although their targets change. For example, thermal transmittance values are always of the `float` data type, and can be applicable for diverse building objects like Wall, Floor, Windows, Doors etc.

Analogously to the use of SHACL constructs to validate linked building data, the SHACL shapes themselves should conform to the SHACL rule language. This helps to ascertain that all the vocabularies were used correctly as intended by the constraint language. Thus, all SHACL shapes developed in the above sections were validated using SHACL itself through an existing web implementation.

6.5 Summary and Conclusion

Conformance checks form the basis to ensure that interoperability is functioning as envisioned. Without proper validation, generated data can potentially risk being unusable, thereby impacting project decision-making. With this in

mind, this chapter focused on the further necessities of data integrity and conformance checks and briefly introduced the available languages for checking RDF-based metadata. Languages like SPARQL, SPIN, OWL, SHACL, ShEx and their usage in AEC for data validation were shown, while noting some of their limitations.

SHACL, the chosen language for validation, was introduced along with its core concepts, constructs, and syntaxes for usage. Then using SHACL, the approach adopted for checking the conformance of both the files/resources' metadata and the metadata generated by the Container Layer was introduced. For each piece of data (i.e., models, images, documents, and index graphs) in this research, the minimum requirements for data integrity, especially from a functional point of view, were also determined. These rules were then translated to SHACL shapes using its constructs.

The chapter also developed a second type of checks focusing the data generated in the proposed architecture. These included the links discovered in Chapter 4 and their associated metadata like provenance, algorithm, confidence score etc. The requirements in terms of constructs to be checked were identified, and tabulated and the corresponding SHACL shapes were developed. A comprehensive list of these shapes was also developed and is available in the Appendix A and as a database on GitHub[62].

[62] https://github.com/SemanticHub/SHACLDB

For both of the above types of checks, the relevant workflow involving the participants, their data flows and points of conformance checks were established using the BIM4Ren project as an example. Finally, a web-based validator bot was developed to visualise the results of the validation (only for models). The bot took the input of shapes graphs from a database and validated them against supplied incoming data, resulting in a 3D visualisation of violations based on the BIM model. It also generated a SHACL violations report specifying the triple nodes that violated specific constraints expressed as SHACL shape.

Chapter 7

Conclusions

The AEC industry has struggled with interoperability issues between different tools for more than three decades now: the (in)famously illustrated Islands of Automation (Matti, 1986), reiterated by numerous researchers in the last 10 years (Beetz, 2009; Poirier et al., 2014; Turk, 2020). Despite 30+ years of development, today the unchanging struggle persists, albeit in a different form. Where before, the main challenges were the lack of standardisation and formalisation of data, now it is the lack of standardisation for connecting data and its management.

To a considerable extent, the introduction and adoption of BIM and Semantic Web technologies have addressed these complex challenges related to the connecting of information and its management requirements. For example, BIM models represent geometry, spatial relationships, properties, quantities, and schedules according to schemata and are machine-interpretable. As a result, information that was previously recorded in physical/digital documents and dependent on manual evaluation can now be accurately processed without any room for ambiguity.

However, these challenges have been shown to recur in terms of managing interconnected data in increasingly distributed CDEs. For instance, information is still stored in a disconnected mode: in distributed partial models (e.g. architectural BIM model, structural BIM model, etc.) and in conventional documentation like material design specifications. Although the information presented is fragmented and contains signifiant overlapping data, it is not explicitly modelled. Additionally, the use of different authoring software has resulted in different underlying schemata. Therefore, the primary challenge is comprehending information that is im-

plicitly connected, but not explicitly stated as such. The key challenges identified in this research for these interconnected information are:

- Unified approach for detecting links between partial heterogeneous data representations
- Technical approach for persistent storage of detected links
- Functional structure for storage of detected links

The last three chapters of this thesis centered on the fundamental aspects of interconnecting information:

- development of approaches and methods for creating persistent links between some of AEC project's heterogeneous data,
- their management within a (virtual) container in a CDE and
- the corresponding conformance checks necessary to ensure its data integrity.

Each chapter attempted to answer the research questions posed in Chapter 1.

This chapter focuses on the discussion of the results obtained. It also discusses their implications for the general research question posed in Section 1.4: *How can heterogeneous data be managed in web-based information containers in a CDE?*. The research questions are reintroduced here:

- **RQ1**: *How can informal representations of related heterogeneous information be translated into formal interconnected representations that can be used throughout the asset lifecycle?*
- **RQ2**: *How can these formal interconnected representations be managed and processed within use case-based Information Containers in a CDE through open data standards?*
- **RQ3**: *How can links be checked for data integrity and conformance of these links be checked against the required schema?*

Based on the research and analysis in the preceding chapters, the next sections dive into the advantages, disadvantages, and scope of further development by comparing the developed approaches with the baseline (both theoretical and practical) established in Chapter 2. Special emphases are

given to the topics - how the proposed architecture can be integrated into existing standards and the potential paths for transferring this into practice.

7.1 Discussion

Though interoperability problems surrounding information management have existed for decades, the adoption of Semantic Web concepts has been increasingly effective in addressing the challenges of managing heterogeneous data in construction projects. The first research question posed at the beginning of this thesis was:

- **RQ1**: *How can informal representations of related heterogeneous information be translated into formal interconnected representations that can be used throughout the asset lifecycle?*

As proposed in this thesis and substantiated by previous and ongoing research, the use of knowledge discovery methods in combination with ontology matching techniques can help identify linkages between data. When these methods utilise the metadata annotations supported by Linked Data, they can effectively extract valuable information without the need for data conversion to other formats. As the strategies for link discovery continue to evolve, it is imperative that the fundamental criteria (i.e. minimum requirements) for annotation and metadata are clearly defined. These requirements were classified into two categories covering functional needs and technical needs:

Technical Requirements	**Functional Requirements**
Vocabularies for representing link relationships	Workflow for link discovery for heterogeneous resources
Vocabularies for representing annotation metadata for a resource	Conformance checks for detected links
Vocabulary for storing provenance information for resources and links	Conformance checks for minimum metadata for incoming resources
Structure for representing links and their provenance information in a container	Functioning of Information containers within a CDE

Table 7.1: Identified requirements summary for representing informal related heterogeneous information into formal interconnected representations

This is achieved using a layered approach, starting with ontologies which can capture the discovery of linked information.

Next, algorithms are employed to extract viable RDF information from non-RDF data such as images and documents. The annotation graphs and the resources (images/-documents/models) were fed into a combination algorithm, which uses lexical, string, and semantic matching methods. Finally, links discovered using this approach were stored in a virtual container, consisting of the metadata of the related resources along with themselves, and the metadata about the object-level links created. For this purpose, a dedicated ontology was created and published to describe the metadata about the links created in the containers and CDEs. Furthermore, advanced conceptual approaches like reification were used to encode the temporal aspects of the created links.

Following the results of the above, the subsequent second research question defined was:

- **RQ2**: *How can formal interconnected representations be managed and processed within use case-based Information Containers in a CDE through open data standards?*

This thesis argued that existing standards provide a conceptual base for managing heterogeneous interconnected information. To manage linked information that is regulated throughout the project lifecycle, this research proposed a three-layered information architecture:

- `Data layer`
- `Container layer`
- `Process layer`

The data-format-independent architecture allows the reuse of existing vocabularies available for link definitions, provenance, and management based on user-defined use cases. It leverages a container-based architecture, inspired by LDP and ICDD, that can be used to store, retrieve and update heterogeneous data for any use case within a project.

The literature review in this thesis also identified the minimum requirements for both the metadata for these representations and the information containers from these standards. Furthermore, it matches the necessary ontology classes and properties to describe the metadata. Leveraging a Linked Data-based linking of partially distributed datasets illustrates the advantages of active data management throughout the project lifecycle.

For processing these data inside a CDE, API protocols based

on the existing LDP and OpenCDE-APIs for the proposed architecture were defined. A microservice architecture integrated with Linked Data-based tools enables the automated exchange of heterogeneous data. With the introduction of a linking architecture which supports file-based data (images, documents), and database-based information (e.g. models, and RDF graphs), the proposed architecture supports both stages 2 and 3 of the BIM maturity model. Furthermore, its detailed definition of how information containers must be modelled within the context of web-based CDEs also complies with the standard processing of information (see Fig. 1.4).

The final research question posed in Chapter 1 refers to the conformity of all data within the scope of this research.

- **RQ3**: *How can link relationships, containers and stored data be checked for integrity and conformance requirements?*

Several past research efforts, both within the AEC and in other domains, have used the RDF-based constraint language, SHACL, to verify the conformance of data to user-defined rules and general data standards. In this research, this rule language was also used to check the conformity of the connecting graphs and the metadata encoded in RDF for images and documents. The data within this research was classified across two dimensions: metadata of incoming heterogeneous resources (models, images, documents), and container data - interlinks between heterogeneous resources, and the link provenance.

Based on RQ2, the essential metadata criteria for all of these resources were established and formally transformed into SHACL rules. To utilise the aforementioned SHACL shapes, process workflows were created by analysing data exchanges among various participants within projects.

Based on RQ2, the minimum metadata requirements for all these resources in terms of conformance checks were defined and translated into SHACL rules. Process workflows were developed for the usage of the above SHACL shapes by examining existing participant data exchanges. Furthermore, a microservice was developed to both execute these rule sets and visualise the non-conformities. SHACL was also used to assess the sufficiency of the metadata for non-RDF data. These resulted in a net constructive impact on link discovery, as incomplete metadata was weeded out prior to its addition to a container.

7.1.1 Advantages over current approaches

Fundamentally, the Semantic Web leverages the concept of a flexible and generic technology stack that allows easy representation of information and its federation with other information from various knowledge domains (Schreiber & Raimond, 2014). In general, the use of OWL ontologies built on top of RDFS vocabularies like the ones used and developed in this research increases the data and constraint expressiveness. Complex constructs, namely cardinality restrictions, class expressions, dependency-based restrictions, expand the range of conformance checks possible for linked data-oriented resources.

The approach proposed in the context of this thesis contributes to the improvement of interoperable information in the following ways:

- Incorporation and application of established guidelines from:
  - ISO 19650 for information management during project lifecycle,
  - DIN SPEC 91391, LDP for determining data and CDE requirements,
  - ICDD (ISO 21597), MMC(DIN SPE C91391) for design of Information Containers
- Segregation and abstraction of ontologies from the bespoke data architectures and software products.
- Creation of specialised ontology to describe linkages including their common types, provenance, authorship, etc.
- Design of Information Container structure that is compatible with interconnected data and provides support for its usage in a CDE.
- Identification and harmonisation of data and CDE requirements from various standards to streamline the use of interconnected information containers during design and construction phases.
- Leveraging SHACL's expressive power to develop conformity checks that ensure data integrity and interoperability.

The capability of the proposed architecture to facilitate linked data-driven container information management fulfills the

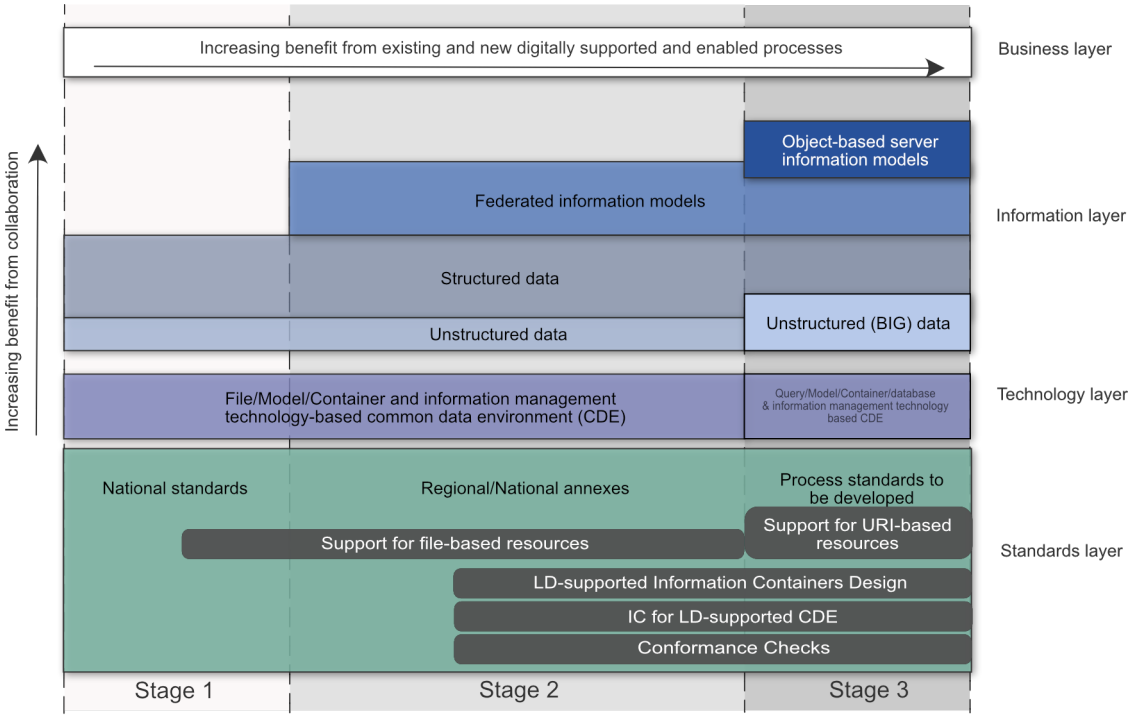

Figure 7.1: Mapping this research's contributions to the BIM Maturity model proposed in ISO 19650

requirement for a container structure and its behaviour in a CDE. This container is utilised in all in all stages of information - i.e. from `WorkInProgress` to `Archived`. The behaviour of uploaded resources and their binding to containers were modelled using API calls. These API calls defined for the interaction of the different parts of the system with other internal parts (refer to Fig. 5.10. For example, image resources stored in the `Data Layer` were bundled into containers, through interaction with the `Container Layer`.

Due to their modular design based on a microservice architecture, these APIs can also be connected to other external services, without further modifications. This functionality addresses the gap of the ICDD ISO 21597 standard, which only provided the container structure and ontologies geared towards archiving, but did not specify the mechanism of its usage in a web-based environment throughout ongoing project execution with continuously evolving datasets.

Based on these points and the BIM Maturity model introduced in Section 2, the proposed approach can be classified into applicability for Stage 2 and Stage 3. Since the latter stage does not yet have a fixed definition, all features that do not fit the Stage 2 definitions were mapped to Stage 3.

The key contributions listed above can be categorised as shown in Fig. 7.1.

7.1.2 Limitations over current approaches

The proposed approach was aimed at presenting a proof-of-concept for facilitating Information Containers hosting interconnected resources. The current research was limited to the design of the containers, their requirements and their functioning in a web-based CDE operating at Stage 3. Consequently, limitations in this research can be identified and categorised based on four major themes. These are listed in Table 7.2.

For example, additional technical and functional aspects which are part of Stage 3 were not covered. These include topics such as authorisation and credential storage for access management, role-based access of information, and reasoning-based on inference of the interconnected resources. Some of the issues encountered in the course of this research are as follows:

- Scalability issues due to slower processing of RDF graphs and OWL-based representation
- SPARQL queries for information retrieval proved to have a high learning threshold and are less intuitive. Alternatives like GraphQL-LD or HyperGraphQL by virtue of their structure can facilitate quicker and more compact querying results
- Lack of vocabulary ranges focused on defining link types in different kinds of data encountered in AEC

The algorithms used for the collection were chosen for ease of implementation, with minimal training data. Consequently, the accuracy, precision, and recall values for these methods were not considered. Rather they were used as proof of concept for detecting commonalities in heterogeneous data and served as value inputs for structuring the interconnected information within the containers.

As pointed out previously, there is a scarcity of ontologies focusing on the description of link relationships between different kinds of AEC data. Halpin et al., 2010 differentiated between four types of generic linkages for representing overlapping concepts between two resources:

- same thing as but different context

Conceptual	Scope	Data	Technical
Role-based data-restriction	simplified algorithms for link discovery	Manual annotation of images and documents	Scalability for large graphs not explored
Mapping for other container/resource metadata ontologies not created	Spatial representation of link relationships not described	..	Lack of UI/UX abstraction layer for non-domain expert usage
Provision for integration of geometrical aspects for link discovery	Assumption of Stage 3 maturity model as basis for CDE requirements	Limited data used for demonstrating proof-of-concept	prototype for demonstrating IC in CDE not developed

Table 7.2: Limitations identified in this research with their overarching theme categorisation

- same thing as but referentially opaque
- represents
- very similar to

The above four generic linkages can be used to distinguish information at a macro-level. Popular upper ontology relationships such as `owl:sameAs`, `skos:exactMatch`, `rdfs:seeAlso` are commonly used for representing generic linkages. Their partial conceptual equivalents from domain ontology like ICDD is `ct:isLinkedTo`, although this definition does not specify the degree of similarity. Generic semantic similarities can also be described Ontologies like *Similarity Ontology* (Halpin et al., 2010).

However, in the AEC context, these can be used as superclassses for the specialised link types defined in ICDD - part 2. Below, is an exemplary view of this integration [63]

[63] This exemplary view demonstrates only the possibility for a low-level generic description of links, with scope for further high-level extension. Hence, the range and domain for the terms from these ontologies are ignored.

With the increase in the combination of AI-supported approaches for link discovery, there have been discussions on the fast-blurring boundaries on AI and its influence on the Semantic Web (Hendler & Berners-Lee, 2010; Hogan, 2020; Patel et al., 2018). In this research, ML-based approaches were primarily intended for image and document annotation, by extracting viable features of interest and labelling them with appropriate object properties. However, this research does not consider the impact of full AI-based approaches on the overall linking approach.

From a technical perspective, the ontologies developed in this research align with the best practices recommended for the publication of Linked Data (Hyland et al., 2014). Additionally, each ontology was evaluated using the *OntOlogy*

Pitfall Scanner! (OOPS!) and contains publicly accessible documentation, a persistent URI-based namespace, and is available in two RDF serialisations (.ttl and .rdf). These ontologies are visualised as part of the documentation[64] using the WebVOWL tool (Lohmann et al., 2015).

[64] see online documentation

Additionally, for querying connected data in the container layer and for validating data in both the container layer and the data layer, separate microservices were also presented to view the results using the container layer.

However, multiple questions and issues remain to be solved before full exploitation of the benefits of the proposed approach can be achieved. For example, the existing knowledge/link discovery algorithms are not designed to recognise features of interest in AEC data. E.g: it is challenging to distinguish between various types of walls or windows through image recognition, despite material classification being commonly incorporated in BIM models. This research did not factor the spatial overlaps between models, which can indeed serve as additional input for link discovery. A decisive problem here was that, while individual algorithms can be used for each specific type of data (i.e., image recognition algorithms for images, OCR for documents, etc.), link discovery between heterogeneous data is not yet a heavily researched field.

In this research, approaches from the ontology matching domain were adapted for AEC data, though it is necessary to study the efficiency of these approaches for these projects. Furthermore, even if similarities between conceptual/semantic terminologies can be discovered using the adopted algorithms, there is still a huge gap in matching elements which are explicitly modelled.

For example, reinforcement bars for a column are modelled directly in a structural BIM model, without the explicit modelling of a column. In this scenario, the relationship between objects has to be discovered using spatial equivalence and implicit object relationship mapping. Implicit object relationship mapping denotes how common objects in AEC models occur w.r.t objects. For example, reinforcement bars are often contained within columns, beams, slabs, etc., electrical sockets are embedded in walls, etc. These mappings can be used to improve knowledge discovery within the AEC domain.

Naturally, all of the data represented in Linked Data-based ontologies will require corresponding interpreters to read and

infer implicit triple statements based on the ontology constructs namely as data and class type restrictions, etc. Smaller tools/services for accessing and retrieving images/documents in their native format and their RDF-based annotation metadata also have to be incorporated into CDEs.

Additionally, dedicated ontologies that address the needs of the AEC domains are negligible, with new developments in the nascent stage compared to those available other domains[65]. This necessitates the creation of new ontologies, and amalgamation of existing generic, foundational ontologies which also have to be maintained over time.

[65] In comparison, the biomedical field contains more than 400 ontologies with approximately 6 million classes(Hoehndorf et al., 2015)

Finally, the current research did not investigate the theoretical structure necessary to define access control for specific parts of the data (RDF graphs, models, images, and documents). However, recently published literature explore the usage of additional vocabularies for defining trust-able access roles and checking their patterns in incoming RDF resources (Werbrouck et al., 2020), while vocabularies such as Web Access Control and Access Control Policies can also be used for restricting sub-graph views (Werbrouck et al., 2023). Rohde et al., 2023 also presented a SHACL-based approach for Access Control using SPARQL query translation for checking the access control policies.

7.1.3 Integration into existing approaches

The BIM4Ren project introduced the concept of Linked Data principles for linking models and checking their conformance to both data requirements between interoperable tools using SHACL (Senthilvel, Krijnen, et al., 2021). The project utilised interconnected tool-chaining for delivering BIM-supported workflows for data collection and management. It used a microservice architecture for the development and implementation of a platform for the above. The SHACL-based checks form a part of data requirements and integrity checks for ensuring interoperability when it is exchanged between various tools.

At the conceptual level, this research adopts the concepts of collaborative information management, states of information, and data integrity principles from ISO 19650 part 1, while retaining the container-based information management and minimum metadata requirements from DIN SPEC 91391, and finally uses a combination of LDP and ISO 21597 ICDD concepts for both metadata requirements, ontologies for their definitions, and structure for storing data and their

links.

However, one point to note here is that part 1 of ISO 21597 explicitly prohibits the extension of the structure for any purpose other than archiving. Nevertheless, in this research, the ICDD structure has been used as an inspiration for designing LD-supported Information Containers and maintaining it through the different container states during the execution of a project. Only when these containers reach the 'archiving' state, are they exportable as ISO 21597 conformant containers.

7.1.4 Knowledge Dissemination - Research and Industry communities

Chapter 2 highlighted the significant gap between existing research and standards on one side, and the business practices and software implementations used in the industry on the other. Though the AEC industry and perhaps even the Operation and Maintenance, including renovation domains currently use extremely fragmented data collection methods and formats, they heavily rely on manual interpretation for extracting useful information for any project-related task. These tasks can range from data segregation, classification based on thematic abstractions, interpretation of contextual clues for relevance to the project, etc.

The issue of linking information resources of any form is a well-known problem, yet rarely explicitly acknowledged in the industry. One reason for this might be the current perspective of the industry on the capabilities for CDE and the framework for their maturity model. For example, ISO 19650 part 1 advocates a single CDE, which is also adopted by the industry, where all data is stored and managed. Ad-hoc links between models and documents are indeed possible in such scenarios. For instance, Revit has specialised custom fields which can be used for these. Revit models can be exported in IFC schema. Additionally, they can also be read in other tools like BIM360 or Autodesk Construction Cloud. However, these links are viewable only within the Autodesk software ecosystem. The exported IFC models will either contain this information, or will other tools like Bentley's Microstation be able to process these link information. This is because link relationships are not explicitly encoded using dedicated pset properties in the IFC schema. Consequently, link relationships are dependent on interoperable features present in specific software and can be lost during data ex-

changes.

While the BIM maturity stages provide a framework for the development of CDE capabilities, they do not capture the dimensions of development required for a truly collaborative environment. Bucher and Hall, 2020 introduces an alternative approach for evaluating CDEs, based on an n-dimensional collaborative aspect.

Bucher and Hall, 2020 also define an exemplary list of 3 dimensions of information interoperability in CDEs: 1D - between tools of one CDE, 2D - between CDEs within a project/firm, 3D - between collaborative and digital twin platforms. However, CDEs do not necessarily have to be equated with software providers. It is often the case that software providers integrate interoperability features when exchanging information between tools present in their ecosystem. For example, Revit and Navisworks can exchange authored resources without losing information.

Additionally, CDEs used with a firm have a higher potential for data exchange between them, especially if the IT landscape is based on an Enterprise Architecture[66]. Companies also invest in these interoperability developments through customisations. And finally, the exchange between different CDEs from different organisations is far more complex, as they require a higher level of interoperability. For example, where within an organisation, access can be granted using application models like Single Sign On (SSO), the same will not be feasible for multiparty access. This can arise from internal data security and confidentiality requirements.

[66] Enterprise Architecture is framework for strategic alignment of an organisation's processes, data, technology and strategy. Under its umbrella, organisations can standardise the IT infrastructure to meet business needs

Thus, the above points are taken as inspiration, and a new proposal for the development of CDE is proposed in Fig. 7.2. It adapts the n-dimensional approach suggested by (Bucher & Hall, 2020), with the following information interoperability dimensions:

- 1D - Between tools of one software provider
- 2D - Between tools used in an organisation
- 3D - Between tools used in a project
- 4D - Between different collaborative tools from different organisations

Currently, most vendor solutions fall in the category of 1-dimensional CDEs, which facilitate interoperable schema (like IFC exports), but only permit the linking of disparate information within their own software ecosystem. To evolve to

higher dimensional CDEs, environments need to be able to link information present in disparate formats, without any loss in information.

The Linked Data principles of URI-based referencing used in this research supports the linkages between different data hosted in all four dimensional levels. However, in the higher dimensions, other requirements such as organisational access-restrictions, role-based access restrictions, can play a role in determining the maturity. One way to accommodate this is to shift the CDE goal from SSoT to Single Viewpoint of Truth (SVoT). In SSoT, everyone has access to the same state of information, without restrictions. It is impractical for one piece of software to host all kinds of AEC-related information. Hence, multiple systems will inevitably host different data. A SVoT uses an interface which can incorporate the different data sources in an external interface. This acts as a viewpoint, in which each project member can access data based on their role within the project.

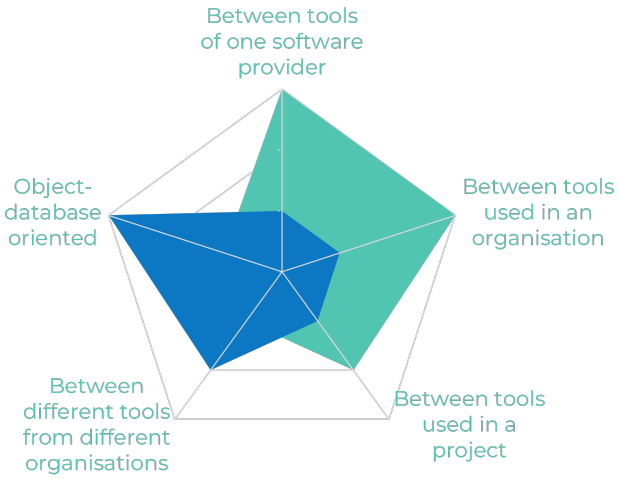

Figure 7.2: N-dimensional assessments for CDE capabilities

Using centralised systems, vendor lock-ins, regardless of which ecosystem is chosen, are inevitable. To address such interoperability risks for individual participants and the industry sector in general, the adoption of Semantic Web technologies, and in particular Linked Data principles, can help establish a framework of open data linking and management, which can then be used for building access-controlled systems.

Some of these issues can also be attributed to the ambiguity in the BIM Stage 3 definitions, which are used as the baseline for implementing Linked Data. An analysis of the

existing literature shows that, although there is some degree of requirements and expectations, it is not concrete enough to designate a full-fledged definition, separating it from perhaps future (additional) stages (Attrill & Mickovski, 2020; Britain, 2014; Shilton, 2018). This thesis contributes clarity to the role and structure of information containers needed for web-based CDEs that handle distributed information. Its development of the API calls based on open source approaches lays the foundation for a SVoT, where data can be accessed and visualised without any dependencies on their storage locations or software capabilities.

The earliest signs of change, from the software and data perspectives are the pre-standardisation efforts and initiatives in the context of buildingSMART: the buildingSMART roadmap initiative aligns well with federated data using distributed CDEs by mentioning the use of modularised Linked Data approach for facilitating storage of IFC-related data (buildingSMART, 2020). The report classifies the technologies currently available in the context of openBIM. However, a full consensus on how to facilitate CDEs that support interconnected data is yet to be explicitly defined or considered.

7.2 Conclusion

The approaches presented in this research for creating and maintaining linked heterogeneous data in a web-based Common Data Environment is a viable approach that borrows overarching concepts and specifications from existing AEC standards. It contains the conceptual framework for capturing link relationships and the practical implementation details related to CDE, such as API call definitions. The approach utilises linked data principles to achieve interoperable technical implementations of conceptual specifications (both defined in specification and also in this research).

The core of this thesis, the data architecture, is abstract enough to be used with or without the *BuildLinks Ontology* or the *Data States Ontoloty*. Instead of these ontologies, other ontologies can be used to create and store links according to the proposed link serialisation structure. The significance of this work lies in its identification of commonalities provided by the existing body of standards, and their gaps, and filling these using Linked Data & Semantic Web concepts.

The proposed approach can be extended to other heteroge-

neous data, e.g. point clouds, scanned documents and drawings, spreadsheets, scheduling files, gbXML files containing energy simulation data, etc. Additionally, the linking algorithms can be modified to be more efficient and accurate while retaining the linking structure and storage in the virtual containers. This demonstrates the flexibility of the modular approach adopted in this work. This novel approach also defines a deeper level of detail for the link storage structure.

As a consequence of using Semantic Web technologies and Linked Data methods, the usage of links between different types of project data averts data redundancy and its resulting inconsistency. Due to the use of standardised languages like SPARQL for querying, and SHACL for conformance checks, and the provision of standardised endpoints for microservices, all data within this thesis are composed following the general principles espoused by LDP and ISO 19650.

The results of this research underline the value addition in combining automated knowledge discovery methods (based on rules and ML), capturing these and managing them using Linked Data concepts, in combination with AEC requirements. This research employed data from renovation project as a usecase for demonstrating the proposed link management approach. However, these kinds of data are very similar to the data generated in green-field projects. Renovation projects share commonalities with greenfield projects interms of lifecycle and phases. Both go through design, procurement, manufacture, construction, maintenance phases, although the level of information available varies significantly due to presence/absence of prior data in renovation projects. Regardless of the type of project, the proposed architecture can be applied for storing and managing link relationships within Information Containers and use them in a CDE.

However, there is a great need for researching and implementing the full potential of the proposed approaches, in particular creating AEC-oriented link and knowledge discovery algorithms for link creation. Additionally, the scalability of the proposed approach in the context of distributed CDEs (with different access rights for different pieces of data) must be investigated in greater detail to comply with legal and contractual data security in a project.

7.3 Future Work

The architecture proposed in this thesis pushes the existing boundaries of what a BIM Stage 3 collaborative environment should look like to manage linked information. Given the current ambiguous nature of this stage, the technical obstacles in terms of theoretical and practical issues were discussed in the 7.1.2 section.

In this thesis, deep linking of heterogeneous data was achieved using a simple method. The resulting computation of the confidence score does not take into account the weights for the usage of different algorithms for images, models, and documents. Furthermore, the objective of this thesis (as stated in Chapter 4) was to lay the conceptual foundation for making deep linking possible. Considering this, the algorithms used were not investigated for their efficiency, accuracy, precision and recall values. There are a few approaches which dive into these topics. For example, Petrova et al., 2019 examined the use of AI-driven models to assess potential links using hash maps.

By deriving the proposed approach from BIM Stages 2 and 3, the organisational, cultural, and legal implications would be interesting topics to explore. By default, the exchange of information requires that all participants can openly share their data within the project ecosystem, and sometimes these are also shared with external participants e.g. municipalities, approval organisations, the general public, etc. Contractual documents such as BIM Execution Plans, Interface Coordination Plans, Exchange Information Requirements need to be adapted to accommodate interconnected data.

In these cases, many topics under active research for the conventional BIM Stage 1 & 2 should also be evaluated for the Stage 3 environment. For instance, these topics can include the legality of interconnected data ownership, governance models, liability in the event of data corruption, misuse, and continued usage as a digital twin.

Furthermore, access and authorisation control for both the data and container layers in the proposed architecture was modelled based on Open Authorization version 2.0 (OAuth2.0)[67] protocols. However, additional research is necessary for exploring these access control approaches based on roles and organisations, especially on whether Linked Data approaches can be also used for here. For example, the virtual container-based index graphs used to specify links (both file-level and

[67] An industry standard protocol used for authorisation over HTTP

deep links) between heterogeneous data account for access restrictions on specific data and their resulting usability when such data are present as links.

A good example of the above scenario is permit documents, which often contain confidential information like document owner, approver, and additional project details. In this case, only partial information should be accessible to a project's team members, and full information to the contract manager or the licensing manager. As these permit documents can be linked to the overall BIM model, a container containing both of these data should be able to allow role-based access to view only certain parts of the linked information.

Perhaps the most important work would be to concretely define the BIM Stage 3 features by relating them to the commonly encountered issues within the AEC industry. The introduction and adoption of Semantic Web technologies with BIM, though has partially solved some burning issues such as interoperability, and modularised efficient data, it also put forth new questions on data ownership, management, and legality of interconnected data changes on project participants. In light of these, newer assessments of project participants' impact should also be investigated.

Bibliography

14:00-17:00. (2018-12). *ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles*. ISO. Retrieved 2020-11-30, from https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/80/68078.html

Agency, E. E. (2020). *EU achieves 20-20-20 climate targets, 55 % emissions cut by 2030 reachable with more efforts and policies — European Environment Agency*. Retrieved 2023-03-14, from https://www.eea.europa.eu/highlights/eu-achieves-20-20-20

Agency, E. E. (2021). *Greenhouse gas emissions from energy use in buildings in Europe*. Retrieved 2023-03-14, from https://www.eea.europa.eu/data-and-maps/indicators/greenhouse-gas-emissions-from-energy/assessment

Al-Sadoon, N., & Scherer, R. (2021). IFC Semantic Extension for Dynamic Fire Safety Evacuation Simulation. *Proceedings of the 38th International Conference of CIB W78*, 184–193. https://itc.scix.net/pdfs/w78-2021-paper-019.pdf

Armijo, A., Elguezabal, P., Lasarte, N., & Weise, M. (2021). A Methodology for the Digitalization of the Residential Building Renovation Process through OpenBIM-Based Workflows. *Applied Sciences*, *11*(21), 10429. https://doi.org/10.3390/app112110429

Attrill, R., & Mickovski, S. B. (2020, September 8). Issues to be addressed with current BIM adoption prior to the implementation of BIM level 3: 36th Annual Conference on Association of Researchers in Construction Management. In L. Scott & C. J. Neilson (Eds.), *Proceedings of the 36th Annual ARCOM Conference* (pp. 335–345). ARCOM. Retrieved February 21, 2024,

from https://arcom.ac.uk/-docs/archive/2020-Indexed-Papers.pdf

Autodesk. (2022). *Harnessing the Data Advantage in Construction*. Retrieved 2023-02-22, from https://construction.autodesk.com/resources/guides/harnessing-data-advantage-in-construction

Azhar, S. (2011). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership and Management in Engineering*, *11*(3), 241–252. https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013). Towards Automated Compliance Checking in the Construction Industry. In H. Decker, L. Lhotská, S. Link, J. Basl, & A. M. Tjoa (Eds.), *Database and Expert Systems Applications* (pp. 366–380). Springer. https://doi.org/10.1007/978-3-642-40285-2_32

Beck, F., Borrmann, A., & Kolbe, T. H. (2020). The need fo ra differentiation between heterogeneous information integration approaches in the field of "BIM-GIS Integration": A Literature Review. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *VI-4-W1-2020*, 21–28. https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-21-2020

Beetz, J. (2009). Facilitating distributed collaboration in the aec/fm sector using semantic web technologies. *Eindhoven: Technische Universiteit Eindhoven*.

Bendouch, M. M., Frasincar, F., & Robal, T. (2023). A visual-semantic approach for building content-based recommender systems. *Information Systems*, *117*, 102243. https://doi.org/10.1016/j.is.2023.102243

Bird, S., Loper, E., & Klein, Ew. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Bloem, P., & De Vries, G. (2014). Machine Learning on Linked Data, a Position Paper. *Proceedings of the 1st Workshop on Linked Data for Knowledge Discovery*, 5. Retrieved 2020-05-10, from http://ceur-ws.org/Vol-1232/paper7.pdf

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

Borrmann, A., Abualdenien, J., & Krijnen, T. (2021). Information containers providing deep linkage of draw-

ings and BIM models: 38th International Conference of CIB W78. *Proceedings of the 38th International Conference of CIB W78*, *w78:2021*, 823–835. https://itc.scix.net/paper/w78-2021-paper-082

Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., & Thanh, N. L. (2012). Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry. *Future Internet*, *4*(3), 830–851. https://doi.org/10.3390/fi4030830

Box, I. (2014). *The Information Economy: A Study of Five Industries*. Retrieved 2023-01-27, from https://cdn.base.parameter1.com/files/base/acbm/fcp/document/2014/06/box-cloud-study_11535206.pdf

Britain, D. B. (2014). *Level 3 Building Information Modelling - Strategic Plan*. HM Government. UK. Retrieved February 21, 2024, from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/410096/bis-15-155-digital-built-britain-level-3-strategy.pdf

Bryde, D., Broquetas, M., & Volm, J. M. (2013). The project benefits of Building Information Modelling (BIM). *International Journal of Project Management*, *31*(7), 971–980. https://doi.org/10.1016/j.ijproman.2012.12.001

Bucher, D., & Hall, D. (2020). Common Data Environment within the AEC Ecosystem: Moving collaborative platforms beyond the open versus closed dichotomy. https://doi.org/10.3929/ETHZ-B-000447240

buildingSMART. (2020). *Technical Roadmap buildingSMART*. Retrieved 2023-06-13, from https://www.buildingsmart.org/wp-content/uploads/2020/09/20200430_buildingSMART_Technical_Roadmap.pdf

buildingSMART. (2022). *Documents API*. Retrieved 2023-05-19, from https://github.com/buildingSMART/documents-API

Caldas, C. H., Soibelman, L., & Han, J. (2002). Automated Classification of Construction Project Documents. *Journal of Computing in Civil Engineering*, *16*(4), 234–243. https://doi.org/10.1061/(ASCE)0887-3801(2002)16:4(234)

Casti, J. L. (1994). *Complexification: Explaining a Paradoxical World Through the Science of Surprise*. New York: Harper Collins.

Celoza, A., de Oliveira, D. P., & Leite, F. (2023). Qualitative Analysis of the Impact of Contracts on Information Management in AEC Projects. *Journal of Construc-*

tion Engineering and Management, *149*(3), 04022185. https://doi.org/10.1061/JCEMD4.COENG-12359

Cheng, Q., Zhang, Q., Fu, P., Tu, C., & Li, S. (2018). A survey and analysis on automatic image annotation. *Pattern Recognition*, *79*, 242–259. https://doi.org/10.1016/j.patcog.2018.02.017

Ciotta, V., Ciccone, A., Asprone, D., Manfredi, G., & Cosenza, E. (2021). Structural e-permits: An openBIM, model-based procedure for permit applications pertaining to structural engineering. *Journal of Civil Engineering and Management*, *27*(8), 651–670. https://doi.org/10.3846/jcem.2021.15784

Comission, E. (2018). *Building Information Modelling based tools & technologies for fast and efficient RENovation of residential buildings | BIM4REN Project | Fact Sheet | H2020*. CORDIS | European Commission. Retrieved 2023-07-22, from https://cordis.europa.eu/project/id/820773

Craig, N., & Sommerville, J. (2006). Information management systems on construction projects: Case reviews. *Records Management Journal*, *16*(3), 131–148. https://doi.org/10.1108/09565690610713192

Crowley, A. (1998). Construction as a manufacturing process: Lessons from the automotive industry. *Computers & Structures*, *67*(5), 389–400. https://doi.org/10.1016/S0045-7949(97)00147-8

Dankers, M., Geel, F., & Segers, N. M. (2014). A Web-platform for Linking IFC to External Information during the Entire Lifecycle of a Building. *Procedia Environmental Sciences*, *22*, 138–147. https://doi.org/10.1016/j.proenv.2014.11.014

Datta, S., Sikka, K., Roy, A., Ahuja, K., Parikh, D., & Divakaran, A. (2019). Align2Ground: Weakly Supervised Phrase Grounding Guided by Image-Caption Alignment. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2601–2610. https://doi.org/10.1109/ICCV.2019.00269

Delgado Camacho, D., Clayton, P., O'Brien, W. J., Seepersad, C., Juenger, M., Ferron, R., & Salamone, S. (2018). Applications of additive manufacturing in the construction industry – A forward-looking review. *Automation in Construction*, *89*, 110–119. https://doi.org/10.1016/j.autcon.2017.12.031

Dimyadi, J., & Amor, R. (2013). Automated Building Code Compliance Checking – Where is it at? *19th Inter-*

national CIB World Building Congress. https://doi.org/10.13140/2.1.4920.4161

DIN SPEC 91391-1 (ICS 35.240.67). (2019-02-19). DIN Deutsches Institut für Normung e. V. (German Institute for Standardization).

D'Oca, S., Ferrante, A., Ferrer, C., Pernetti, R., Gralka, A., Sebastian, R., & Op 't Veld, P. (2018). Technical, Financial, and Social Barriers and Challenges in Deep Building Renovation: Integration of Lessons Learned from the H2020 Cluster Projects. *Buildings*, *8*(12), 174. https://doi.org/10.3390/buildings8120174

Donkers, A. (2022). Ontology design template. https://doi.org/https://doi.org/10.5281/zenodo.6816899

Dubois, A., & Gadde, L.-E. (2002). The construction industry as a loosely coupled system: Implications for productivity and innovation. *Construction Management and Economics*, *20*(7), 621–631. https://doi.org/10.1080/01446190210163543

Eastman, C. M. (2011, April 19). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. John Wiley & Sons.

El-Diraby, T. E. (2014). Validating Ontologies in Informatics Systems: Approaches and Lessons learned for AEC. *Journal of Information Technology in Construction*, *19*, 474–493. https://www.itcon.org/paper/2014/28

El-Gohary, N. M., & El-Diraby, T. E. (2010). Domain ontology for processes in infrastructure and construction. *Journal of Construction Engineering and Management*, *136*(7), 730–744. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000178

Elhacham, E., Ben-Uri, L., Grozovski, J., Bar-On, Y. M., & Milo, R. (2020). Global human-made mass exceeds all living biomass. *Nature*, *588*(7838), 442–444. https://doi.org/10.1038/s41586-020-3010-5

EN 17632-1:2022 - Building information modelling (BIM) - Semantic modelling and linking (SML) - Part 1: Generic modelling patterns. (2022). Retrieved 2023-02-15, from https://standards.iteh.ai/catalog/standards/cen/512f6571-2a12-4c4f-9027-793be26b1af5/en-17632-1-2022

Eriksson, H. (2007). An Annotation Tool for Semantic Documents. *The Semantic Web: Research and Applications*. Retrieved 2023-03-25, from https://link.springer.com/chapter/10.1007/978-3-540-72667-8_54

Esser, S., Abualdenien, J., Vilgertshofer, S., & Borrmann, A. (2022). *Requirements for event-driven architectures in open BIM collaboration*. https://doi.org/10.7146/aul.455.c195

Euzenat, J., & Shvaiko, P. (2013). *Ontology Matching*. Springer. https://doi.org/10.1007/978-3-642-38721-0

F. Sowa, J. (2021). *Semantics for Interoperable Systems*. Semantics for Interoperable Systems. Retrieved 2023-07-21, from http://www.jfsowa.com/ikl/

Fuchs, S., & Scherer, R. J. (2017). Multimodels — Instant nD-modeling using original data. *Automation in Construction*, *75*, 22–32. https://doi.org/10.1016/j.autcon.2016.11.013

Gayo, J. E. L., Prud'hommeaux, E., Boneva, I., & Kontokostas, D. (2017). *Validating RDF Data*. Morgan & Claypool. Retrieved 2020-10-15, from http://www.morganclaypool.com/doi/10.2200/S00786ED1V01Y201707WBE016

Godager, B., Onstein, E., & Huang, L. (2021). The Concept of Enterprise BIM: Current Research Practice and Future Trends. *IEEE Access*, *9*, 42265–42290. https://doi.org/10.1109/ACCESS.2021.3065116

Goedert, J. D., & Meadati, P. (2008). Integrating Construction Process Documentation into Building Information Modeling. *Journal of Construction Engineering and Management*, *134*(7), 509–516. https://doi.org/10.1061/(ASCE)0733-9364(2008)134:7(509)

Greenberg, J., Sutton, S., & Campbell, D. G. (2003). Metadata: A Fundamental Component of the Semantic Web. *Bulletin of the American Society for Information Science and Technology*, *29*(4), 16–18. https://doi.org/10.1002/bult.282

Gu, N., & London, K. (2010). Understanding and facilitating BIM adoption in the AEC industry. *Automation in Construction*, *19*(8), 988–999. https://doi.org/10.1016/j.autcon.2010.09.002

Hagedorn, P. (2018). Implementation of a Validation Framework for the Information Container for Data Drop. *Tagungsband 30. Forum Bauinformatik*. https://www.researchgate.net/publication/327916596_Implementation_of_a_Validation_Framework_for_the_Information_Container_for_Data_Drop

Hagedorn, P., Block, M., Zentgraf, S., Sigalov, K., & König, M. (2022). Toolchains for Interoperable BIM Workflows in a Web-Based Integration Platform. *Applied Sciences*, *12*(12), 5959. https://doi.org/10.3390/app12125959

Hagedorn, P., & König, M. (2020). Rule-Based Semantic Validation for Standardized Linked Building Models | SpringerLink. *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*, *98*, 772–787. https://doi.org/10.1007/978-3-030-51295-8_53

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

Hagedorn, P., Senthilvel, M., Schevers, H., & Verhelst, L. B. (2023). Towards usable ICDD containers for ontology-driven data linking and link validation. *The 11th Linked Data in Architecture and Construction Workshop*. https://linkedbuildingdata.net/ldac2023/files/papers/papers/LDAC2023_paper_2079.pdf

Halaschek-Wiener, C., Golbeck, J., Schain, A., Grove, M., Parsia, B., & Hendler, J. (2005). PhotoStuff-An Image Annotation Tool for the Semantic Web. Retrieved 2023-03-19, from https://www.semanticscholar.org/paper/PhotoStuff-An-Image-Annotation-Tool-for-the-Web-Halaschek-Wiener-Golbeck/05b81a1d7c0af0f94bdc2ead71b6a2f341f6671d

Halpin, H., Hayes, P. J., McCusker, J. P., McGuinness, D. L., & Thompson, H. S. (2010). When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, & B. Glimm (Eds.), *The Semantic Web – ISWC 2010* (pp. 305–320). Springer. https://doi.org/10.1007/978-3-642-17746-0_20

Hamdan, A.-H., & Scherer, R. (2021). Areas of Interest - Semantic description of component locations for damage assessment. *EG-ICE 2021 Workshop on Intelligent Computing in Engineering*, 411. https://doi.org/10.13140/RG.2.2.32606.97607

Hendler, J., & Berners-Lee, T. (2010). From the Semantic Web to social machines: A research challenge for AI on the World Wide Web. *Artificial Intelligence*, *174*(2), 156–161. https://doi.org/10.1016/j.artint.2009.11.010

Hessel, J., Lee, L., & Mimno, D. (2019). Unsupervised Discovery of Multimodal Links in Multi-image, Multi-sentence Documents. *arXiv:1904.07826v2*, 1–17. https://doi.org/10.48550/arXiv.1904.07826

Hoare, C., Aghamolaei, R., Lynch, M., Gaur, A., & O'Donnell, J. (2022). A linked data approach to multi-scale energy modelling. *Advanced Engineering Informatics*, *54*, 101719. https://doi.org/10.1016/j.aei.2022.101719

Hoehndorf, R., Schofield, P. N., & Gkoutos, G. V. (2015). The role of ontologies in biological and biomedical research: A functional perspective. *Briefings in Bioinformatics*, *16*(6), 1069–1080. https://doi.org/10.1093/bib/bbv011

Hogan, A. (2020). The Semantic Web: Two decades on. *Semantic Web*, *11*(1), 169–185. https://doi.org/10.3233/SW-190387

Hosamo, H. H., Imran, A., Cardenas-Cartagena, J., Svennevig, P. R., Svidt, K., & Nielsen, H. K. (2022). A Review of the Digital Twin Technology in the AEC-FM Industry. *Advances in Civil Engineering*, e2185170. https://doi.org/10.1155/2022/2185170

Hu, R., Rohrbach, M., & Darrell, T. (2016). Segmentation from Natural Language Expressions. *Proceedings, Part I 14*, *9905*, 108–124. https://doi.org/10.1007/978-3-319-46448-0_7

Hyland, B., Atemezing, G., & Villazón-Terrazas, B. (2014). *Best Practoces for Publishing Linked Data*. W3C Working Group Note on Best Practices for Publishing Linked Data. https://doi.org/10.1007/978-1-4614-1767-5_2

Islam, A. Q. M. S. (2015). *Semantic annotation of tabular data in PDF documents via crowdsourcing*. Rheinische Friedrich-Wilhelms-Universitaet Bonn. Bonn. http://eis-bonn.github.io/Theses/2015/AQM_Saiful_Islam/thesis.pdf

Karan, E. P., Irizarry, J., & Haymaker, J. (2016). BIM and GIS Integration and Interoperability Based on Semantic Web Technology. *Journal of Computing in Civil Engineering*, *30*(3), 04015043. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000519

Karlapudi, J., Valluru, P., & Menzel, K. (2021). An explanatory use case for the implementation of Information Container for linked Document Delivery in Common Data Environments.

Khan, L. (2007). Standards for image annotation using Semantic Web. *Computer Standards & Interfaces*, *29*(2), 196–204. https://doi.org/10.1016/j.csi.2006.03.006

Knublacuh, H. (2022). *SHACL and OWL Compared*. Retrieved 2022-10-31, from https://spinrdf.org/shacl-and-owl.html

Krijnen, T., & van Berlo, L. (2016). Methodologies for requirement checking on building models. *Design and Decision Support Systems in Architecture and Urban Planning*, 1–11. Retrieved 2023-10-02, from https://research.tue.nl/en/publications/methodologies-for-requirement-checking-on-building-models-a-techn

Kumar, B., Cai, H., & Hastak, M. (2017). An Assessment of Benefits of Using BIM on an Infrastructure Project, 88–95. https://doi.org/10.1061/9780784481219.008

Lagazio, I. (2018). *Create a federated BIM model by collecting native models in a multiviewer environment.* Create a federated BIM model by collecting native models in a multiviewer environment | Navisworks Products | Autodesk Knowledge Network. Retrieved 2023-02-20, from https://knowledge.autodesk.com/support/navisworks-products/getting-started/caas/simplecontent/content/building-E2-80-94coordination-E2-80-94create-federated-model-different-sources.html

Lee, Y.-C., Eastman, C. M., & Lee, J.-K. (2015). Validations for ensuring the interoperability of data exchange of a building information model. *Automation in Construction*, *58*, 176–195. https://doi.org/10.1016/j.autcon.2015.07.010

Lohmann, S., Link, V., Marbach, E., & Negru, S. (2015). WebVOWL: Web-based Visualization of Ontologies. In P. Lambrix, E. Hyvönen, E. Blomqvist, V. Presutti, G. Qi, U. Sattler, Y. Ding, & C. Ghidini (Eds.), *Knowledge Engineering and Knowledge Management* (pp. 154–158). Springer International Publishing. https://doi.org/10.1007/978-3-319-17966-7_21

Maderlechner, G., Panyr, J., & Suda, P. (2006). Finding Captions in PDF-Documents for Semantic Annotations of Images. In D.-Y. Yeung, J. T. Kwok, A. Fred, F. Roli, & D. de Ridder (Eds.), *Structural, Syntactic, and Statistical Pattern Recognition* (pp. 422–430). Springer. https://doi.org/10.1007/11815921_46

Maier, F., & Fair Cape Consulting LLC. (2020). *Model Development Standards in the Construction Industry and Beyond* (UT-20.14). Retrieved 2023-02-16, from https://rosap.ntl.bts.gov/view/dot/55769

Margffoy-Tuay, E., Pérez, J. C., Botero, E., & Arbeláez, P. (2018). Dynamic Multimodal Instance Segmentation Guided by Natural Language Queries. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 656–672). Springer

International Publishing. https://doi.org/10.1007/978-3-030-01252-6_39

Martinez-Gil, J., & Aldana-Montes, J. F. (2012). An overview of current ontology meta-matching solutions. *The Knowledge Engineering Review*, *27*(4), 393–412. https://doi.org/10.1017/S0269888912000288

Matti, H. (1986). *Integration of Construction Computing Islands*.

Migilinskas, D., Popov, V., Juocevicius, V., & Ustinovichius, L. (2013). The Benefits, Obstacles and Problems of Practical Bim Implementation. *Procedia Engineering*, *57*, 767–774. https://doi.org/10.1016/j.proeng.2013.04.097

MMC Multimodell-Container. (2016). Retrieved 2023-02-15, from https://github.com/buildingSMART/MMC

Musto, C., Basile, P., Lops, P., de Gemmis, M., & Semeraro, G. (2017). Introducing linked open data in graph-based recommender systems. *Information Processing & Management*, *53*(2), 405–435. https://doi.org/10.1016/j.ipm.2016.12.003

Nawari, N. O. (2018, March 6). *Building Information Modeling: Automated Code Checking and Compliance Processes* (1st Edition). CRC Press. https://doi.org/10.1201/9781351200998

Nentwig, M., Soru, T., Ngonga Ngomo, A.-C., & Rahm, E. (2014). LinkLion: A Link Repository for the Web of Data. In V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, & A. Tordai (Eds.), *The Semantic Web: ESWC 2014 Satellite Events* (pp. 439–443). Springer International Publishing. https://doi.org/10.1007/978-3-319-11955-7_63

Oraskari, J., Beetz, J., & Senthilvel, M. (2021). SHACL is for LBD what mvdXML is for IFC. *Proceedings of the 38th International Conference of CIB W78*, 693–702. Retrieved 2023-05-21, from https://itc.scix.net/pdfs/w78-2021-paper-069.pdf

Orlandi, F., Graux, D., & O'Sullivan, D. (2021). Benchmarking RDF Metadata Representations: Reification, Singleton Property and RDF. *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 233–240. https://doi.org/10.1109/ICSC50631.2021.00049

Paris, P.-H., Hamdi, F., & Cherfi, S. S.-s. (2019). Interlinking RDF-based datasets: A structure-based approach. *Procedia Computer Science*, *159*, 162–171. https://doi.org/10.1016/j.procs.2019.09.171

Patel, P., Ali, M. I., & Sheth, A. (2018). From Raw Data to Smart Manufacturing: AI and Semantic Web of Things for Industry 4.0. *IEEE Intelligent Systems*, *33*(4), 79–86. https://doi.org/10.1109/MIS.2018.043741325

Pauwels, P. (2014). Supporting Decision-Making in the Building Life-Cycle Using Linked Building Data. *Buildings*, *4*(3), 549–579. https://doi.org/10.3390/buildings4030549

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

Pauwels, P., & McGlinn, K. (Eds.). (2022). *Buildings and Semantics: Data Models and Web Technologies for the Built Environment*. CRC Press. https://doi.org/10.1201/9781003204381

Pauwels, P., & Zhang, S. (2015). Semantic rule-checking for regulation compliance checking: An overview of strategies and approaches: 32nd International CIB W78 conference (J. Beetz, L. van Berlo, T. Hartmann, & R. Amor, Eds.). *Proceedings of the 32nd CIB W78 Conference on Information Technology in Construction*, 619–628.

Pauwels, P., Zhang, S., & Lee, Y.-C. (2017). Semantic web technologies in aec industry: A literature overview. *Automation in Construction*, *73*, 145–165. https://doi.org/https://doi.org/10.1016/j.autcon.2016.10.003

Peis, E., Morales-del-Castillo, J., & Delgado-López, J. (2008). Semantic Recommender Systems. Analysis of the state of the topic. *Hipertext.net; Núm.: 6 Edició en anglès*.

Pektaş, Ş. T., & Pultar, M. (2006). Modelling detailed information flows in building design with the parameter-based design structure matrix. *Design Studies*, *27*(1), 99–122. https://doi.org/10.1016/j.destud.2005.07.004

Petrova, E., Pauwels, P., Svidt, K., & Jensen, R. L. (2019). Towards data-driven sustainable design: Decision support based on knowledge discovery in disparate building data. *Architectural Engineering and Design Management*, *15*(5), 334–356. https://doi.org/10.1080/17452007.2018.1530092

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., & Lazebnik, S. (2015). Flickr30k En-

tities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. *Proceedings of the IEEE International Conference on Computer Vision*, 2641–2649. https://arxiv.org/abs/1505.04870

Pocobelli, D. P., Boehm, J., Bryan, P., Still, J., & Grau-Bové, J. (2018). BUILDING INFORMATION MODELS FOR MONITORING AND SIMULATION DATA IN HERITAGE BUILDINGS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-2*, 909–916. https://doi.org/10.5194/isprs-archives-XLII-2-909-2018

Poinet, P., Stefanescu, D., & Papadonikolaki, E. (2021). Collaborative Workflows and Version Control Through Open-Source and Distributed Common Data Environment. In E. Toledo Santos & S. Scheer (Eds.), *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering* (pp. 228–247). Springer International Publishing. https://doi.org/10.1007/978-3-030-51295-8_18

Poirier, E., Forgues, D., & Staub-French, S. (2014). Dimensions of Interoperability in the AEC Industry. *Construction Research Congress 2014*, 1987–1996. https://doi.org/10.1061/9780784413517.203

Polter, M., Katranuschkov, P., & Scherer, R. (2020). A Generic Workflow Engine for Iterative, Simulation-Based Non-Linear System Identifications. *2020 Winter Simulation Conference (WSC)*, 2671–2682. https://doi.org/10.1109/WSC48552.2020.9384096

Preidel, C., Borrmann, A., Oberender, C., & Tretheway, M. (2016). Seamless Integration of Common Data Environment Access into BIM Authoring Applications: The BIM Integration Framework. *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2016*, 119–128. Retrieved 2023-01-20, from https://www.taylorfrancis.com/chapters/edit/10.1201/9781315386904-20/seamless-integration-common-data-environment-access-bim-authoring-applications-bim-integration-framework-preidel-borrmann-oberender-tretheway

QUDT; Quantities, Units, Dimensions and Types. (2011). Retrieved February 25, 2024, from https://fairsharing.org/FAIRsharing.d3pqw7

Radchuk, O., Grün, K., Hartmann, T., Tomar, R., Haspel, L., Koplanovic, S., Gavin, K., Jongeling, R., Chacón Flores, R. A., Gonçalves, J., Campos, J., & Papanikolaou, V. (2021, April 9). *D6.1 Standardization plan*.

Retrieved February 16, 2023, from https://upcommons.upc.edu/handle/2117/355040
Accepted: 2021-10-29T14:40:04Z

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

Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., & Fanizzi, N. (2012). Mining the Semantic Web: Statistical learning for next generation knowledge bases. *Data Mining and Knowledge Discovery*, *24*(3), 613–662. https://doi.org/10.1007/s10618-012-0253-2

Rohde, P. D., Iglesias, E., & Vidal, M.-E. (2023). SHACL-ACL: Access Control with SHACL. In C. Pesquita, H. Skaf-Molli, V. Efthymiou, S. Kirrane, A. Ngonga, D. Collarana, R. Cerqueira, M. Alam, C. Trojahn, & S. Hertling (Eds.), *The Semantic Web: ESWC 2023 Satellite Events* (pp. 22–26). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-43458-7_4

Rubin, D. L., Mongkolwat, P., Kleper, V., Supekar, K., & Channin, D. S. (2009). Annotation and Image Markup: Accessing and Interoperating with the Semantic Content in Medical Imaging. *IEEE Intelligent Systems*, *24*(1), 57–65. https://doi.org/10.1109/MIS.2009.3

Sadeghi, M., Elliott, J. W., Porro, N., & Strong, K. (2019). Developing building information models (BIM) for building handover, operation and maintenance. *Journal of Facilities Management*, *17*(3), 301–316. https://doi.org/10.1108/JFM-04-2018-0029

Sainz, F. (2022). *BIM4Ren OSAP is launched!* Bim4Ren. Retrieved 2023-02-05, from https://bim4ren.eu/bim4ren-osap-is-launched/

Scherer, R., Katranuschkov, P., Kadolsky, M., & Laine, T. (2012). Ontology-based building information model for integrated lifecycle energy management. *ISES Deliverable D*, *8*, 24–32.

Schiller, K., Hubert, M., Faschingbauer, G., Weber, A., Demharter, J., Mueller, W., Diaz, J., Baier, C., Schapke, S.-E., & Fuchs, S. (2016). *DIN SPEC 91350:2016-11, Verlinkter BIM-Datenaustausch von Bauwerksmodellen und Leistungsverzeichnissen* (ICS 35.240.67; 91.010.20). DIN Deutsches Institut für Normung e. V. (German Institute for Standardization). https://doi.org/10.31030/2581152

Schreiber, G., & Raimond, Y. (2014). *RDF 1.1 Primer*. Retrieved 2023-05-29, from https://www.w3.org/TR/rdf11-primer/

Schroepfer, T. (2006). Global design practice: IT-based collaboration in AEC-projects. *90*.

Schulz, O., & Beetz, J. (2021). Image-documentation of existing buildings using a server-based BIM Collaboration Format workflow. *EG-ICE 2021 Proceedings*, 108–117. https://doi.org/10.18154/RWTH-2021-06884

Senthilvel, M., & Beetz, J. (2020). A Visual Programming Approach for Linked Building Data Validation. *EG-ICE 2020 Workshop on Intelligent Computing in Engineering*, 403–411. https://doi.org/http://dx.doi.org/10.14279/depositonce-9977

Senthilvel, M., Krijnen, T., & Böhms, M. (2021). *D4.3 - Data requirements and validation technologies for renovation* (Deliverable). BIM4Ren. Retrieved 2023-06-25, from https://bim4ren.eu/download/d4-3-data-requirements-and-validation-technologies-for-renovation/

Senthilvel, M., Oraskari, J., & Beetz, J. (2020). Common Data Environments for the Information Container for linked Document Delivery. *Proceedings of the 8th Linked Data in Architecture and Construction Workshop (LDAC 2020)*, *2636*, 14. https://doi.org/urn:nbn:de:0074-2636-4

Senthilvel, M., Oraskari, J. T., & Beetz, J. (2021). Implementing Information Container for linked Document Delivery (ICDD) as a micro-service. *EG-ICE 2021 Workshop on Intelligent Computing in Engineering*. https://doi.org/10.18154/RWTH-2021-08891

Shen, L., & Chua, D. (2011). Application of building information modeling (BIM) and information technology (IT) for project collaboration. *Proceedings of The International Conference on Engineering, Project and Production Management (EPPM)*, 67–76. http://www.ppml.url.tw/EPPM/conferences/2011/download/SESSION3/67_76.pdf

Shilton, M. (2018). Digital Futures – BIM in Landscape Design: A UK Perspective. *Journal of Digital Landscape Architecture 3*, 236–240. Retrieved February 21, 2024, from https://doi.org/10.14627/537642025

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, 69–82. https://doi.org/10.1016/j.autcon.2015.03.003

Soman, R. K. (2019). Linked-data based dynamic constraint solving framework to support look-ahead-planning in construction. *Proceedings of the 36th International Conference of CIB W78*, 871–880. Retrieved 2021-05-04, from https://www.researchgate.net/publication/336311906_Linked-data_based_dynamic_constraint_solving_framework_to_support_look-ahead-planning_in_construction

Spiecher, S., Arwe, J., & Malhotra, A. (2015). *Linked Data Platform 1.0*. Retrieved 2021-05-06, from https://www.w3.org/TR/ldp/

Stolk, S., & McGlinn, K. (2020). Validation of IfcOWL datasets using SHACL. *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2020*, *2636*, 91–104. https://ceur-ws.org/Vol-2636/07paper.pdf

Svetel, I., & Pejanović, M. (2010). The role of the semantic web for knowledge management in the construction industry. *Informatica*, *34*(3).

Technical Committee : ISO/TC 59/SC 13. (2020). *ISO 21597-1:2020 Information container for linked document delivery Exchange Specification - Part 1: Container*. ISO. Retrieved 2021-03-14, from https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/43/74389.html

Törmä, S. (2013). Semantic Linking of Building Information Models. *2013 IEEE Seventh International Conference on Semantic Computing*, 412–419. https://doi.org/10.1109/ICSC.2013.80

Torralba, A., Russell, B. C., & Yuen, J. (2010). LabelMe: Online Image Annotation and Applications. *Proceedings of the IEEE*, *98*(8), 1467–1484. https://doi.org/10.1109/JPROC.2010.2050290

Turk, Ž. (2020). Interoperability in construction – Mission impossible? *Developments in the Built Environment*, *4*, 100018. https://doi.org/10.1016/j.dibe.2020.100018

und Gebäudetechnik (GBG), V.-G. B. (2018). *VDI 2552 Part 5 - Building Information Modelling: Data Management* (ICS 35.240.6). erein Deutscher Ingenieuree.V. Düsseldorf. Retrieved 2023-02-21, from https://nautos.de/7TE/search

van Berlo, L., Dijkmans, T., Krijnen, T., & Roef, R. (2020). *D2.2 Framework and interface specification of BIM bots in the BIM4Ren environment* (LC-EEB-02-2018). BIM4Ren project Grant Agreement No. 820773. Re-

trieved 2023-02-27, from https://bim4ren.eu/download/d2-2-framework-and-interface-specification-of-bim-bots-in-the-bim4ren-environment/

van Berlo, L., v. Jagt, M., v. Walsum, R., Klein, W., & Mueller, I. (2016). *D2.5 REPORT ON IMPROVED USAGE OF BIM TECHNOLOGY*. EU FP7 project ELASSTIC. http://www.elasstic.eu/userdata/file/Public%5C%20deliverables/ELASSTIC-D2.5-FINAL%20Report%5C%20on%5C%20Improved%5C%20usage%5C%20of%5C%20BIM%5C%20Technology-2016.04.18.pdf

van Nederveen, S., Beheshti, R., & Willems, P. (2010). Building Information Modelling in the Netherlands: A Status Report. *W078-Special Track 18th CIB World Building Congress*, *361*, 28. Retrieved February 29, 2024, from https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=fd7b25dca1cc3697e9460014071cad58ae164353#page=33

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

Volz, J., Bizer, C., Gaedke, M., & Kobilarov, G. (2009). Discovering and Maintaining Links on the Web of Data. In A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, & K. Thirunarayan (Eds.), *The Semantic Web - ISWC 2009* (pp. 650–665). Springer. https://doi.org/10.1007/978-3-642-04930-9_41

(W3C), W. W. W. C. (2007). *The Semantic Web Layer Cake*. Retrieved 2023-03-14, from https://www.w3.org/2007/03/layerCake.png

Werbrouck, J., Taelman, R., Verborgh, R., Pauwels, P., Beetz, J., & Mannens, E. (2020). Pattern-based access control in a decentralised collaboration environment. *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2020*, *2636*. Retrieved March 1, 2024, from https://www.semanticscholar.org/paper/Pattern-based-access-control-in-a-decentralised-Werbrouck-Taelman/bb0903ee6ab82c018516ec7922908

Werbrouck, J., Schulz, O., Oraskari, J., Mannens, E., Pauwels, P., & Beetz, J. (2023). A generic framework for federated CDEs applied to Issue Management. *Advanced Engineering Informatics*, *58*, 102136. https://doi.org/10.1016/j.aei.2023.102136

Werbrouck, J., Senthilvel, M., Beetz, J., & Pauwels, P. (2019). Querying heterogeneous linked building datasets with context-expanded GraphQL queries. *Proceedings of the 7th Linked Data in Architecture and Construction Workshop, LDAC 2019*, *2389*, 21–34. Retrieved 2023-02-16, from http://hdl.handle.net/1854/LU-8623179

Werbrouck, J., Tarkiewicz, M., Senthilvel, M., Beetz, J., & Weise, M. (2019). *D1.4 Overall Requirements for the integration of tools & services in the BIM4Ren digitalised renovation workflows*. BIM4Ren Research Project.

Wilkinson, M. D., Dumontier, M., J Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 160018. https://doi.org/10.1038/sdata.2016.18

Wood, H. L., Piroozfar, P., & Farr, E. R. P. (2013). UNDERSTANDING COMPLEXITY IN THE AEC INDUSTRY. *Procs 29th Annual ARCOM Conference*, 859–869. Retrieved 2022-10-27, from https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=09659795b458470c5f7fc24edd9188cc18edaba4

Yalcinkaya, M., & Singh, V. (2015). Patterns and trends in Building Information Modeling (BIM) research: A Latent Semantic Analysis. *Automation in Construction*, *59*, 68–80. https://doi.org/10.1016/j.autcon.2015.07.012

Yang, C., Dong, M., & Fotouhi, F. (2005). Region based image annotation through multiple-instance learning. *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 435–438. https://doi.org/10.1145/1101149.1101245

Ye, X., Sigalov, K., & König, M. (2020). *Integrating BIM- and Cost-included Information Container with Blockchain for Construction Automated Payment using Billing Model and Smart Contracts*. https://doi.org/10.22260/ISARC2020/0192

Zhang, C., Beetz, J., & de Vries, B. (2018). BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. *Semantic Web*, *9*(6), 829–855. https://doi.org/10.3233/SW-180297

Appendix A

Mapping between ontologies

```
1
2  # Copyright 2020-2023 Madhumitha Senthilvel.
3
4  # This work is licensed under a Creative Commons Attribution License.
5  # This copyright applies to the Vocabulary Specification and
6  # accompanying documentation in RDF. Regarding underlying technology,
7  # the Vocabulary uses W3C's RDF technology, an open Web standard that can
       be freely
8  # used by anyone.
9
10 @prefix : <https://w3id.org/bot/IFCOWL4_ADD2Alignment#> .
11 @prefix owl: <http://www.w3.org/2002/07/owl#> .
12 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
13 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
14 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
15 @prefix dcterms: <http://purl.org/dc/terms/> .
16 @prefix vann: <http://purl.org/vocab/vann/> .
17 @prefix voaf: <http://purl.org/vocommons/voaf#> .
18 @prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
19 @prefix schema: <https://schema.org/> .
20 @prefix ct: <https://standards.iso.org/iso/21597/-1/ed-1/en/Container> .
21 @prefix ldp: <http://w3.org/ns/ldp#> .
22 @base <https://w3id.org/bot/IFCOWL4_ADD2Alignment> .
23
24 <https://w3id.org/bot/IFCOWL4_ADD2Alignment> rdf:type owl:Ontology ,
       voaf:Vocabulary ;
25     dcterms:title "ISO 21597 to LDP alignment"@en ;
26     dcterms:description """This ontology defines proposed alignments with the
           ISO 21597 ICDD ontology"""@en ;
27     dcterms:issued "2022-12-04"^^xsd:date ;
28     dcterms:modified "2022-12-25"^^xsd:date ;
29     dcterms:license <http://creativecommons.org/licenses/by/3.0/> ;
30     owl:versionInfo "v1.0.0" ;
31     owl:versionIRI <github> ;
32     owl:priorVersion <github> ;
33     rdfs:seeAlso <xxx> ;
34     dcterms:creator <https://orcid.org/0000-0003-0733-9157> ;
35     dcterms:creator "Madhumitha Senthilvel" ;
36     owl:imports   <https://standards.iso.org/iso/21597/-1/ed-1/en/Container
           > ,
37                   <http://w3.org/ns/ldp#> .
38
39 dcterms:title a owl:AnnotationProperty .
40 dcterms:description a owl:AnnotationProperty .
41 dcterms:issued a owl:AnnotationProperty .
42 dcterms:modified a owl:AnnotationProperty .
```

```
43 dcterms:creator a owl:AnnotationProperty .
44 dcterms:contributor a owl:AnnotationProperty .
45 dcterms:license a owl:AnnotationProperty .
46 vann:preferredNamespacePrefix a owl:AnnotationProperty .
47 vann:preferredNamespaceUri a owl:AnnotationProperty .
48 vcard:fn a owl:AnnotationProperty .
49 schema:name a owl:AnnotationProperty .
50
51 <https://orcid.org/0000-0003-0733-9157> a vcard:Individual , schema:Person;
52     vcard:fn      "Madhumitha Senthilvel" ;
53     schema:name "Madhumitha Senthilvel" .
54
55 ####################
56 #    Classes
57 ####################
58 ct:Container rdfs:subClassOf ldp:Container.
59 ct:InternalDocument rdfs:subClassOf ldp:Resource.
60
61 ####################
62 #    Properties
63 ####################
64 ct:ContainsDocument rdfs:equivalentProperty ldp:contains,
65                                             ldp:hasMemberRelation,
66                                             lpd:isMemberOfRelation.
```

Appendix B

List of SHACL shapes

courier

```
hld:fileNameShape
 sh:NodeShape;
 sh:targetClass [dc:Image, dc:Dataset];
 sh:property [
  sh:path [dc11:fileName ct:fileName ct:fileType
      rdf:type] ;
  sh:minCount 1;
  sh:maxCount 1;
 ].
```

Listing B.1: SHACL shape for checking the file types for Models conform to IFC file extensions formats

```
hld:fileNameShape2
 sh:NodeShape;
 sh:targetClass [ifcowl:IfcWallStandardCase, ifcowl
     :IfcWall, ifcowl:IfcStructuralWall, ifcowl:
     Window, ifcowl:Door];
 sh:property [
  sh:path [dc11:fileName ct:fileName ct:fileType
      rdf:type] ;
  sh:minCount 1;
  sh:maxCount 1;
 ].
```

Listing B.2: SHACL shape for checking that there exists only one value for Thermal Transmittance for all Windows Doors Walls

```
hld:imagelinkDiscoveryShape
 sh:NodeShape;
 sh:targetClass [dc:Image, dc:Dataset];
 sh:property [
  sh:path [ns1:annotationRegion] ;
  sh:minCount 1;
 ],
sh:property [
```

```
 sh:path [ns1:hasRegion] ;
 sh:minCount 1;
 ],
 sh:property [
 sh:path [omg:hasSimpleGeometry];
 sh:minCount 1;
 ].
```

Listing B.3: SHACL shape for checking the Image contains necessary annotations for link discovery

```
hld:authorshipShape
 sh:NodeShape;
 sh:targetClass [dc11:Document ct:InternalDocument
    ct:ExternalDocument dc11:Image dcat:Model];
 sh:property [
  sh:path [ct:createdBy ct:createdOn schema:
     createdOn];
  sh:minCount 1;
  sh:maxCount 1;
 ].
```

Listing B.4: SHACL shape for checking the incoming images models documents contain authorship metadata

```
hld:filetypeShape
 sh:NodeShape;
 sh:targetClass [dc11:image ct:InternalDocument ct:
    ExternalDocument dc11:document dcat:document
    dc11:model];
 sh:property [
  sh:path [dc11:fileName ct:fileName ct:fileType] ;
  sh:minCount 1;
  sh:maxCount 1;
 ],
 sh:property[
  sh:path [ct:description schema:description];
  sh:minCount 1;
 ].
```

Listing B.5: SHACL shape for checking the image/document/model has a file name/file type and description associated with it

```
hld:ModelShape1
 sh:NodeShape;
 sh:targetClass [ifcOWL:IfcWallStandardCase bot:
    Wall beo:Wall ifcOWL:IfcWall];
 sh:property [
  sh:path [ifcOWL:thermaltransmittance bot:
     thermaltransmittance s4bldf:
     thermaltransmittance] ;
  sh:minCount 1;
  sh:maxCount 1;
  sh:minValue 1.2;
  sh:maxValue 2.5;
```

```
  sh:message "Thermal transmittance value is not
      within the acceptable limits"
 ].
```

Listing B.6: SHACL shape for checking the existence of a property for thermal transmittance and its value is within permissible limits

```
hld:ImageShape1
   a sh:NodeShape;
 sh:targetClass ct:linktype lol:linkType
 sh:property [
  sh:path hld:confidenceScoreshape hld:
      linkAlgorithmshape hld:algorithmAuthorshape
 ].

hld:confidenceScoreshape
 a sh:NodeShape;
sh:targetClass lol:confidencescore;
sh:property
[
sh:path schema:value;
sh:minValue 1;
sh:maxValue 1;
sh:datatype [xsd:integer xsd:float];
].

hld:linkAlgorithmshape
 a sh:NodeShape;
 sh:targetClass lol:linkAlgorithm;
sh:property [
sh:path schema:value;
sh:minValue 1;
sh:maxValue 1;
].

hld:linkAuthorshape
 a sh:NodeShape;
 sh:targetClass lol:AlgorithmAuthor;
sh:property [
  sh:path schema:Value[
    sh:minValue 1;
].
```

Listing B.7: SHACL shape for checking the metadata generated by link discovery

```
hld:fileTypeModelShape
 sh:NodeShape;
 sh:targetClass [dc11:Model];
 sh:property [
  sh:path [dc11:fileType ct:format] ;
  sh:minCount 1;
  sh:maxCount 1;
  sh:value ["application/x-extension-ifc"]
 ].
```

Listing B.8: SHACL shape for checking the file types for Models conform to IFC file extension formats

```
hld:ImageTitleShape
 sh:NodeShape;
 sh:targetClass [dc:Image];
 sh:property [
  sh:path [dc11:title] ;
  sh:minCount 1;
  sh:maxCount 1;
  sh:datatype xsd:string;
 ].
```

Listing B.9: SHACL shape for checking that images contain title metadata as a literal

```
hld:createdLinksShape
 sh:NodeShape;
 sh:targetClass [lol:linktype ct:linktype];
 sh:property [
  sh:path [dc11:creator] ;
  sh:minCount 1;
  sh:maxCount 1;
  xsd:datatype schema:Person;
 ].
```

Listing B.10: SHACL shape for checking that the created links have an author

```
hld:documentShape
 sh:NodeShape;
 sh:targetClass foaf:Document;
 sh:property [
  sh:path [nif:annotation dc11:description dc11:
     title] ;
  sh:minCount 1;
 ].
```

Listing B.11: SHACL shape for checking that the documents contain annotations and a description

```
hld:documentShape2
 sh:NodeShape;
 sh:targetClass [foaf:Document];
 sh:property [
  sh:path nif:Annotation ;
  sh:node [hld:docAnnotationshape];
 ].

hld:docAnnotationshape
 a sh:NodeShape;
 sh:property[
 sh:path [nif:confidence ao:hasTopic prov:
```

```
    generatedAt];
13    sh:minCount 1;
14    sh:maxCount 1;
15  ].
```

Listing B.12: SHACL shape for checking that the document's annotation contains a confidence score, topic and provenance information

```
1  hld:imageAnnotationShape
2   sh:NodeShape;
3   sh:targetClass [foaf:image];
4   sh:property [
5    sh:path [geo:hasGeometry omg:hasGeometry] ;
6    sh:minCount 1;
7    sh:node hld:geometryShape;
8   ].
9
10 hld:geometryShape
11  a sh:NodeShape;
12  sh:targetClass [geo:hasGeometry omg:hasGeometry];
13  sh:property [
14    sh:path [geo:WKT dc11:title nif:confidence prov:
        generatedAtTime];
15    sh:minCount 1;
16    sh:maxCount 1;
17  ].
```

Listing B.13: SHACL shape for checking that all images contain annotated regions, with well-defined (non-empty) coordinates, an annotation title, provenance information

```
1  hld:imageMetadataShape
2   sh:NodeShape;
3   sh:targetClass foaf:image;
4   sh:property [
5    sh:path [dc11:description dc11:title] ;
6    sh:minCount 1;
7   ].
```

Listing B.14: SHACL shape for checking that all images contain at least one description and one title

```
1  hld:imageCreationDataShape
2   sh:NodeShape;
3   sh:targetClass [foaf:image];
4   sh:property [
5    sh:path [dc11:createdOn dc11:creator] ;
6    sh:minCount 1;
7    sh:maxCount 1;
8   ].
```

Listing B.15: SHACL shape for checking the presence of inherent metadata of images such as creation date and creator

```
hld:documentMetadataShape
 sh:NodeShape;
 sh:targetClass [dc:Document, ct:InternalDocument];
 sh:property [
  sh:path [dc11:fileName ct:fileName ct:fileType
      rdf:type] ;
  sh:minCount 1;
  sh:maxCount 1;
 ].
```

Listing B.16: SHACL shape for checking the documents and their file name, file type

Appendix C

Prefixes

ao	:	http://purl.org/ontology/ao/core#
arp	:	http://www.arpenteur.org/ontology/Arpenteur.owl#
beo	:	https://pi.pauwel.be/voc/buildingelement#
blink	:	https://w3id.org/blink#
bot	:	https://w3id.org/bot#
cs	:	https://w3id.org/cs#
ct	:	https://standards.iso.org/iso/21597/-1/ed-1/en/Container#
DBpedia	:	http://dbpedia.org/resource/classes#
dc11	:	http://purl.org/dc/elements/1.1/
dcterms	:	http://purl.org/dc/terms/
dicom	:	http://purl.org/healthcarevocab/v1#
els	:	https://standards.iso.org/iso/21597/-2/ed-1/en/ExtendedLinkSet#
ex	:	www.example.com/rwth/dc/ms-thesis/ex#
exif	:	http://www.w3.org/2003/12/exif/ns#
express	:	https://w3id.org/express#
foaf	:	http://xmlns.com/foaf/0.1/
fog	:	https://w3id.org/fog#
geo	:	http://www.opengis.net/ont/geosparql#
hld	:	http://www.example.com/rwth/dc/ms-thesis/hld#
imo	:	http://imgpedia.dcc.uchile.cl/ontology#
inst	:	www.example.com/rwth/dc/ms-thesis/inst#
inst1	:	www.example.com/rwth/dc/ms-thesis/inst1#
inst2	:	www.example.com/rwth/dc/ms-thesis/inst2#
inst3	:	www.example.com/rwth/dc/ms-thesis/inst3#
instA	:	www.example.com/rwth/dc/ms-thesis/instA#
instS	:	www.example.com/rwth/dc/ms-thesis/instS#
inst4	:	www.example.com/rwth/dc/ms-thesis/inst4#
lio	:	http://purl.org/net/lio#
nif	:	http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#
omg	:	https://w3id.org/omg#
opm	:	http://www.w3id.org/opm#
owl	:	http://www.w3.org/2002/07/owl#

props	: https://w3id.org/props#
prov	: http://www.w3.org/ns/prov#
qudt	: http://qudt.org/schema/qudt#
rdfs	: http://www.w3.org/2000/01/rdf-schema#
rdf	: http://www.w3.org/1999/02/22-rdf-syntax-ns#
s4bldg	: https://w3id.org/def/saref4bldg#
s4watr	: https://w3id.org/def/saref4watr#
saref	: https://w3id.org/saref#
schema	: http://schema.org/
seas	: https://w3id.org/seas/
sh	: http://www.w3.org/ns/shacl#
skos	: http://www.w3.org/2004/02/skos/core#
stat	: http://www.w3.org/ns/posix/stat#
void	: http://rdfs.org/ns/void#
vra	: http://simile.mit.edu/2003/10/ontologies/vraCore3#