The European Master in Building Information Modelling is a joint initiative of:

Universidade do Minho

Univerza v Ljubljani

POLITECNICO MILANO 1863

Compliance checking for construction project data with linked data

Marin Ljuban

BIM A+

UMinho | 2023

Universidade do Minho
Escola de Engenharia

Marin Ljuban

Compliance checking for construction project data with linked data

BIM A+ European Master in Building Information Modelling

Co-funded by the Erasmus+ Programme of the European Union

September 2023

Universidade do Minho
Escola de Engenharia

Marin Ljuban

Compliance checking for construction project data with linked data

BIM A+ European Master in Building Information Modelling

Master Dissertation
European Master in Building Information Modelling

Work conducted under supervision of:
José Luís Duarte Granja
José Carlos Basto Lino
Mathias Bonduel (Tutor in company)

Co-funded by the Erasmus+ Programme of the European Union

September, 2023

AUTHORSHIP RIGHTS AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES

ACKNOWLEDGEMENTS

First and foremost, I thank the BIM A+ Consortium and the European Commission for granting me the Erasmus Mundus Scholarship, thus making this past year possible.

I thank Mr. Jose Carlos Basto Lino for teaching me much more about life than an average mentor would and should. I thank the whole Neanex team and especially dr. ir. Mathias Bonduel for invaluable guidance and patience in answering many questions throughout the past months. I thank Maria Laura Leonardi for thorough checking of the dissertation text on multiple occasions.

A big thank you to all the people I've met in the past year for making it unforgettable, especially Marija and Rui for making Portugal feel like home.

Finally, a big thank you to my friends and family back home for their support and understanding.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

RESUMO

O sector da construção tem assistido a um aumento da digitalização nos últimos anos, levando à produção de quantidades cada vez maiores de dados. Estes dados estão frequentemente isolados em sistemas e formatos de ficheiros específicos, o que dificulta a sua ligação numa representação holística de um bem construído. A investigação existente sugere que esta lacuna entre silos díspares pode ser colmatada utilizando Semantic Web Technologies (SWT). A aplicação das SWT resulta numa representação formal e explícita do conhecimento, tornando possível assegurar a verificação da conformidade dos dados do projeto de construção provenientes de diferentes fontes. Embora promissora, a escalabilidade da aplicação das SWT no ambiente construído foi prejudicada pela falta de diretrizes de modelação claras. Tais diretrizes foram recentemente introduzidas no mercado europeu sob a forma de uma norma, a EN 17632-1:2022: Building Information Modelling (BIM) - Semantic Modelling and Linking (SML), que prescreve um modelo de informação de nível superior e um conjunto de padrões de modelação da informação. Considerando que o modelo e formato de informação Industry Foundation Classes (IFC) é atualmente o padrão de intercâmbio mais utilizado para dados BIM, esta dissertação irá analisar e apresentar os meios de expressar tanto o esquema IFC como os conjuntos de dados IFC de acordo com as diretrizes especificadas na norma EN 17632. Para a primeira contribuição, o modelo de informação de nível superior SML é alargado com conceitos do esquema IFC. Em segundo lugar, os conjuntos de dados IFC convencionais são convertidos num gráfico de dados ligados aplicando a versão baseada em SML do esquema IFC, bem como os padrões de modelação da informação SML. Numa incursão no final da investigação, são exploradas as possibilidades de verificação da conformidade dos conjuntos de dados IFC convertidos, primeiro em teoria e depois na aplicação prática num contexto mais vasto do processo de licenciamento digital de edifícios.

Palavras chave: Semantic Modelling and Linking, IFC, programação, Semantic Web Technologies, Licenciamento Digital, Verificação da Conformidade

ABSTRACT

The construction industry has seen an increase in digitalisation in recent years, leading to production of ever larger amounts of data. This data is often siloed in specific systems and file formats, making it hard to connect it in a holistic representation of a built asset. Existing research suggests that this gap between disparate silos could be bridged using Semantic Web Technologies (SWT). Application of SWT results in a formal and explicit knowledge representation, making it possible to ensure compliance checking of construction project data coming from different sources. While promising, scalability of SWT application in the built environment was hindered by the lack of clear modelling guidelines. Such guidelines were recently introduced to the European market in the form of a standard, EN 17632-1:2022: *Building Information Modelling (BIM) – Semantic Modelling and Linking (SML),* prescribing a Top Level Information Model and a set of information modelling patterns. Considering that the Industry Foundation Classes (IFC) information model and format is currently the most used exchange standard for BIM data, this dissertation will analyse and present the means of expressing both the IFC schema as well as IFC datasets according to the guidelines specified in the EN 17632 standard. For the first contribution, the SML Top Level Information Model is extended with concepts from the IFC schema. Secondly, conventional IFC datasets are converted to a Linked Data graph applying the SML-based version of the IFC schema as well as the SML information modelling patterns. In an expedition at the end of the research, the possibilities of compliance checking over the converted IFC datasets are explored, first in theory and then in practical application in a wider context of digital building permitting process.

Keywords: Semantic Modelling and Linking (SML), IFC, programming, Semantic Web Technologies, digital building permitting, compliance checking

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The motivation for this research stems from a very complex answer to a very simple question. The question goes as follows: Why is Building Information Modelling (BIM), despite many proven benefits, already not business as usual? The answer is multi-faceted, complex, and even just a thorough analysis of the stated question from different perspectives would surpass the scope of a master's thesis. However, if there is one statement that would describe these issues most coherently, it would be that the process of BIM implementation requires a complete redesign of the built asset management process. As shown in Fig. 1.1, the construction industry is by far the most fragmented industry out of the five analysed. Such fragmentation makes the implementation of BIM methodology in the whole life-cycle a great challenge from technical and social perspective.

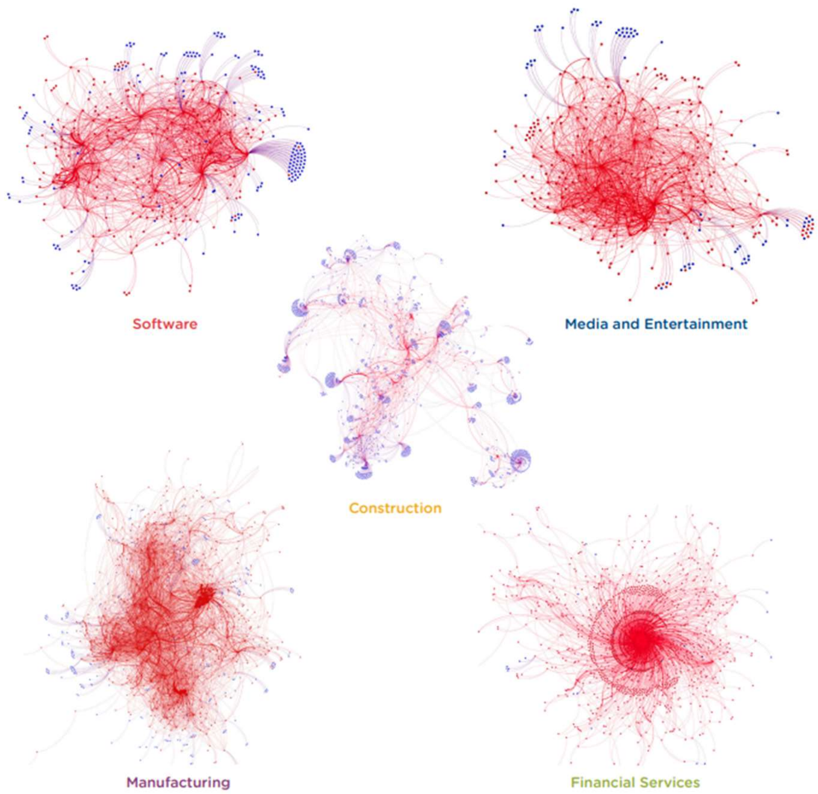

Figure 1.1 Visualization of stakeholder interaction in studied five industries (*The Information Economy: A Study of Five Industries*)

Having that in mind, Fig. 1.2 shows a comparison between idealized digital workflows and conventional workflows. It clearly shows distinct milestones where most of the information loss occurs. The information loss in conventional workflows first happens due to the designer change in moving from the conceptual design to the detailed design stage, and again when the actual construction begins. Since conventional workflows commonly do not pay close attention to information management strategy, handover is usually done in a traditional, paper-based way that results in additional information loss.

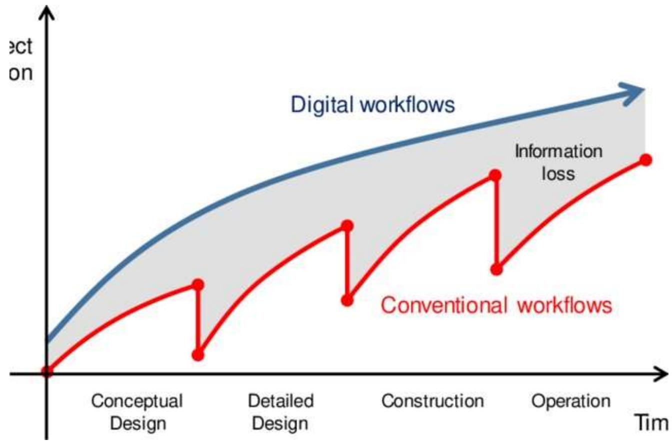

Figure 1.2 Comparison between an idealized digital and conventional workflow (Borrmann et al., 2018)

Fig. 1.2 also shows clearly that the actual implementation of digital workflows needs to start in the design process. That is the current state of the art on BIM. However, even if digital workflows are applied, the problem that remains is that the main deliverables are still very commonly paper deliverables such as tables and 2D drawings. Therefore, digital workflows are just a tool in the design process to produce the same results as before, and it is logical that designers are reluctant to move to digital workflows since this brings additional work for them without bringing any significant benefits. However, to bring additional benefits to the designers the whole construction process would need to go through significant changes, for which the incentive should come from the appointing party. To overcome the information loss in the design stage, a well-organized digital information management workflow should be established. Due to the high complexity and fragmentation of the construction industry, it is commonly not feasible and realistic to simply demand BIM in the whole life cycle. It is, therefore, useful to examine which benefits the appointing party could have by this, partial implementation of BIM in the first phase of a construction project.

Looking at it from a business perspective, one of the benefits that the appointing party could have only from the designer's model without involving the downstream actors is faster regulatory compliance checking from public authorities. To legally develop a construction project, the project needs to be reviewed and issued a building permit. The current state of the art of permitting process, although already lengthy and inefficient, is severely challenged by growing number of building projects on one side, and the lack of qualified personnel on the other (Fauth and Soibelman, 2022). The report (*Dealing with construction permits*, 2020) by the World Bank pointed out differences between different areas of the world by tracking the procedures, time, and costs of obtaining regulatory compliance to build a warehouse. The best result is achieved in the countries belonging to the Organisation for Economic Co-operation and Development (OECD), having the lowest number of procedures (12.7) and percentage of total cost (1.5%), but still takes a fairly long time to get approved (152.3 days). Examining the differences in the OECD, big discrepancies are visible. In example, Denmark has an efficient (64 days) and inexpensive process (0.6%) process with a small number of procedures (7). On the other hand, Belgium also has a process that cannot be considered expensive (0.9%) and does not involve a great number of procedures (10) but takes a long time (212 days). That can result in investment uncertainties which hinder real-estate developments. In short, obtaining building permits is an unstandardized process, often lengthy and uncertain and still mostly done manually in processes fragmented across many public institutions (Fauth and Soibelman, 2022). Therefore, its digitalisation should bring benefits to both public institutions issuing the permits, as well as to the appointing parties requesting them. Digital Building Permit is hence seen as a priority in many public initiatives throughout the world.

An important part of the digital building permitting (DBP) process is automation of compliance checking. Multiple approaches to automated compliance checking (ACC) exist (Zhang et al., 2022). When deciding which method of ACC should this research try to contribute, it is important to holistically consider interoperability constraints that could appear as the BIM model is used in different contexts.

For instance, to ensure proper urban planning, the delivered BIM model would need to be integrated into a more consolidated 3D city model that consist of multiple BIM models, as well as Geographic Information System (GIS) terrain models (Harrie and Jensen, 2018). Additionally, thermal characteristics could be assessed by checking the gbXML model. Moreover, the model used for compliance checking could be used in the latter stages of the construction project, on the construction site and even facility management phase of the project. The process should therefore need to be flexible enough to integrate with additional data sources as they appear and develop based on new use - cases. Table 1.1 contains just some of the possible sources of information that could be assessed in the digital building permit process.

Table 1.1 Overview of some sources of information in the building permitting process

Data Schema	File format	Summary
Industry Foundation Classes	STEP, TTL, JSON..	Standardised digital description of the built environment, including buildings and civil infrastructure; including objects, processes and people
CityGML	XML	Standardised data model and exchange format for 3D models of cities and landscapes
CityJSON	JSON	JSON implementation of the CityGML; similar application as CityGML
gbXML	XML	Transfer of relevant building data (BIM) to engineering analysis tools. Enables interoperability between BIM and building performance simulation

Linked Data, as a subset of Semantic Web Technologies (SWT) provides concepts and technical standards to overcome interoperability constraints between various data formats only loosely related to each other (Borrmann et al., 2018). Linked data has already seen application in a wide variety of use cases in the building life cycle. Still mostly academic and research oriented, these include proof-of-concept applications from compliance checking in the design stage (Kovacs and Micsik, 2021) and construction management (Schlachter, 2020) up to the and operations and maintenance phase (Chamari et al., 2022). In addition to newly designed buildings, it can also be used for semantic modelling of existing buildings with a special focus on particularly valuable historic buildings through the application of the Heritage BIM concept (Bonduel, 2021), (Werbrouck, 2017). Due to this versatility and adaptability to integrate heterogenous data from many different use cases, linked data is considered to be one of the promising means of bridging the interoperability gap in the project life-cycle (Borrmann et al., 2018). However, the versatility and adaptability can lead to different representations of the same concepts that, instead of promoting standardisation and interoperability actually make it harder to repeat patterns from one project to another. An attempt to standardise information management in construction projects utilising BIM is the new EN standard, EN 17632-1:2022 Building information modelling (BIM) -- Semantic modelling and linking (SML) -- Part 1: Generic modelling patterns. This standard contains a generic, top-level construction ontology, as well as guidelines about modelling patterns and application of levels of semantic modelling capability for different use cases (vocabulary, basic/advanced ontology) (Koehorst et al., 2022).

The research gap identified is that, to the best of this research's knowledge, there has been no attempt to express the core data schema in digital construction industry internationally standardized (ISO 16739-1:2018) Industry Foundation Classes (IFC) data schema, in relation to the EN 17632-1:2022 standard. This research will therefore explore how it would be possible to connect the two open standards. An additional research contribution will be to explore validation of such datasets in the context of digital building permits by utilisation of the Shapes Constraint Language (SHACL).

To address the stated knowledge gap the principle of knowledge gathering and dissemination in an international environment was employed as much as possible. This was achieved through collaboration with Neanex, a company based in Antwerp, Belgium. Neanex specialises in application of linked data in construction projects and their inputs, mainly voiced through regular meetings with Dr. Ir. Mathias Bonduel played a crucial role in shaping this whole research. Besides regular, bi-weekly meetings with Mr Bonduel and the faculty mentor, Mr Lino, several one-time events happened that shaped the research as well. These will be listed in chronological order. Firstly, a meeting with Mr Rui De Klerk, a researcher at University of Lisbon, studying combination of semantic web technologies and procedural modelling techniques, was carried out at the very start of the thesis that helped to introduce the author to the field of linked data in construction to the author. Similarly, the introduction process was sped up by two multi-day workshops held by the Dutch company Semmtech that introduced the author to relevant linked data standards as well as the tools for their industrial application. Several months into the research, in June, a week-long Linked Data in Architecture (LDAC) conference was attended in Matera, Italy. The first part, the summer school, was used to deepen the knowledge through attending lectures and participation in hackathon on knowledge validation. The second part, the workshop, was used to compare the research to the current research trends to assess its relevancy. To gather as much input as possible from the experts in the field, the research was presented at the monthly meeting of the World Wide Web Consortium Linked Building Data Community Group (W3C LBD CG) where researchers from around the world could get deeper insight into the research, ask questions and give inputs that shaped the research. Finally, based on all the knowledge gathered before, a two-week guest stay was carried out in Antwerp at Neanex premises to speed up the development process and align the research direction with the company interests.

The final output of the research is this thesis document, structured as follows. After this introduction, state of the art will be presented aiming to achieve two goals. On one hand, it aims to give a critical overview that could help to shape this research, while on the other hand it introduces most important concepts, taking into consideration that these concepts, often not well-known in the BIM community, will be referred throughout the research. Firstly, the Industry Foundation Classes data schema will be presented, following a description of Semantic Web Technologies with specific focus on the subset called Linked Data, and an even more specific application of Linked Data in Architecture and Construction. To consider the relevancy of the research, the state of the art will be concluded by an overview of research on digital building permit topic, both in industry and in the academy. After the state of the art, a theoretical framework that would allow correlation of the EN 17632 and ISO 16379 standards will be developed. In the next step, the proposed framework will be further developed in code, resulting in prototype applications that will be tested on specific use-cases. Finally, the thesis will conclude with remarks about relevant findings and future outlook.

This page is intentionally left blank

2. STATE OF THE ART

This chapter will discuss the state of the art and plans regarding concepts relevant to digital building permits, namely Industry Foundation Classes (IFC), Linked Data, and their connection. These concepts serve as a foundation for understanding how data can be structured, interconnected, and leveraged within the context of digital permitting processes and the approach that will be applied in later chapters. Finally, to consider overall relevancy of the research, a brief overview of current efforts regarding digital building permits will be discussed.

2.1. Industry Foundation Classes (IFC)

The true value of BIM lies in continuous use of a digital building model across disciplines and life-cycle phases. Such usage of BIM is commonly named in the industry as BIG BIM, in contrast with little BIM that aims to use BIM as a siloed, discipline specific solution (Borrmann et al., 2018). However, due to the complexity and diversity of use-cases a Building Information Model should go through its life-cycle, it is hardly possible to consider only one tool that could provide all the functionalities for the whole life-cycle. On the contrary, it is much more likely that the building model will need to be used in a variety of different tools aimed at different purposes. To prevent manual data entry and reduce the risk of errors, is necessary to have a comprehensive and vendor-neutral building information model as a basis for the data exchange, i.e. OPEN BIM. (Borrmann et al., 2018). However, the alignment of different stakeholders across such a fragmented industry is a very difficult and lengthy process. Process of digital product description started in the 1980s and has so far achieved significant progress (Turk, 2023a). One of the most notable standards is the Industry Foundation Classes (IFC) data model, nowadays commonly used for information exchange. Due to its importance in general as well as in the context of this paper, historic developments of the IFC will be briefly presented. Afterwards, the structure and most important concepts of object relationships will also be presented. Moreover, current issues of the IFC data model will be stated. Finally, the chapter will conclude with an overview of the roadmap for future IFC development.

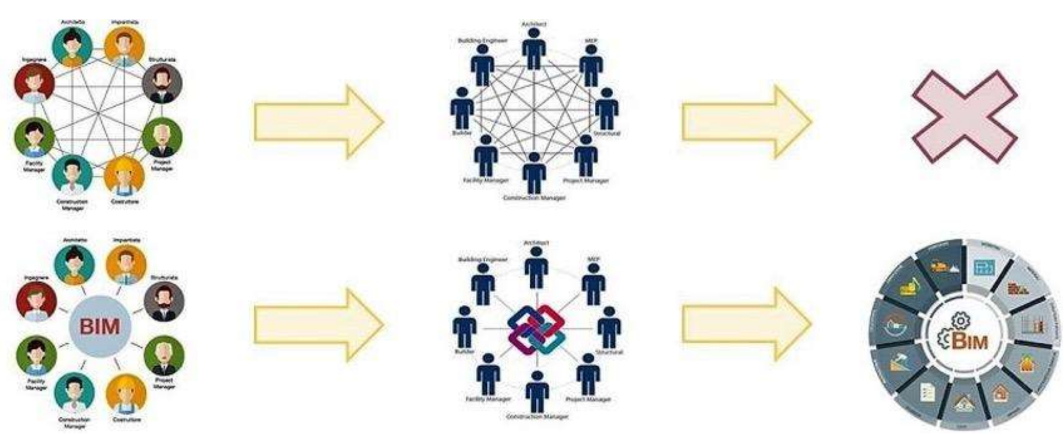

Figure 2.1 Illustration of importance of IFC in interoperability (ACCA Software, 2023)

2.1.1. History of the IFC

Methods for data exchange of product models started developing as early as 1970s in many different interest groups, e.g., US Ministry of Defense and the German Association of the Automotive Industry (Turk, 2023b). Most of these early efforts were limited to geometry exchange. In the 1980s, Standard for the Exchange of Product (STEP) was published (Borrmann et al., 2018). STEP is a comprehensive ISO standard (ISO 10303) that describes how to represent and exchange digital product information. The core idea of a neutral format is to reduce a number of interfaces by establishing a neutral format as an intermediary between proprietary formats, as shown on Fig. 2.2.

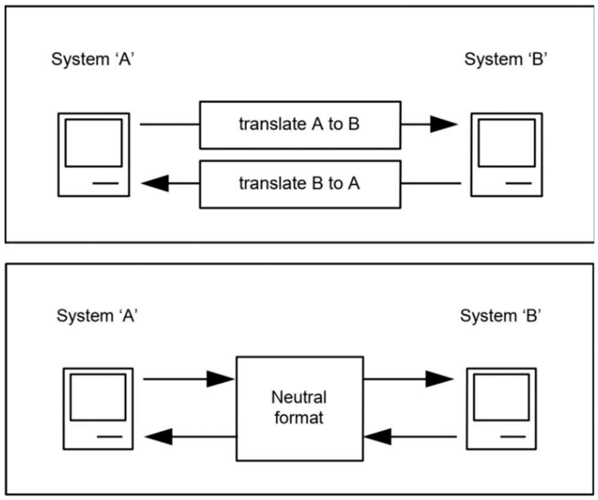

Figure 2.2 Reduction of direct interfaces by establishment of a neutral format (Turk, 2023a)

Certain successes were accomplished in standardization of building product modelling with the ISO STEP approach. However, the standardization through ISO included a slow and complicated process for reaching a consensus, so another approach was taken through the foundation of International Alliance of Interoperability (IAI) in 1995. Group of engineering offices, construction companies and software developers collaborated on several research projects (often EU funded) and went on to form IAI to speed up the development of standards (Turk, 2023b). First version of the IFC, IFC 1.0 was published in 1997 and since then many were published, with two of these (IFC 2x3 and IFC4) reaching the stamp of ISO and becoming a standard for interoperability in the industry as shown in Fig 2.3.

A notable change happened in 2005 in the IAI. Administratively, IAI changed its name to buildingSMART to put more emphasis on business benefits of integrated design and construction. More importantly, a mindset shift happened inside the consortium from achieving complete software interoperability to an approach that goes beyond purely technical aspects. In practical terms, buildingSMART changed the focus to narrower scopes, enabling developments of specifications that were easier to implement for software vendors. This approach was formalized in "The useful minimum" report in 2006 that defines the concept of the useful minimum as "The minimum scope for data exchange, which makes IFC based exchange a better solution than any other available format." (Laakso and Kiviniemi, 2012).

The IFC2x3 TC1 published in 2007 implemented these principles and became the most widely used and official version of the schema. 10 years later, the IFC4 ADD2 TC1 was published and remains the reference IFC schema, although, due to slow implementation by software vendors, it could be argued that the IFC 2x3 TC1 is still used more widely than the latest official version ("History and versions of IFC – BIM Supporters,"). The most notable IFC versions are listed in Fig. 2.3.

Figure 2.3 Most notable IFC schema versions through time (adapted from (buildingSMART, 2023a))

Currently, the IFC 4.3 TC1 schema version is waiting for the approval to be standardized as an ISO standard. This extension of the IFC 4 schema aims to improve infrastructure object representation, such as railways. IFC 4.4.0 is currently in development and will extend the IFC 4.3 schema, adding additional features, mainly for tunnels ("The status of IFC 4.3 and the benefit of further extensions as IFC 4.4 - buildingSMART International," 2022).

2.1.2. Technical overview of the IFC

The brief version of technical overview of the IFC that follows aims to present the parts of the schema that are the most relevant for the research that will be carried out. A detailed, comprehensive technical overview of the IFC schema can be found in *Building Information Modeling: Technology Foundations and Industry Practice* (Borrmann et al., 2018).

2.1.2.1. EXPRESS data model

The IFC did not go through the ISO standardization process first, but the technology used in the ISO STEP development formed the foundation of the IFC, both through the definition of a data model (EXPRESS, ISO STEP Standard part 11), as well as the way to describe specific data instances of that data model (STEP Physical File, ISO Step Standard part 21) (Turk, 2023c).

EXPRESS is a declarative language to define object-oriented data models. These data models can be modelled in textual or in graphical form, as an EXPRESS-G diagram, displayed in Fig. 2.4.

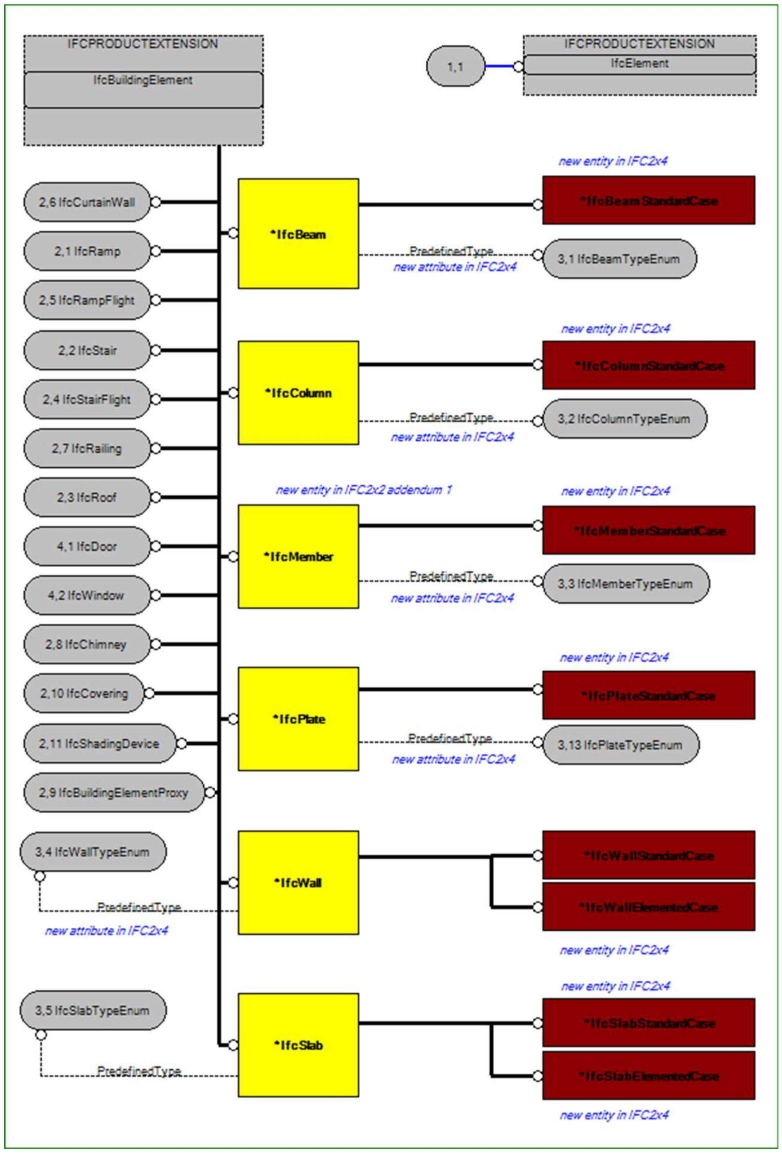

Figure 2.4 EXPRESS-G diagram depicting a part of the IFC Schema (buildingSMART, 2023b)

EXPRESS follows object-oriented principles such as abstraction and inheritance. Abstraction enables sorting objects into classes that have attributes and relationships with other classes, while inheritance simplifies modelling by implying that a subclass should have the same attributes and relationships as the class it was derived from, alongside additional class-specific attributes and relationships Inheritance is visible in the fact that the IfcRoot is the most abstract class of and is a supertype of all IFC entities. IfcRoot has 4 attributes, GlobalId, OwnerHistory, Name and Description. Through inheritance every IFC entity will have these 4 attributes as well, so it is not necessary to explicitly state them anywhere else except the IfcRoot attributes.

2.1.2.2. IFC file formats

A common misconception is that the IFC is a file format. That idea is also helped by the fact that the most common serialization of the IFC is the IFC-STEP with the .ifc extension. In reality, the IFC data model (modelled in EXPRESS) can be serialized to a certain extent in several formats, as shown in table 3.1, derived from buildingSMART official documentation.

Table 2.1 Different IFC serializations (buildingSMART, 2023c)

Format	Extension	Summary
STEP Physical File (STEP)	.ifc	Most widely used format in practice. Based on the ISO 10303-21 standard
Extensible Markup Language (XML)	.ifcXML	Enhanced readability, broad range of software tools. Based on the ISO 10303-28 standard
ZIP	.ifcZIP	It is possible to embed STEP or XML serialization in a zip file
Terse RDF Triple Language (Turtle)	.ttl based on ifcOWL	Useful for linked data applications. Turtle is the most common way of RDF serialization
Resource Description Framework (RDF/XML)	.rdf based on ifcOWL	Useful for linked data applications. Another way of RDF serialization
JavaScript Object Notation (JSON)	.json	Enhanced readability and a broad range of tools. Still a candidate
Hierarchical Data Format (HDF)	.hdf	Storing IFC data in a provisional database, providing high performance access. Based on ISO 11030-26. Still a candidate
SQLite	.sqlite	Storing IFC data within a relational database. In experimental stage

Despite many available serializations, the STEP serialization is still the most common way to express the IFC schema. EXPRESS data modelling and STEP serialization of the IFC are closely interconnected and specifics of the EXPRESS model are not easily translatable into other serialization formats, such as OWL. An additional problem is the existence of several IFC specific things, such as IfcString, IfcURIReference etc. There is no real need to copy such concepts in any other serialization besides the IFC-STEP (Van Berlo, 2022).

2.1.2.3. Organization in layers

IFC aims to cover all the data exchange happening in the building lifecycle, therefore its data model is very extensive and complex. To illustrate, the data model currently consists of 885 entities (classes), 462 property sets, 96 quantity sets and thousands of properties in standardized property sets, among others (Van Berlo, 2022). To make maintenance of such a model simpler, it is structured in 4 distinct layers. The "ladder principle" is respected, meaning that the elements from the upper layers can reference the elements from the lower layers, while the opposite is not allowed.

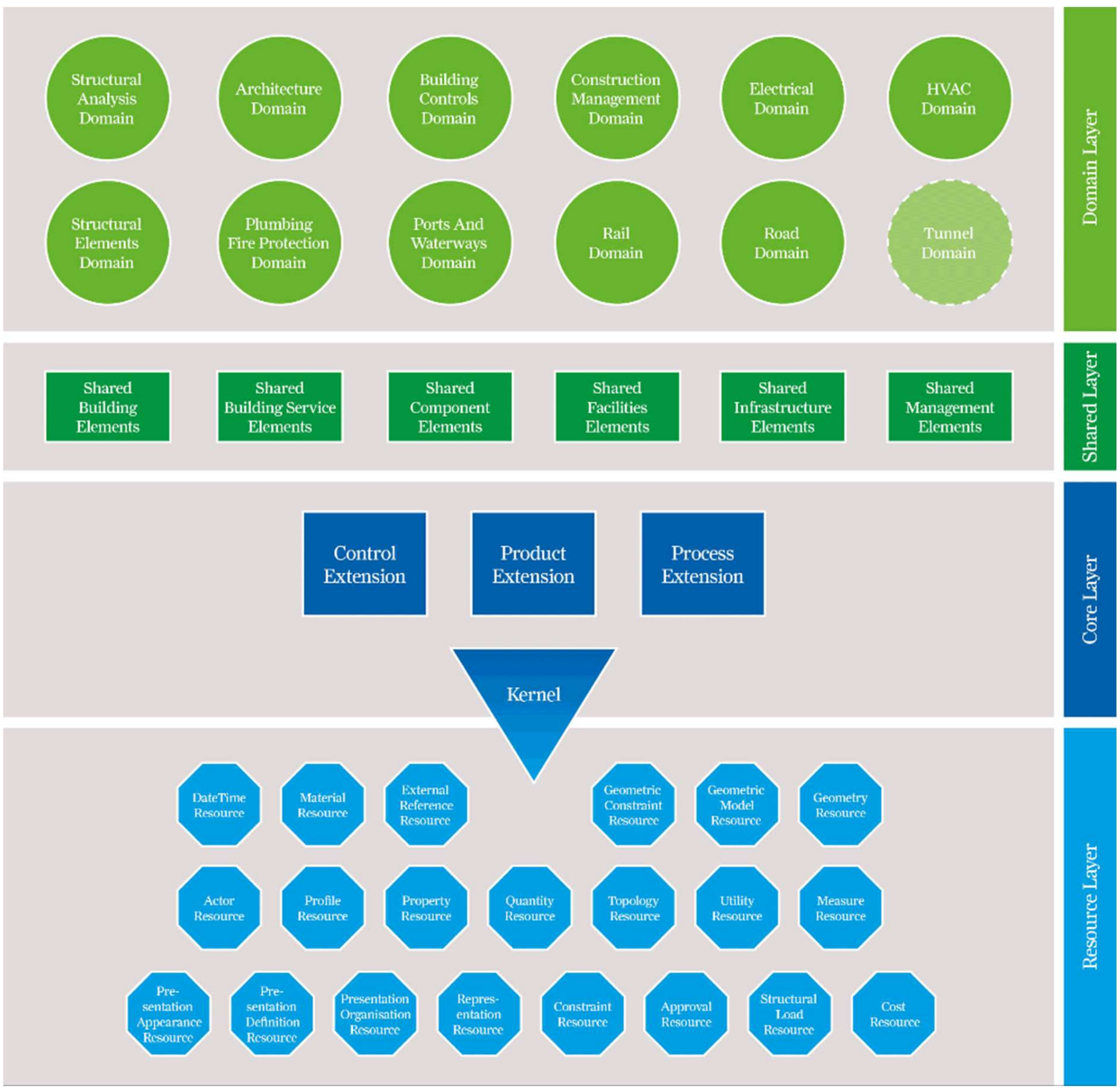

Figure 2.5 Layers of the IFC Schema (adapted by (Davies, 2022) from (buildingSMART, 2023a)

Resource layer: Resource layer is the lowest level of the IFC Schema. It is a collection of generic entities, explaining concepts such as materials, geometry, quantities, and many others. The entities from the resource layer cannot be standalone entities and never have taxonomical relationships with classes from other layers. They exist only if called by another entity from the upper layers via specific relationships, e.g. IfcGeometryResource contains entities that define geometric relations. A product, defined in as an IfcProduct (from the Core layer explained below) can be characterized by a placement, through the ObjectPlacement attribute pointing to the IfcPlacement class from the resource layer (Borgo et al., 2015).

Core layer: Core layer is where the elementary IFC classes can be found. These define basic concepts which are later referenced and re-used by the layers above through the inheritance principle characteristic for object-oriented models. There are four distinct parts of the core layer (Borrmann et al., 2018):

- IfcKernel: Contains main abstract classes such as IfcRoot, IfcObject, IfcProduct, IfcRelationship
- IfcCoreExtension: Specializes the generic concepts from IfcKernel with AEC/FM specific concepts. Further divided into IfcProductExtension, IfcProcessExtension, IfcControlExtension.
    - IfcProductExtension: Physical and spatial objects, such as IfcElement, IfcElementAssembly, IfcBuildingElement
    - IfcProcessExtension: Classes for describing processes and operations such as IfcEvent, IfcTask
    - IfcControlExtension: Classes for control objects such as IfcControl and IfcPerformanceHistory

Interoperability layer: Interoperability layer is one level higher than the Core, and it contains classes that are actual building elements that are used by a range of disciplines, across different domains such as IfcColumn, IfcWall.

Domain layer: The highest level, contains domain specific clases that are typically used for intra-domain exchange. These cannot be referenced by another layer or another domain-specific schema.

2.1.2.4. Relationships

Expressing relationships between objects is a powerful function and one of the key qualities of current, EXPRESS based IFC data model. Such ability looks at individual building elements from the point of view of the whole model, in interaction with other objects. The IFC data model follows the objectified relationship principle. That means that the relationships are not formed directly, but through a special intermediary object that represents the relationship. This element is always a subclass of the IfcRelationship class, there are six relationship types that serve specific basic functions of the IFC data model (buildingSMART, 2023):

- IfcRelAssociates - relation to external source of information, such as classification, document, contract

- IfcRelDecomposes – represents concepts of composed objects through a whole/part hierarchy, enabling navigation from the whole to the parts and vice versa

- IfcRelDefines – a generic, abstract relationship, depending on the subtype it assigns an object type to its occurrence, assigns a property set to an object instance, or assigns a property set template to a property set

- IfcRelConnects – a connectivity relationship that connects objects under some criteria

- IfcRelDeclares – handles the declaration of objects or properties to a project of project library

- IfcRelAssigns – a generalization of the link relationship between IfcObject and its subtypes

As mentioned, the IFC Schema allows expression of whole/part relationship via IfcRelDecomposes. There are special types if the more generic IfcRelDecomposes relationship, depending on the dependency they are expressing. The one that is most relevant for this research is the aggregation relationship IfcRelAggregates which can be applied to all subtypes of IfcObjectDefinition (in other words, any kind of a semantically treated object or process) Decomposition implies that the whole depends on the definition of the parts and parts depend on the existence of the whole. In example, roof can be an aggregation of slabs, rafters etc., and a beam. The same applies to objects containing spatial semantics, such as IfcProject, IfcBuilding, IfcBuildingStorey, IfcSpace and IfcSpatialZone.

Additionally, depending on the criterium, objects can be connected by special types of IfcRelConnects connectivity relationship. Relationships such as port connections (IfcRelConnectsPortToElement), interference, (IfcRellIntereferesElement), fill (IfcRelFillsElement) as well as several others exist. The one of particular significance for this research is IfcRelContainedInSpatialStructure that can be used to express relationships between non-spatial and spatial objects. For example, a sanitary terminal can be contained in the spatial structure of a kitchen (which is an instance of an IfcSpace). In case of an element not being contained in a specific space (in example, a pipe that is in the floor or a wall) it can be contained a different entity, such as IfcBuildingStorey.

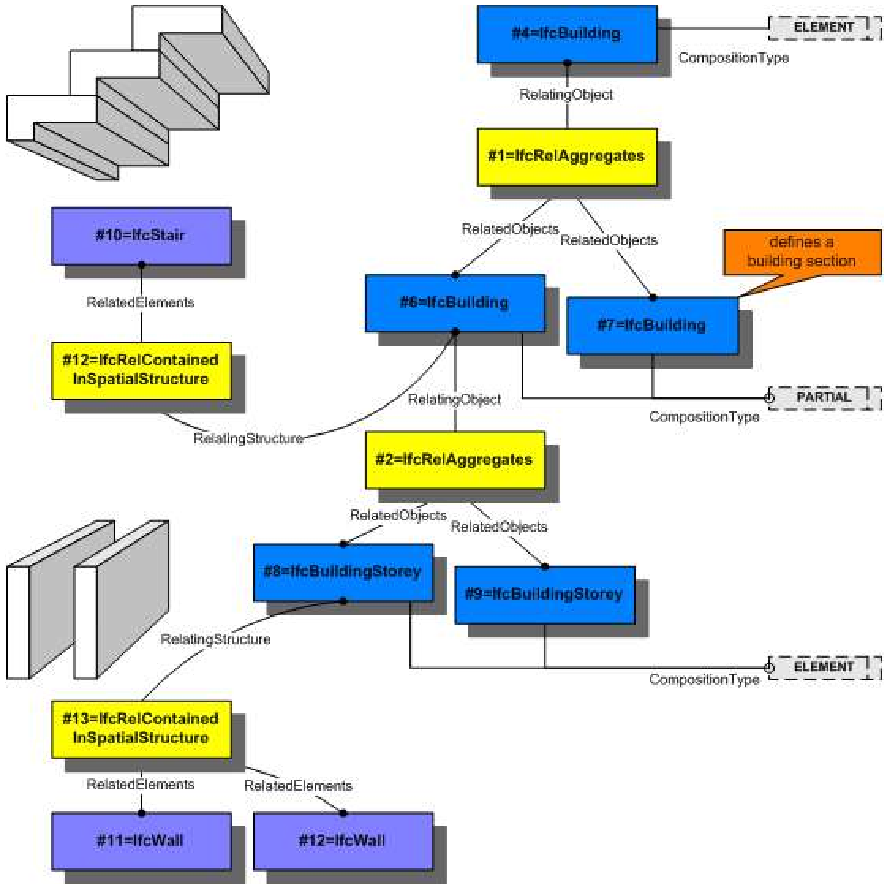

Figure 2.6 An example of aggregation and spatial decomposition relationships (buildingSMART, 2023)

2.1.2.5. Geometric representation

There is a strict division between the semantics and the geometry in the IFC data model. All objects are defined semantically and can later be linked to one or more geometric representations. The identity remains with the semantic object, not with the geometric representation. This enables many different geometric representations of an object. Depending on the use-case, these representations can be simple (e.g. triangulated geometry for visualization programs) or more complex (e.g. Brep or CSG geometry for design authoring tools). The geometry representations can also be two dimensional (e.g. it is possible to store a drawing within a semantic object).

2.1.2.6. Extension Mechanisms

Universal, key characteristics of building elements can be defined directly in the IFC data model. However, there are many specific characteristics depending on various parameters such as location, codes etc., that define all the necessary characteristics that a specific object instance should have. Adding these directly to the IFC schema would make it even more bloated and hard to work with. These additional characteristics (classifications, ontologies) can be defined in the IFC by the user if needed, through the mechanism of addition of the IfcProperty instances. These properties are grouped into IfcPropertySets and assigned to specific objects. Such flexible approach enable use of the IFC in specific projects and domains even if the original IFC data model does not contain such characteristics beforehand. However, this can lead to a large number of user defined properties, thus hindering interoperability between different users. To confront this issue while keeping IFC data model to the minimum size, standardization of these external properties is needed. IFC is not made to be the only standard to be used in the building design, but rather a standard that would be the connection between different standards (van Berlo, 2022). To facilitate this, the definitions of properties for different languages and locations are available in the buildingSMART Data Dictionary (bSDD) ("buildingSMART Data Dictionary," 2023).The bSDD can be referenced directly from the core IFC model.

2.1.2.7. Naming convention

Finally, to distinguish between different concepts in the IFC schema as simply as possible, a naming convention was established as follows:

- Data item names for types, entities, rules, and functions start with the prefix Ifc and follow the CamelCase convention without underscores (e.g. any IfcBuildingElement can have in IfcGeometricRepresentation item consisting of IfcSurfaces, IfcSolidModels or IfcCurves)
- Attribute names also follow the CamelCase convention but without the prefix (in example, the entity IfcDoor has an attribute OverallWidth)
- Property set definitions that stat with the prefix "Pset_" and continue in English with CamelCase naming convention.
- Quantity set definitions start with the prefix "Qto_" and continue in English with CamelCase naming convention
- Enumerations are written in English with titlecase naming convention

2.1.3. Current issues of the IFC data schema

IFC is an open and internationally standardized data model whose maturity is proven in practical use as a main data exchange facilitator in the industry. The benefits of using an open data model are many, some being so important that certain countries, such as Singapore, the Netherlands and Finland prescribe the IFC deliverables as a requirement of public building projects (Borrmann et al., 2018). However, there are several issues that still hinder developments of the Architecture, Engineering and Construction (AEC) industry towards a higher degree of interoperability.

Firstly, there are several issues not directly related to the IFC specifically, but with the EXPRESS modelling language and its serialization in the most common IFC file format, IFC-STEP. Despite several possible serializations (shown in chapter 3.2.2), in practice a lot of implementation particularities of the IFC are based on STEP serialization, and serializing to the other formats besides STEP necessarily results in an bespoke serialization of the data model that cannot be used with generic libraries (*Technical Roadmap*, 2020). Moreover, EXPRESS is not based on mathematically rigid theory like OWL, making it impossible to use some existing algorithms and technologies. The popularity of EXPRESS and STEP is low outside of the few specific industries, which makes the use of ontologies from other domains impossible.

The complexity of the IFC data model is another issue. On one hand, it enables description of a building in various ways in different stages of its lifecycle. On the other, such complexity brings some disadvantages along. The sheer size of the data model is so big that it is very hard to develop a comprehensive software implementation of it. Partial implementation is possible, and even encouraged by buildingSMART through the concept of Model View Definition (MVD), that is defined as a subset of the IFC needed to satisfy Exchange Requirements (ER) needed for a certain workflow (buildingSMART, 2023).

2.1.4. Planned future developments

The developments of the IFC are thoroughly discussed in the buildingSMART Technical Roadmap. The Roadmap presents the overview of planned work of buildingSMART for several notable initiatives (BCF, bSDD, IDS) alongside the IFC, as shown in Fig. 2.7. The general buildingSMART goal in the future is to move from bespoke towards generic solutions that are more easily used by actors such as modellers, developers, and implementers.

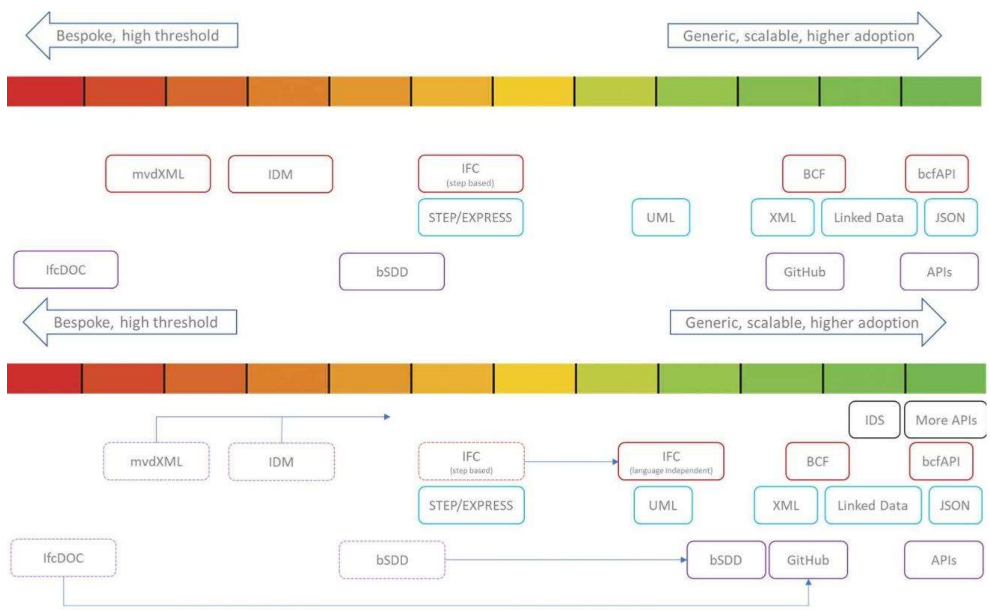

Figure 2.7 Comparison between current (top) and planned (bottom) developments of buildingSMART standards and solutions (*Technical Roadmap*, 2020)

The IFC is currently optimized for file-based exchange through the application of advanced EXPRESS modelling solutions. As a result, the current structure is hard to represent in formats other than STEP. The IFC is very complex, but also ambiguous in software implementation. That can cause different implementation of same concepts that are not interoperable, and the implementation itself is often very complex and difficult (Rasmussen et al., 2020).

To become optimized for transactional exchanges and enable use-cases such as automated microservices, working with connected Common Data Environments (CDEs) and real-time representation of data streams, the IFC needs to move from EXPRESS and STEP based structure towards a language independent data model that would enable more efficient serializations in different formats, such as JSON, RDF, HDF5, as shown in Fig 2.8. The left part of the figure contains STEP centric IFC concepts such as IfcString, IfcURIReference etc. is shown in blue, and the right part shows planned, universal IFC base structure.

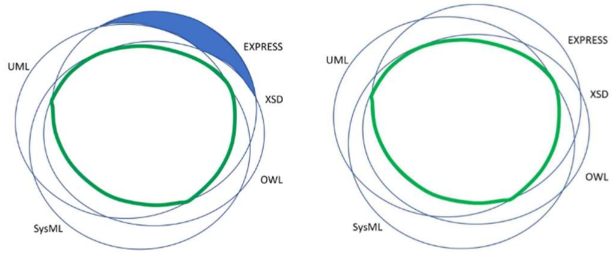

Figure 2.8 Comparison of current (left) and planned (right) IFC base structure (Van Berlo, 2022)

The complexity of the IFC causes additional implementation issues. The current IFC implementation revolves around the mentioned MVD concept. These define the subset of the schema to be implemented, restrictions and the conformance level expected of the implementations. Every MVD needs to be individually implemented in the tool, and an agreement is always necessary before the new version of IFC can be released. To solve this, the monolithic schema needs to transform to a modular one. These modules can act individually and have separate release cycles, maintenance, and responsibilities. To make them a part of the whole, an interoperability layer can serve as a base for the extensions. The advantages of this modularization would include shorter release cycles, faster support of the IFC in software, and stronger interoperability between extensions.

2.1.5. Connecting IFC to other data schemata

Significant effort has already been made in connecting BIM (namely IFC) models and other data schemata with the goal of improving interoperability between different kinds of built environment representations.

BIM and GIS integration is one of the core components of urban digital twins that would enable interconnectivity between larger (GIS) and smaller scale (BIM) representations of the built environment. There are several different approaches towards establishing the connection between these domains, such as unidirectional transformations from IFC to CityGML (Donkers et al., 2016) or in the opposite direction (Salheb, et al., 2020), or even integrated processing of data by adding an intermediary data model such as QL4BIM system (Daum et al., 2017). A bibliometric analysis (Shkundalov and Vilutienė, 2021) of BIM, GIS and Web environment integration has shown growth in the last decade, with the main investigation field being BIM and GIS interoperability challenges. These challenges include a lack of technologies and methods for BIM and GIS integration which occur due to the fact that most BIM authoring tools use proprietary formats that cannot be processed outside their native environment. An additional problem, the lack of unified standards is considered one of the main obstacles that need to be investigated.

Building energy modelling (BEM) plays an important role in design and assessment of relevant building parts, such as materials, HVAC systems, operations schedules among others, meaning that the interoperability between BIM and BEM, i.e. their most common open-source formats, IFC and gbXML, should be as seamless as possible. This complex task, dealing with both semantics and the geometry of the building is often tackled with a direct, one-to-one conversion between the two data models. A semantic gap caused by insufficient simulation parameters exists in the IFC-to-gbXML conversion but could be bridged by energy modelling presets. The geometric gap has been a long-term source of discussions. Often BIM tools are not suited for BEM, and BIM geometry export results in missing components and modellers therefore often need to turn back to manual energy modelling from scratch (Yang et al., 2022).

An additional challenge is connection of BIM (IFC) and Internet of Things (IoT) technologies that would enable live representation of a smart building. While IFC defines several hundred entities, the IFC4 has only a basic set of IoT-related elements that are not sufficient to represent IoT networks and the interconnection between different nodes such as servers, gateways, or the authorisation that a user needs to access it. (Ruiz-Zafra et al., 2022). Several methods to connect BIM and IoT data exist, such as the use of existing application programming interfaces (APIs), transforming the IFC in a relational database and creating a new query language (Tang et al., 2019).

Additionally, digital building permitting (DBP) process should contain data from these data schemata in a certain extent, as well as additional information from the actual permitting process that very often varies depending on the country and municipality specifics. Additionally, it should be possible to use the model from the DBP process in latter stages, such as construction and operations phase.

Taking this into account, a solution needs to be found that would allow flexibility and adaptability on an individual, project basis. The solution chosen for the remainder of this research is a set of technologies named Semantic Web Technologies (SWT), well known for their ability to connect heterogeneous data through a semi-structured and flexible data model

2.2. Semantic Web & Linked Data

The lifecycle of a built asset involves numerous interactions between various actors, which necessarily create an issue of information exchange created in different formats by different tools. This information is often exchanged by specialized data exchange models such as IFC. However, new, and often complex use cases are emerging. To cover all use cases, these specialized data models would need to be extremely complex, and many tools would need to have bespoke interfacing solutions depending on the use case. Due to the complexity, it would be hard to develop one standard that would be equally powerful in representation of different aspects of the built environment. In example, the most mature data model, the IFC, is powerful in geometry representation and product properties. However, its weakness is dynamic data and GIS representation, among others. (Pauwels et al., 2022). Therefore, the emerging paradigm is not to create one standard that would encapsulate all possible use cases, but rather to find ways of making different data sources interconnected without direct translation from one to the other. The Linked Data concept (along with the interrelated knowledge graphs and Semantic Web concepts) has had notable success in merging heterogeneous data sources, both overall as well as in the AEC industry specifically. Since the utilization of the Linked Data concept is an important part of the research developments in this paper, the most important underlying concepts and technology will be presented in this chapter.

2.2.1. Semantic Web Technologies (SWT)

The efforts to make computers connected to each other started in the 1950s, firstly through military and later in academic efforts in the US. Wider adoption, however, started in the 1990s with the introduction of the internet. Internet is a network of interconnected devices that comply to common standards. Any device that complies to these standards can access the internet (Turk, 2023a). The World Wide Web, commonly referred to as the web can be thought of as an additional layer on top of internet, created as an attempt to organize the ways data is structured to enhance interoperability. The development of the Web started in 1989 by Sir Tim Berners-Lee and colleagues at CERN, and has so far had three notable versions (Britannica, 2023).

- Web 1.0 – The first iteration of the Web put users in a passive spot. The content was produced by few people for many consumers and a typical user did not have any means of producing internet content.
- Web 2.0 –Web, also called “web of documents” is the one that is the most common today. It emphasizes the user role in content production. A good example of the Web 2.0 paradigm are website such as Wikipedia, Youtube and blogs. The shortcoming of the Web 2.0 is that, while producing content, most of this content is available only as document in human-readable and human-understandable form.
- Web 3.0 – Commonly referred to as the “web of data” or Semantic Web, develops standards to make the data on the internet machine-readable. This data could then be processed by the machines, facilitating use-cases such as harmonization of heterogeneous data, customized user-experience, and querying.

The Semantic Web promises organization of the available information by defining meaning of certain objects (people, buildings, building components) and the relationships between these objects. To achieve this, the Semantic web stack consists of several building blocks, as shown in Figure 2.9. It is also visible that Linked Data Concept is a subset of the SWT using some of the SWT technologies that deal specifically with publishing data. The organization responsible for standardization is the World Wide Web Consortium (W3C). For the sake of understanding the rest of the research even for the non-technical audience, a brief, non-comprehensive overview of the standards applied in this research will be presented.

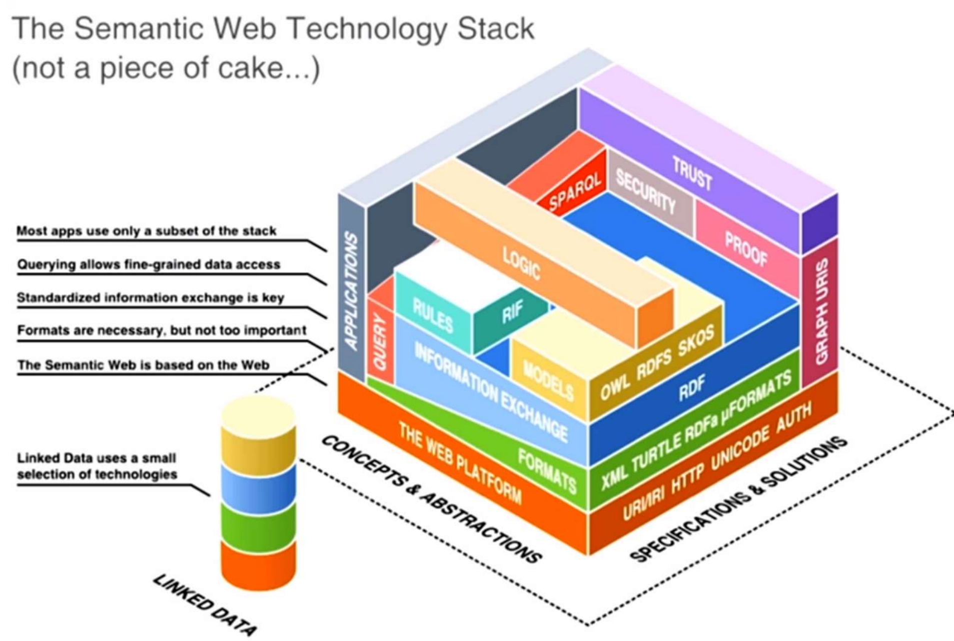

Figure 2.9 Semantic Web Technology Stack (Sack, 2016)

2.2.1.1. Data Modelling

A big challenge of interoperability in general is achievement of structural interoperability, making sure that the data is exchanged through common way of information representation (Turk, 2023a). To achieve this, several ways of data modelling, such as Direct-Edge-Labelled (DEL) graphs, heterogeneous graphs and property graphs exist. Application of each of these models brings some benefits as well as certain difficulties, and the choice should be made depending on the application (Hogan et al., 2020). A DEL based data model, called the Resource Description Framework (RDF) specifies that all things that are described can be called resources. A relationship between two resources is called a triple. RDF is recommended by the W3C and has seen application in many fields, such as social constructs, pharmacy, and the one of specific interest to this research, the AEC industry.

The RDF is designed to have a simple data model, easy to process and manipulate. Additionally, the formal semantics enable inference. The simplicity of the data model is ensured by structuring the expressions in triples. Each triple consists of three parts. A subject, a predicate, and an object. A simple knowledge of representation using triplets is shown in Fig. 2.10.

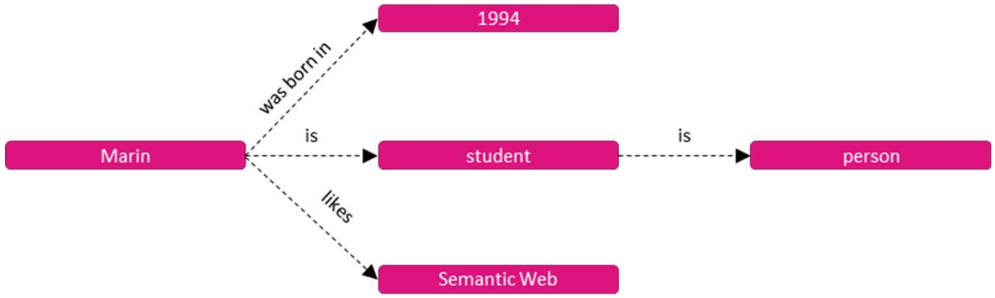

Figure 2.10 Knowledge representation using triplets

This knowledge graph consists of 4 triples, and contains following statements:

1. Marin is a student.
2. Marin likes Semantic Web
3. Marin was born in 1994
4. Student is a person

An additional statement is available through inference. If Marin is a student, and students are persons, therefore Marin is a person.

This data model is very simple and could be useful for internal use. However, it is still very ambiguous. Concepts such as Marin, Semantic web, or even student or a person are not interpretable by a computer and could lack meaning outside the one given by a specific person. To resolve this issue, an unambiguous way of representing individual nodes is by attaching Universal Resource Identifiers (URI) to them. These ensure that each concept is uniquely identified and there is no uncertainty when that object is referenced by another one. These URI can also resolve to Universal Resource Locations (URLs) if they have an established place on the internet to which they should point to. An example of the knowledge graph from Fig 2.10 updated with the use of URIs is shown in Fig. 2.11.

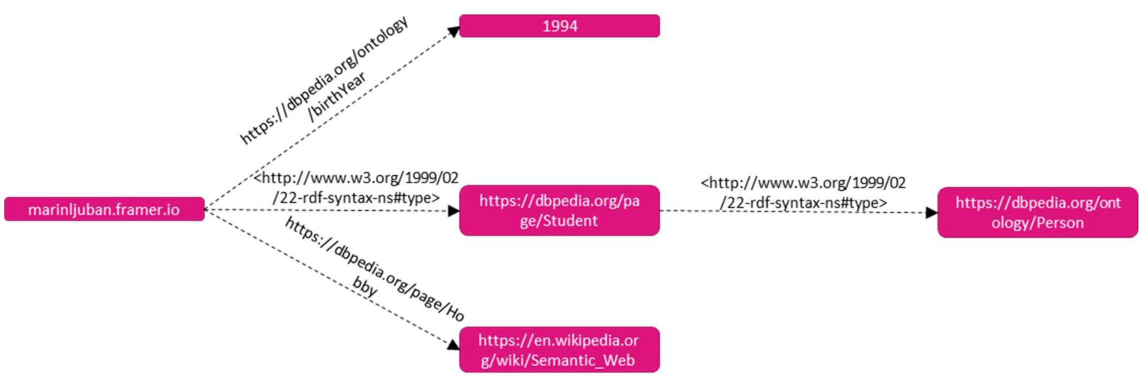

Figure 2.11 Knowledge representation using triplets and URIs

Additionally, besides the graphical way of representing RDF statements, these can be represented in other formats. To achieve syntactic interoperability, RDF can be serialized in various formats, such as

RDF/XML, N-triples etc. An N-triples representation of the knowledge from Fig. 2.11 is visible in Fig. 2.12.

```
<marinljuban.framer.io> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://dbpedia.org/page/Student>

<https://dbpedia.org/page/Student> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://dbpedia.org/ontology/Person>

<marinljuban.framer.io> <https://dbpedia.org/ontology/birthYear>
<"1994">
```

Figure 2.12 Knowledge expressed in N-triples serialization of RDF

The Fig. 2.11. and Fig. 2.12 show that a node can be specific entity identified by an URI, but it can also be a literal value, such as the integer value of 1994. Literals can only be found as an object of a triplets, while the subject and predicate are necessarily URIs.

Both figures show that introduction of URIs to data modelling can make the expressions very cumbersome. To ensure syntactic interoperability in a way that is more human-readable the statements are commonly written in the Terse Triple Language (Turtle) syntax. It is allowed to use both shortened URIs (with the namespace) as well as full URIs interchangeably, using the same graph.

SPARQL PROTOCOL AND RDF QUERY LANGUAGE (SPARQL)

```
  PREFIX dbp: <http://www.dbpedia.org/page/>
  PREFIX dbo: < http://www.dbpedia.org/ontology>
  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

 <marin.Ljuban.framer.io> rdf:type dbp:Student;
                          dbo:birthyear 1994.

dbo:Student rdf:type dbo:Person .
```

Figure 2.13 Knowledge expressed in Turtle serialization of RDF

2.2.1.2. Semantics

Besides achieving semantic interoperability, i.e. presenting data in machine readable form, it is necessary to ensure that the data is machine-understandable. Therefore, a shared terminological framework needs to be established to ensure that the concepts are understood by all actors in an unambiguous way. That is commonly done through definition of ontologies. An ontology is defined as "a formal, explicit specification of a shared conceptualization", with the meaning as follows (Studer et al., 1998):

- Formal – the ontology should be machine-readable
- Explicit – the concept types and the constraints on their use are explicitly defined
- Conceptualization – an abstract model of some world phenomenon is created by identification of relevant concepts of this phenomenon

- Shared – The knowledge captured by an ontology is agreed upon by a group of individuals

Depending on the interpretation, vendor-neutral schemata such as IFC can be thought of as ontologies. As mentioned, IFC is defined in a less formal language (EXPRESS) and follows two assumptions of formal logic application that are not aligned with the web ontology principles (Bonduel, 2021).

The first principle is called Closed World Assumption (CWA). The CWA means that a certain statement needs to be known as true to actually be true. If a statement does not exist, it is considered as non-existent or false. On the contrary, Open World Assumption (OWA) is often used in development of ontologies for Web Environment. The OWA assumes that a lack of a certain statement does not necessarily that it does not exist, it just cannot be found in the current data. Since the internet will always remain incomplete, such a statement can possibly be found in another place on the internet.

The second principle is the Unique Name Assumption (UNA). This principle assumes that one name means one concept. Usually, a central organization should be in charge of connecting concepts with terminology, such is the case of buildingSMART and the IFC. On the contrary, a principle often used in Web Environment is the No Unique Name Assumption (NUNA). NUNA assumes that there is a possibility that the same concepts have different terminology in different contexts, so it is not necessary to have a central organization regulating the terminology. Instead, a harmonization between ontologies can be done (e.g. BRICKschema and RealEstateCore harmonization efforts ((Wallin and Fierro, 2022)).

To express semantics, a common language is needed. In example, it could be useful to clusters of resources under a common term, i.e., all humans could be defined as belonging to a class called human. This way, all the properties that define the class human will also be inherited by the subclasses (such as male, female, student etc.). Additionally, characteristics should be defined. In example, the birthYear property should be explicitly defined as belonging to the human class, while its value should be an integer. To model these facts, the RDF Schema (RDFS) was created ("RDF 1.2 Schema,"). It provides data modelling vocabulary for RDF data, intended for those users that primarily need a classification hierarchy (taxonomy) with typing of properties. (Antoniou et al., 2005).

When the vocabulary of RDFS is not enough, it can be extended by additional vocabularies. One commonly used vocabulary is the Simple Knowledge Organization System (SKOS ("SKOS Simple Knowledge Organization System Reference,")). SKOS defines classes and properties to represent concepts by giving them definitions labels (e.g. prefLabel, altLabel etc.), semantic relations (e.g. broader, narrower, related) and mapping properties (broadMatch, narrowMatch, relatedMatch). Another commonly used vocabulary and the W3C recommendation is the Web Ontology Language (OWL ("OWL 2 Web Ontology Language Document Overview (Second Edition),".)). OWL provides more expressivity and allows more complex statements than RDFS. To see these vocabularies in practice, a simple observation of three concepts, person, teacher, and student will be made. SKOS makes it possible to define each concept by giving them definitions and labels in several languages (in example, prefLabel for person is "Human"@en, but can also have altLabel such as "Persoon"@dt or "Osoba"@hr for Dutch or Croatian labels). RDFS makes it possible to state that both teacher and student are a subclass of the human class, and OWL makes it possible to express that the class student is disjoint with the class teacher, meaning that a human cannot be a student and a teacher at the same time (if such a notion needs to be defined). Additionally, OWL can express statements such as that a human can have no more and no less than one birthyear (while the exact definition and labels of birthYear property are defined with SKOS). SKOS, RDFS and OWL respect the OWA and NUNA.

When discussing ontologies, an informal but important distinction is made between the terminology and application. Universal terminological vocabulary is stated in so called TBox (terminological), while assertional statements regarding specific individuals are called ABox statements. Due to the fact that ABox statements are specific, they cannot be considered shared, which is one of the key traits of an ontology. Therefore, only TBox statements are often considered as ontologies (Bonduel, 2021). An example ABox and TBox statements is shown in the Figure 2.14.

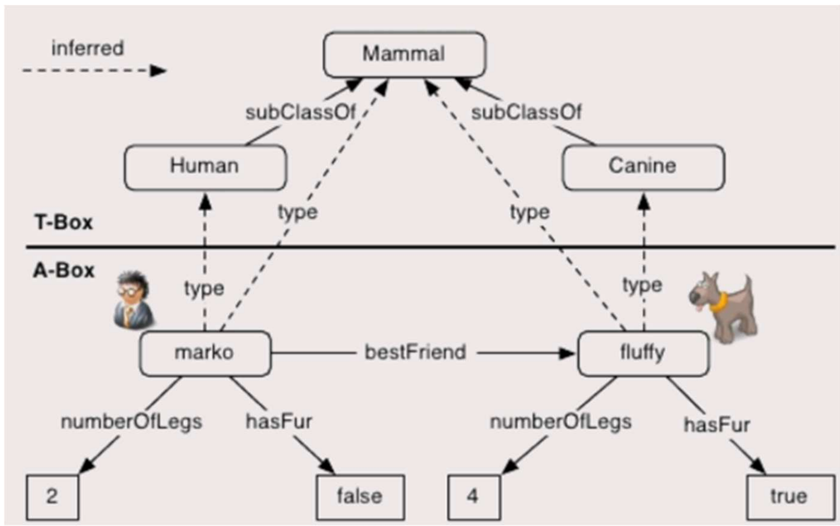

Figure 2.14 Example of TBox and ABox (Pauwels, 2023)

2.2.1.3. Querying

The W3C recommendation for extracting data encoded in RDF is SPARQL Protocol and RDF Query Language (SPARQL). Basic constructs of SPARQL are triple patterns. However, SPARQL allows variables as terms. These variables are marked with question marks and returned after the data graph evaluation. (Hogan et al., 2020). A typical query is broken down into three parts, with the fourth optional part.

- Definition of prefixes – Similarly to the Turtle expression in Figure 2.13, each SPARQL query has definition of prefixes in the beginning
- Commands – Commands specify what should be done with the variables in the scope of the query
    - SELECT – Returns selected variables in a set
    - CONSTRUCT – Returns variables as an RDF dataset
    - INSERT – inserts the data in an RDF dataset
- Graph path – Specifies the way that the variables should be found, identified by the WHERE keyword
- Optional parts help to refine the query with keywords such as:
    - DISTINCT – Return only one instance of a resource even if it appears multiple times

- FILTER – Filter by a certain rule (e.g. string equality)
- LIMIT – Show only a certain number of matches, even if the actual number is bigger
- OFFSET – Start showing the matches from a certain index

The simplest SPARQL query shown on Fig. 2.15. should work regardless of the dataset queried, returning a set of all the subjects, predicates, and objects regardless of the edge type.

SPARQL

```
SELECT ?s ?p ?o
WHERE {
?s ?p ?o }
```

Figure 2.15 Simplest SPARQL Query

Considering the small dataset shown on Fig. 2.13, an example query could look as shown in Fig. 2.16.

SPARQL PROTOCOL AND RDF QUERY LANGUAGE (SPARQL)

```
PREFIX dbp: <http://www.dbpedia.org/page/>
PREFIX dbo: < http://www.dbpedia.org/ontology>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?person ?year
WHERE {
?person rdf:type dbp:Student .
?person dbo:birthyear ?year
}
```

Figure 2.16 Example SPARQL Query

The resource found as the ?person variable would be the URI (<marinljuban.framer.io>), while a literal would be returned for the ?year variable (“1994”).

2.2.1.4. Validation

RDFS and OWL enable inference and help in managing real-world complexities with the NUNA and OWA assumptions. However, some datasets should be evaluated as the only source of truth for a certain workflow. In example, a BIM model submitted for the digital permitting process should contain all the information that is needed. Some success was made with inference-based validation (RDFS and OWL) and query-based validation (SPARQL). SPARQL queries can handle most validation needs and SPARQL is implemented in most RDF products. However, writing SPARQL queries for validation can be very complex and verbose, making it difficult and hardly scalable (Gayo et al., 2018). The fairly novel (2014 onwards) but acclaimed approach are the shape languages. There are two notable representatives of the shape language application for RDF validation, Shape Expressions (ShEx) and Shapes Constraint Language (SHACL). Since SHACL is a W3C Recommendation (de-facto standard) and is implemented in more tools, it will be used in the remainder of the research.

SHACL is divided into two parts. SHACL Core describes a core RDF vocabulary to define shapes and constraints, while SHACL-SPARQL describes extension mechanisms for SPARQL. (Gayo et al., 2018). A simple SHACL shape shown in Fig. 2.17. It puts two different constraints on the :Student class. The first one regarding the node itself (NodeShape) states that the node should be an IRI type. The second (PropertyShape) specifies that a maximum of one birthYear property of the student class should exist and be of integer datatype. Besides the whole class (sh:targetClass), the targeting can be done on the instance level (sh:targetNode).

TERSE TRIPLE LANGUAGE (TTL)

```
PREFIX dbp: <http://www.dbpedia.org/page/>
PREFIX dbo: < http://www.dbpedia.org/ontology>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sh: <http://www.w3.org/TR/shacl>
PREFIX new: <http://www.marin.com/mynewnamespace#>


new:StudentValidation a sh:NodeShape ;
:sh:targetClass dbp:Student ;
sh:property new:birthyearShape .

new:birthyearShape a sh:propertyShape ;
sh:path dbo:birthyear ;
sh:maxCount 1 ;
sh:datatype xsd:integer .
```

Figure 2.17 Basic SHACL shapes graph

SHACL processors take a data graph (e.g., Fig. 2.13) and a shapes graph (e.g., Fig. 2.17) as an input and return a validation report structured as another RDF graph. If the validation report conforms, its data graph is very simple (Fig 2.18).

```
:report a sh:ValidationReport;
        sh:conforms true.
```

Figure 2.18 Validation report of a valid SHACL report

SHACL can assess datasets for two types of validation:

- Validation of requirements in terms of existence – assessment if the information on a certain object exists which is usually described in the ontology itself (e.g. each IfcSpace should have a property of IfcSpaceTemperatureWinterMin)
- Validation of specific values – usually project specific and found in the graph of the project itself (e.g. the value of IfcSpaceTemperatureWinterMin should be at least 21 degree Celsius)

If a dataset does not conform, the output of the validation report contains error metadata. This metadata contains additional information about the problems in the dataset, and can contain some or all elements from the table 2.2.

Table 2.2 SHACL error metadata (Gayo et al., 2018)

Property	Description
sh:focusNode	Node that caused the result.
sh:resultPath	Pointing to value of sh:path.
sh:value	Value node that violated constraint.
sh:sourceShape	Shape that given focus node validated against.
Sh:sourceConstraintComponent	The IRI identifying the violating component
sh:detail	Further details about the cause of the error
sh:resultMessage	Technical details about the error
sh:resultSeverity	Result severity

2.2.2. Linked Data

The Linked Data term was coined by Tim Berners Lee as a design note to the Semantic Web project. There are four stated main principles of the Linked Data (Bizer et al., 2008):

1. Use URIs as names for things
2. Use HTTP URIs so these names can be found
3. Provide useful information about these names through open standards (RDF, SPARQL)
4. Include links to other URIs

As an extension to (and variation of) Linked Data principles, Berners-Lee additionally defined a 5-star deployment scheme for Linked Open Data (LOD) promoting open-source nature and interconnectivity of knowledge. Each step is a continuation of the step before, as shown in Fig. 2.19.

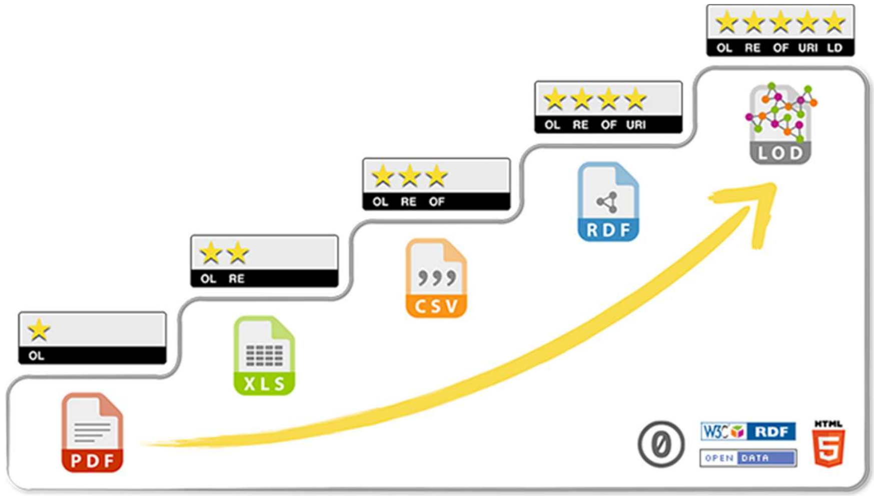

Figure 2.19 5 star schema for Linked Open Data (LOD) ("Linked Data - Design Issues," 2023)

1. Make data available on the web under open license
2. The data from step 1 should be structured
3. The data should additionally be published in a non-proprietary format
4. URIs should be used as identifiers
5. The data should contain links to other datasets for context

The Linked Open Data concept gained success through the Linked Open Data Cloud Project. More that 1200 data sets from a variety of knowledge domains and with a minimum of 1000 triples have been published so far, as shown in Fig 2.20.

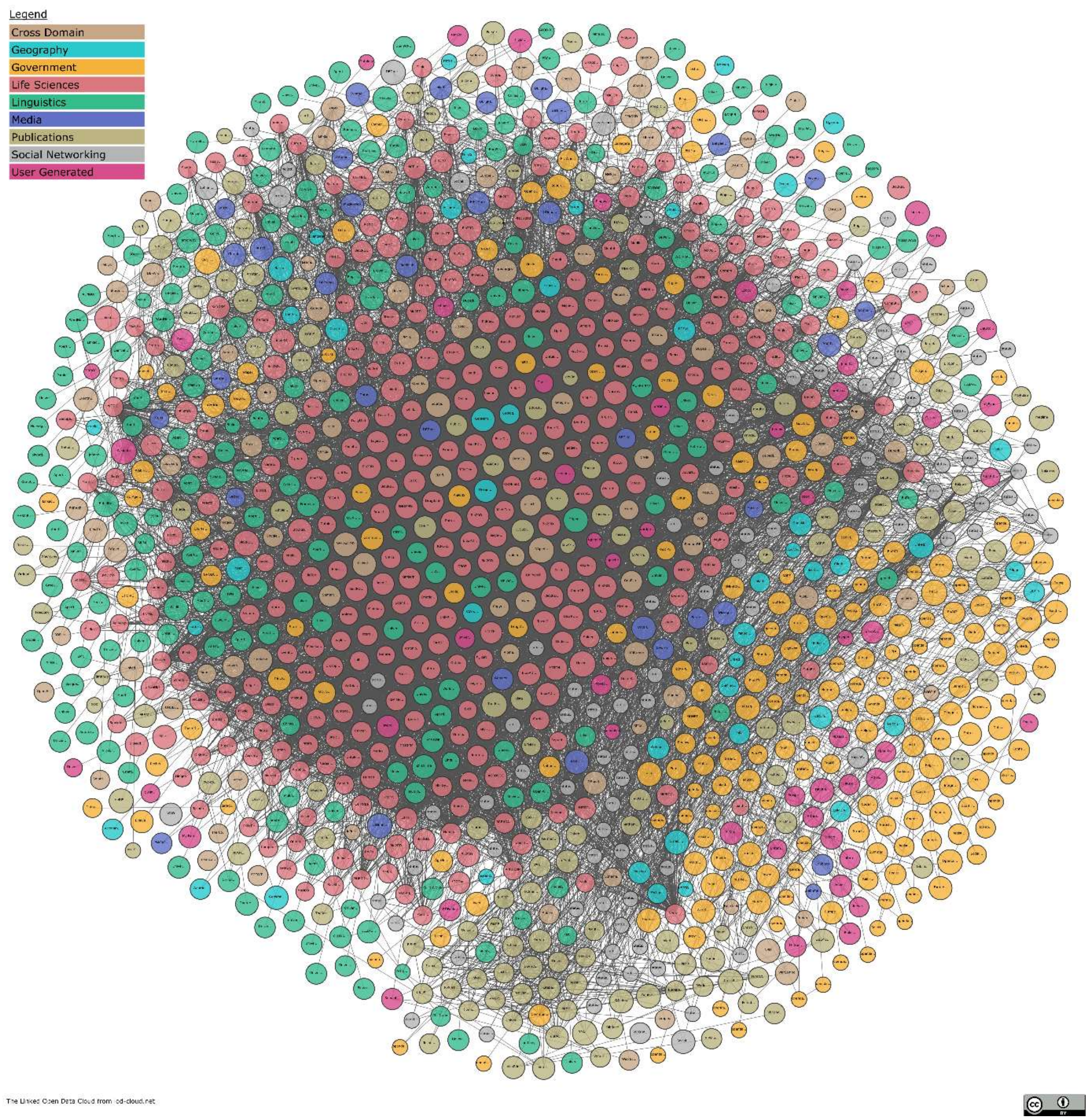

Figure 2.20 The Linked Open Data Cloud in November 2020 ("The Linked Open Data Cloud," 2020)

From this interpretation, it can be deduced that Linked Data is a subset of Semantic Web Technologies dealing primarily with data publishing. Therefore, the focus of the research will be on querying (SPARQL) and validation (SHACL), whereas more complex parts of the SWT such as trust, proof and security are left out of consideration.

2.2.3. Linked Data in Architecture and Construction

AEC industry is fragmented between many stakeholders, very often with unique workflows, standards and ways of structuring data. Moreover, to achieve comprehensive data management in the lifecycle of a built asset, many different forms of data should be harmonized. Building regulations, 3D models with different detail levels, management of construction items, data streams from the IoT systems, connection with smart building models are all challenges that could be solved with linked data. In example, current federated models are mostly created by geometrical superimposition with a lack of non-geometrical context. Linked data can help in creating semantically richer models in comparison to federated models for example by interlinking domain models as shown in Fig. 2.21. Use of linked data in AEC is gaining traction with development of proof-of-concept online tools that use Linked Data and Natural Language Processing (NLP) to create text-based queries of BIM models (Rasmussen).

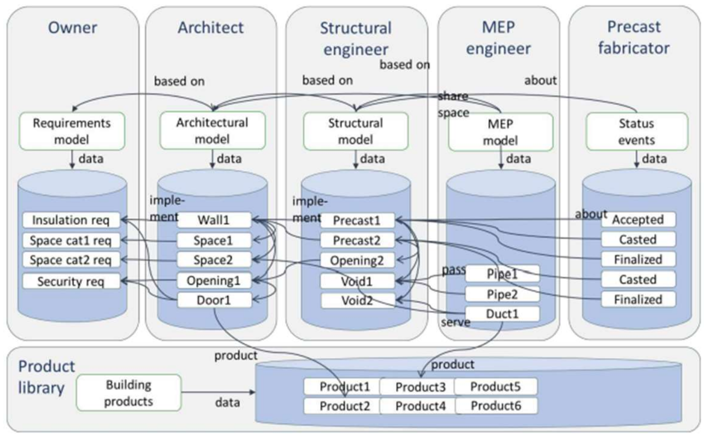

Figure 2.21 Workflow for interlinking of several domain BIM models(Borrmann et al., 2018)

Due to these and many other benefits, notable efforts were made to make BIM data (most often the IFC data model) available in Linked Data form. These attempts were made mainly using two methods (Terkaj and Pauwels, 2017):

1. Direct translation of the IFC data model in the Linked Data form (using OWL and RDFS)
2. Conceptualization of the IFC structure by using novel ontologies not directly related to the IFC data model

2.2.3.1. IfcOWL

As shown above, while the EXPRESS data modelling techniques and STEP serialization still form the basis of data exchange in BIM some of their inherent aspects limit the expressivity, as well as the ability to connect BIM data with other, heterogeneous forms of data. The IfcOWL is an attempt to overcome these limitations by the direct translation of the IFC data model in the linked data form, as one of two possible methods mentioned in the chapter 2.2.4. Several notable efforts in expressing the IFC data through direct translation were made, and resulted in official IFC2X3 and IFC4 IfcOWL versions made using the EXPRESStoOWL tool (Pauwels and Terkaj, 2016), that are to be used in the development of this paper. The official version of ifcOWL for the IFC4 Add2_TC1 version has 1798 classes, shown hierarchically on Figure 2.22.

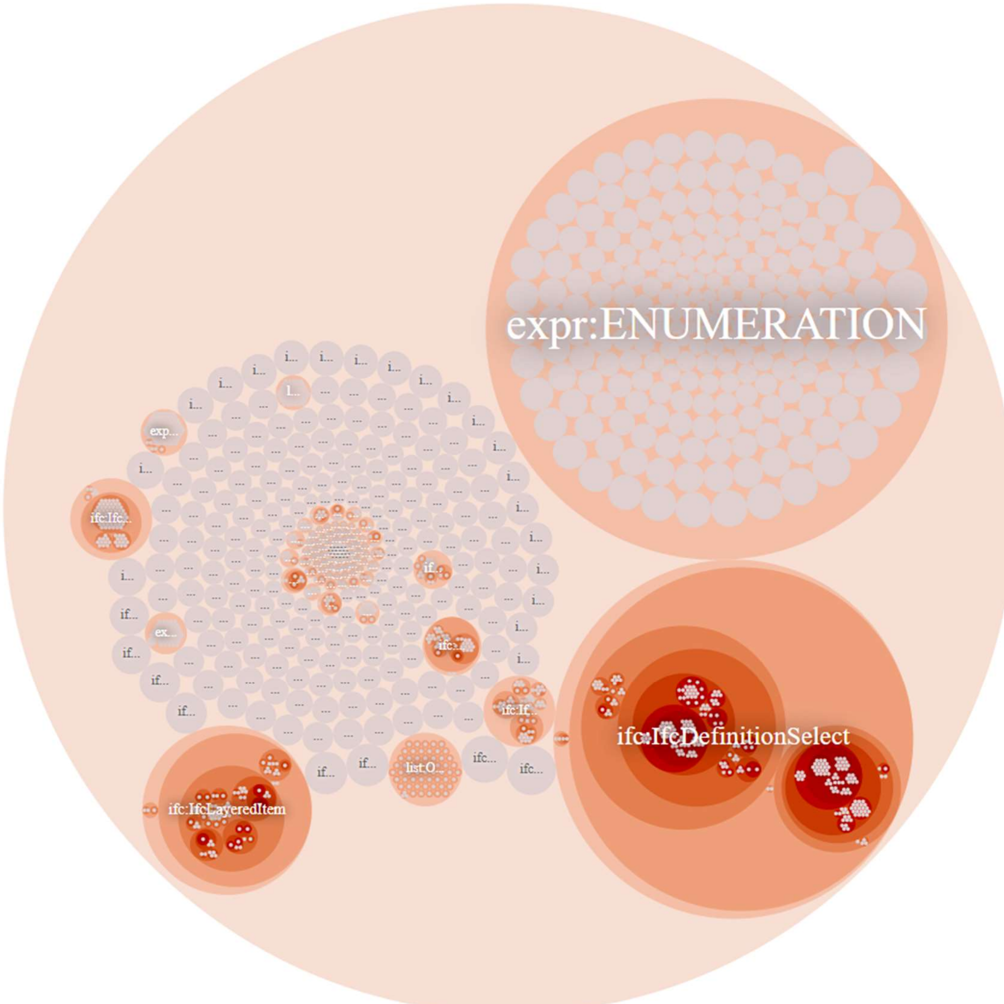

Figure 2.22 Class hierarchy of IfcOWL 4 Add2_TC1

Monolithic IFC structure still results in large and complex ontologies. Therefore, continuous research on possibilities of modularization of the IfcOWL continue, with the modular IfcOWL expected to improve usability, performance and alignment with other ontologies (Terkaj and Pauwels, 2017).

2.2.3.2. Linked Building Data

As mentioned, besides the direct translation of the IFC, modularity can be achieved by redesigning the existing Building Product Model. Notable efforts led by this notion were developed under the World Wide Web Consortium Linked Building Data Community Group (W3C LBD CG) umbrella, while other, independently created ontologies were aligned to the newly developed ones as a part of the efforts of the W3C LBD CG. All these ontologies aim to be simple, extensible, and modular, making it possible to interconnect them into a comprehensive building model. A graphic presentation of relevant ontologies from the LBD-CG and their interconnection with external ontologies in creation of existing datasets is shown in Fig. 2.23.

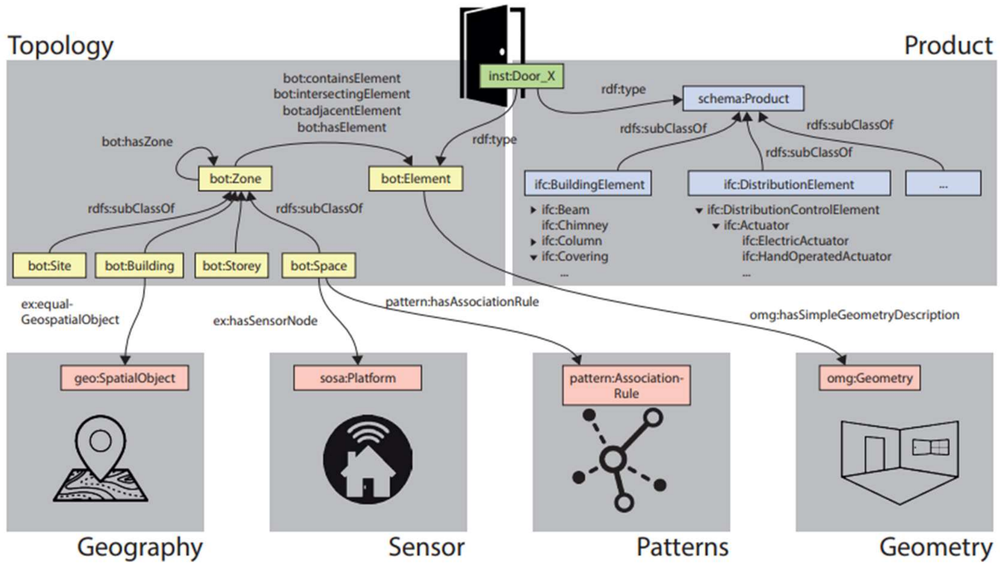

Figure 2.23 Conceptual overview of modules and ontologies in a Linked Building Data cloud (Petrova and Pauwels, 2021)

Some of the ontologies developed are:

Building Topology Ontology (BOT) - BOT is the core ontology of this ecosystem. Following the presented design principles, BOT contains just 7 classes, as shown in Fig. 2.24, making it simple but also extensible by other ontologies.

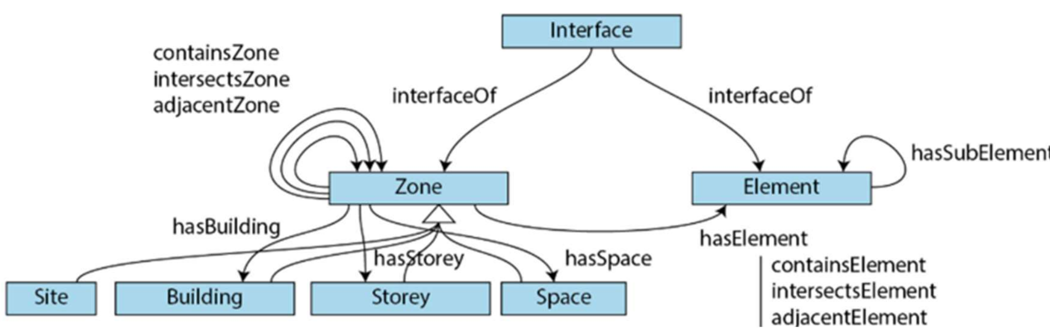

Figure 2.24 BOT ontology structure (Rasmussen et al., 2017)

Ontology for Property Management (OPM) – The ontology for property management describes temporal properties subject to changes as the building design evolves.

Uniform Project Ontology (UPonto) – facilitating data exchange and project analytics for capital projects

Building Element Ontology (BEO) – Ontology for building elements

Flow Systems Ontology (FSO) – enabling decomposition of flow systems, their mass and energy flows

External Ontologies related to the LBD community ontologies include:

Ontology for Managing Geometry (OMG) – linking between concepts and the geometry descriptions of these concepts, as well as between geometry descriptions and related properties

File Ontology for Geometry formats (FOG) – extension of OMG for schema specific geometry descriptions (e.g. DWG, Revit, E57)

Building Product Ontology (BPO) – describes building products in a schematic way, focusing on assemblies and component interconnections

Existing ontologies such as QUDT, SSN/SOSA and others can be aligned with these modular ontologies (Pauwels et al., 2022).

2.2.3.3. EN 17632-1:2022

While the efforts described in chapters 2.2.3.1. and 2.2.3.2. are very valuable to the overall development of linked data in the field of built environment, they deal only with the BIM subset of information in the lifecycle of one or multiple built assets. In reality, multiple information models exist besides BIM, such as GIS and electronic document management (EDM). An attempt to align conceptual modelling of various domains in the built environment was developed and formalized in 2022 in the Dutch NEN2660 standard, as well as its international counterpart EN 17632 the following year. This document addresses "the semantic and syntactic interoperability for the information describing assets going through their life cycle in the built environment."

Only the text part of the EN standard was available at the time of developing the research. Due to the unavailability of the international standard, further research was done using the NEN2660 Dutch standard and therefore, both terms (NEN2660 and EN 17632) are used interchangeably in the remainder of the research. The NEN2660 specification has 4 normative parts:

Terms (SKOS) – definition of terms made using the SKOS W3C recommendation

Classes and properties (RDFS) – made using the RDFS classes and properties, describing taxonomy of the SKOS terms

Classes and properties (OWL) – defines OWL class and properties for the definitions made in the RDFS file, enabling inference and OWA and NUNA assumptions

Shapes (SHACL) – defines SHACL shapes for the classes and properties in the RDFS file, enabling validation using the CWA and UNA assumption

Besides the normative parts, the specification has 4 informative parts (TriG, JSON-LD, single graph of constituent ontologies and RDF/XML), as well as 4 examples (2 bridges, road network, and a hospital).

2.2.3.4. Object Type Libraries (OTLs)

Object Type Library (OTL) is an informational model that can be built by combining ontologies, such as these mentioned along with many others. As formally defined in the Platform Linked Data Nederland, an OTL is "is a library with standardised object-types names (e.g. road, viaduct) and properties or specifications. An object is described with its object-type data, geometry data and metadata, Metadata are data (or information) about the data of objects. Metadata are needed because each object type has its own properties. How the object types are grouped is called an ontology. The OTL can be linked to a data dictionary, with the definitions of object-types." An example of a small OTL containing custom window types and their alignment to the IfcOWL and NEN2 660 ontologies is visible in the Figure 2.25

```
@prefix bs: <https://w3id.org/nen2660/def#> .
@prefix otl: <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ifc4: <http://ifcowl.openbimstandards.org/IFC4_ADD2#> .

otl:BuildingElement a owl:Class ;
    owl:subClassOf bs:RealObject ;
    rdfs:seeAlso ifc4:IfcBuildingElement ;
    skos:prefLabel "BuildingElement"@en .


otl:Window a owl:Class ;
    owl:subClassOf otl:BuildingElement ;
    rdfs:seeAlso ifc4:IfcWindow ;
    skos:prefLabel "Window"@en .

otl:WINDOW a owl:Class ;
    owl:subClassOf otl:Window ;
    rdfs:seeAlso ifc4:IfcWindow ;
    skos:prefLabel "Window"@en .

otl:SKYLIGHT a owl:Class ;
    owl:subClassOf otl:Window ;
    rdfs:seeAlso ifc4:IfcWindow ;
    skos:prefLabel "Window Skylight"@en .


otl:DOME a owl:Class ;
    owl:subClassOf otl:Window ;
    rdfs:seeAlso ifc4:IfcWindow ;
    skos:prefLabel "Window Dome"@en .
```

Figure 2.25 Snippet of an OTL structure in TTL format

2.2.4. Current status of IFC to Linked Data conversion

2.2.4.1. IFC-to-RDF Conversion Service

The first practical attempt to publish IFC data as linked data was published in 2012 (Pauwels and Deursen, 2012). Based on the previous strategies of general EXPRESS schema conversion, the IFC-to-RDF service developed a way to map IFC concepts into the nearest equivalents in OWL. The conversion process can be summarized in three distinct steps:

1. Generation of classes and properties

For each entity in the IFC EXPRESS Schema a corresponding owl:Class is generated along with the properties. IFC attributes are converted into OWL properties (owl:DatatypeProperties and owl:ObjectProperties). Simple datatypes are transformed using the owl:DatatypeProperty. A naming issue is encountered based on the differences between IFC (CWA) and the OWL (OWA) which was solved by appending integers to the OWL property names with the naming conflicts.

2. Basic restrictions for classes and properties

A hierarchical ontology structure is created using the rdfs:subClassOf for relation in express (e.g. IfcBuildingElement is rdfs:subClassOf IfcElement). The classes on the same level (e.g. IfcBuildingElement, IfcGeographicElement, IfcCivilElement) are expressed with the owl:disjoint construct. Additionally, the generated properties contain the rdfs:Range and rdfs:Domain constructs. The enumerations (such as SKYLIGHT, DOME, WINDOW for IfcWindow) are expressed using the owl:one of construct.

3. Advanced restrictions for classes and properties

This step should represent some of the more advanced features of the EXPRESS schema, such as cardinality restrictions and value restrictions and OPTIONAL, UNIQUE and DERIVE keywords of EXPRESS schema.

This approach resulted in instantiation of the ifcOWL ontology, as well as an application written in Java that converts an IFC model to an RDF Abox graph structured according to ifcOWL, showing that linked data is a valid approach for addressing existing interoperability issues, but with consideration of mapping process between information models and their RDF representation.

2.2.4.2. IFC to Linked Building Data Converter

The IFC to Linked Building Data Converter continues the work done in the previous step, aiming to overcome the limitations of the ifcOWL schema such as size and complexity by connecting it with the Linked Building Data ontologies described under 2.2.3.2.

The implementation of the converter is done in Java by utilizing the IFC-to-RDF converter to temporarily convert the IFC file to an ifcOWL Abox graph. Upon creation of this graph, the converter creates the LBD nodes starting from the IfcSite towards IfcBuilding instances, then following the IfcBuildingStorey (of the previously found IfcBuilding) etc., until all the IFC building elements and properties are found, as shown in Fig. 2.26. Such approach results in simplified and more user-friendly graphs easier to query.

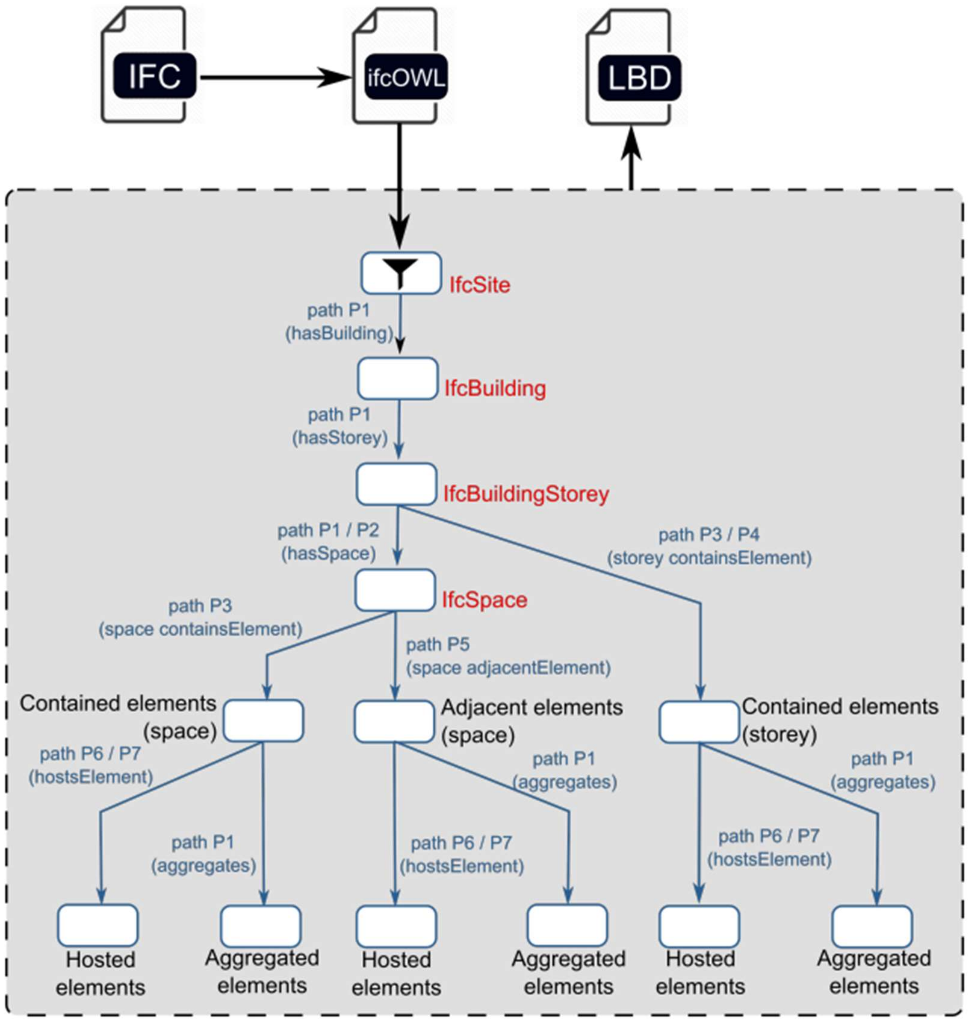

Figure 2.26 The implementation of IFCtoLBD conversion process (regarding BOT ontology) (Bonduel et al., 2018)

2.3. State of the art in digital building permitting

The last part of the state of the art is to consider relevancy of the researched topic by looking at at research trend in digital building permitting. Before execution on the construction site, every project in the Architecture, Engineering and Construction (AEC) industry, needs to apply for a building permit to the public authorities. In this process, the design is checked against regulations and based on . outcome, the building permit is granted or rejected. The process of building design is very complex and fragmented, with 10 – 12 disciplines participating in an average real – estate development project (Kovacs and Micsik, 2021). This results in a very high number of requirements and regulations, which are checked manually with low efficiency and accuracy, as well as high cost. The World Bank survey of 190 economies has shown that only 27% of the economies use an e – submission based process, and OECD economies are leading with 48% implementing efficient e-submission formats (Chakaroun et al., 2023). To put this state of the art in context, a framework derived from (Noardo et al., 2022) shows conceptual evolution of a completely manual, paper-based process to a completely automated and integrated process. Lagging economies are on level 1, leading economies are on the level 2, and further digitalisation to levels 3 and 4 will lead to significant increases in in efficiency, enabling faster and more transparent process while reducing costs (Noardo et al., 2022).

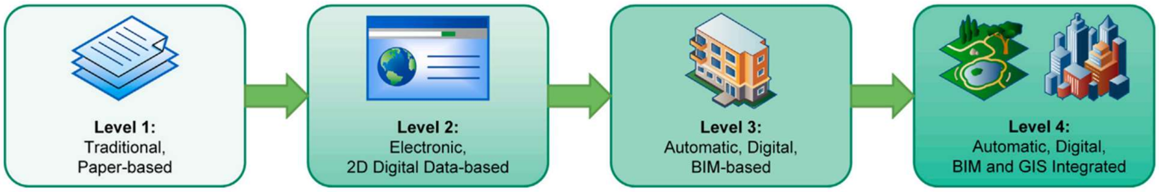

Figure 2.27 Levels of building permitting (Noardo et al., 2022)

These benefits are the reason why automated compliance checking (ACC) for digital building permits has been researched for more than 50 years (Zhang et al., 2022). However, the research in the field of ACC has seen significant uptake in the last seven years, in connection with significant increase in the adoption rate of digital representation of building and geospatial data across the world (Harrie and Jensen, 2018). BIM aims to capture all relevant information in the life-cycle of a built asset and therefore presents an optimal information container to check against specific regulatory requirements in the design stage of a construction project, while GIS enables analysing these BIM models in a broader context of the whole city. In the rest of the chapter, overview of the state-of-the-art academic research in the field as well as the factors affecting adoption will be presented. The chapter will conclude with the most notable practical developments currently applied in Europe.

2.3.1. Academic research

A relatively new study from January 2022 (Noardo et al., 2022) classifies the developments in academic research of digital building permit depending on the scope of the research in the whole digital building permitting process, divided in 8 steps as shown on Fig. 2.28.

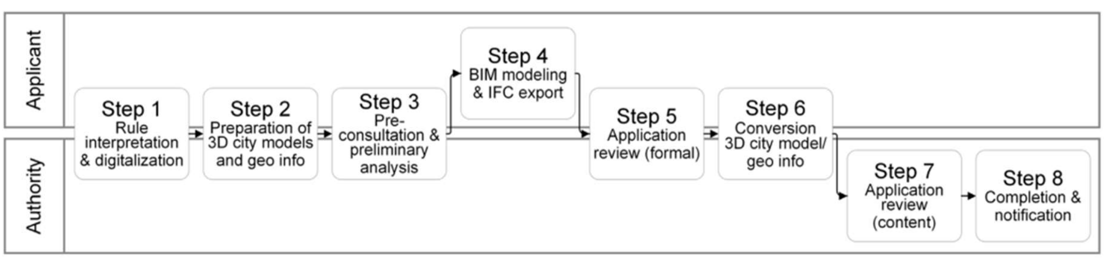

Figure 2.28 Contributions per step (Noardo et al., 2022)

From the perspective of total number of papers published per year in the period of 2001 - 2020, a significant uptake in academic research from 2015 onwards is noted, proving an increasing interest in the field. The research shows that majors efforts are undertaken to digitize regulations and the technical aspects of ACC. All the other important fields, such as scalability of solutions, education of public officers and efficiency of joint BIM – geospatial systems are still not addressed sufficiently.

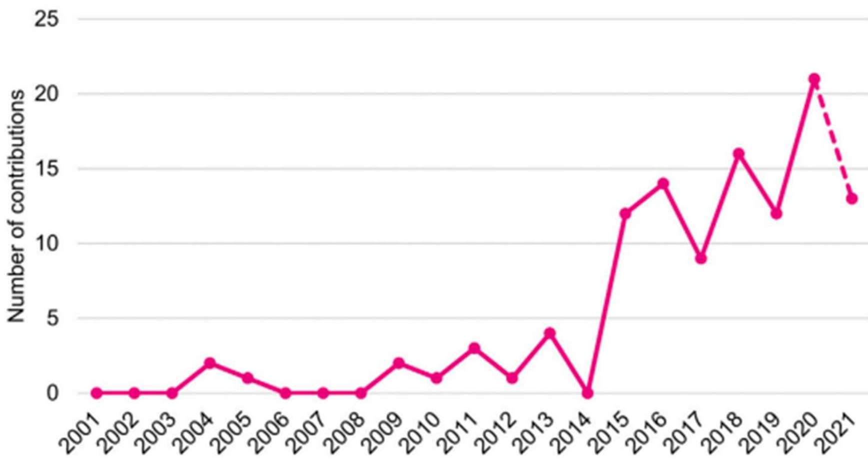

Figure 2.29 Contributions per year (Noardo et al., 2022)

Several important conclusions in the context of this research can be made. Firstly, the interest in continuously growing, but it is not distributed equally per steps. Actually, the step receiving most attention is coming near the ending. The research that will be carried out as a part of this work still mostly deals with this step but will try to take into consideration the need for a holistic approach to such a complex process, where it is applicable.

2.3.2. Factors affecting adoption

Specific research on BIM-based building permitting is limited, but many studies about the factors affecting the BIM adoption generally were carried out in countries such as China, Finland, Norway, Singapore and USA, among others. A study systematically reviewed existing research on BIM adoption in general (Ullah et al., 2020) and identified three main groups of factors, presented in table 2.3.

Table 2.3 Factors affecting BIM adoption in general (Ullah et al., 2020)

Technological factors	Compatibility
	Complexity
	Trialability
	Relative advantage
Organizational factors	Top management support

	Behavioural intention
	Training and learning
	Leadership
	Innovativeness
	Awareness
	Motivation
	Trust
	Organizational culture
Environmental factors	Client pressure
	Competitive pressure
	Partner pressure

Additional research evaluated detected factors against a qualitative research through conducting interviews with 7 stakeholders with different roles in the process, from municipalities to software developers (Ullah et al., 2022). which acts as the public authority responsible for issuing the building permits, certificates of occupancy and demolition permits. The case study shows that both technical and non – technical factors are important, with organizational awareness and top-management support having a positive effect on the BIM adoption. Some factors affecting BIM adoption in general are applicable to the building permit process, with addition of others, that are building permit specific. A list of all the factors affecting BIM adoption for the building permit process is shown in table 2. Interpretation of the data collected formed a categorization of factors affecting digital building permit process in the same three categories as above, as shown in table 2.4.

Table 2.4 Factors affecting BIM adoption for the building permit process (Ullah et al., 2022)

Technological factors	Complexity in developing and using BIM-based building permit system
	Relative advantages/disadvantages of BIM for building permits
	Existing building permit system
Organizational factors	Management support for BIM-based building permit process
	Organizational culture
	BIM awareness
	Training and learning for BIM-based building permit process
Environmental factors	External pressure
	Legal context

Several factors are found important in the context of this research. Firstly, the high software cost is often a barrier to BIM adoption to both private and public enterprises. However, the differences occur in the technical context of software utilization. Many tools exist for common BIM workflows related to the design stage. Whereas the out-of-the-box commercial tools are viable for use in standardized design workflow common in private enterprises, public institutions have a much different use-case, the one that is not dealing with information production, but its validation against specific requirements. Therefore, development of BIM tools tailored to specific institutions is needed. These tools are currently missing, so the cost is not the only barrier to entry, but also a lack of available tools. Considering the lack of experts in public institutions as well as the lack of external pressure from the market, the newly developed tools should be as easy to use as possible. A web-based application is therefore perceived as most suitable technology. Additionally, although the complexity of the industry makes interoperability a necessity, private actors can still decide to keep their developments in a closed proprietary ecosystem if the circumstances allow so. Public institutions are bound to prescribe their demands in non-proprietary, open formats. Therefore, most of the compliance checking research and deployment solutions concentrated on the use of IFC as the standard in building related information exchange.

2.3.3. Notable initiatives

Implementation through pilot projects started in the late 2000s with the CORENET ePlanCheck project in Singapore. Out of other notable non-European countries, Korea and New Zealand, as well as UAE are developing programmes that aim to deliver digital building permitting in practices.

Focusing on Europe, several projects are being developed. A non-exhaustive list of developments, derived from (Noardo et al., 2020) is shown in table 2.5 below.

Table 2.5 Notable European efforts in practical implementation of digital building permits (Noardo et al., 2020)

Country	Organization/project
Finland	KIRA-Digi, Sova3D
Norway	eByggeSak
Sweden	SmartBuiltEnvironment
Germany	Xplanung, Xbau
Netherlands	Several municipalities, e.g. Rotterdam
United Kingdom	Centre for Digital Built Britain
France	CTSB
Slovenia	e-prostor
Italy	Lombardia
International	buildingSMART Regulatory Room, EU-BIM Task Group

All these efforts accomplished certain successes and brought value to their respective communities. However, they are mostly characterized by strong bias towards a certain discipline or the nation in which the project is developed in. This is obviously necessary as a starting point but does little in terms of scalable deployment of digital building permitting wider than the country of the pilot project itself. To tackle these challenges, an international strategy is necessary.

2.3.3.1. European Network for Digital Building Permit (EUnet4DBP)

The idea of a unified strategy was formalized in 2020 through formation of the European Network for Digital Building Permit (EUnet4DBP). The network aims to create a diverse network of actors from different institution types, ranging from research, public institutions, and governmental institutions, all the way to private enterprises and freelance individuals. These actors have diverse knowledge across multiple domains such as BIM, GIS, building regulation and software development. Collaboration across disciplinary and geographical constraints should produce a strategy to develop digital building permit tools and efforts in a flexible, scalable and reusable way (Noardo et al., 2020). Three main pillars of EUnet4DBP were defined as process, rules and requirements, and technology. Based on these pillars, a list of ambitions was developed, show in interaction on the Fig 2.30.

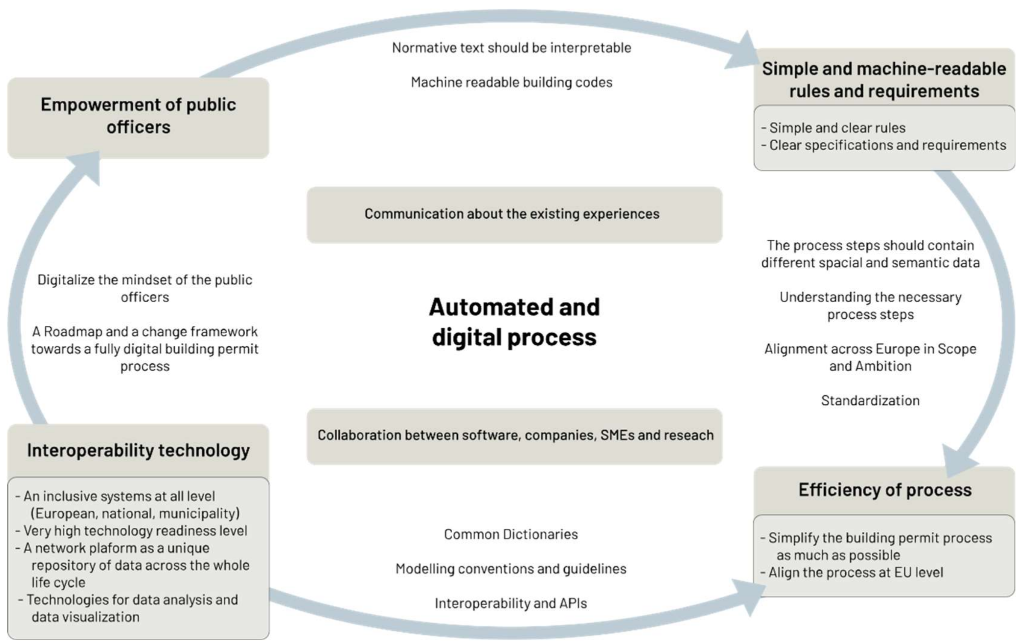

Figure 2.30 Ambitions of EUnet4DBP (*EUnet4DBP*)

2.3.3.2. DigiPLACE

To facilitate the adoption of digitalisation and interoperability in the AEC industry DigiPLACE project aimed to develop a framework as a base for future development of digital construction platforms. From September 2019 and in duration of 18 months, a consortium of 19 partners from 11 countries was coordinated by Politecnico di Milano to produce a set of common guidelines named Reference Architecture Framework for digital construction platforms. These guidelines aim to enable interoperability and data sharing in construction among all relevant stakeholders. A perimeter of public digital platforms is proposed in DigiPLACE, as shown in the Fig 2.31.

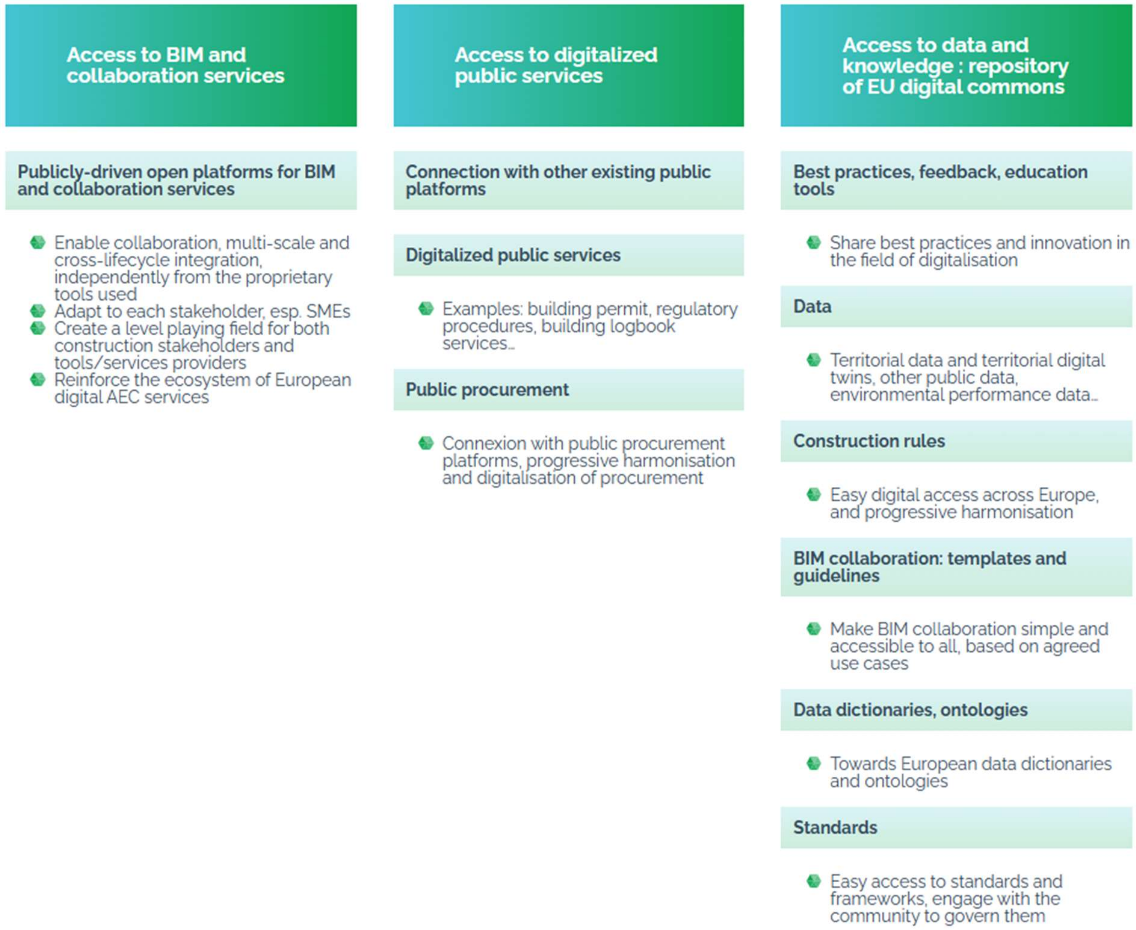

Figure 2.31 Proposed perimeter of public digital platforms for construction (*DigiPLACE*)

Most points stated in DigiPLACE apply to this research to a certain degree, with the most notable being the digitalised building permit service and establishment of a need for common European data dictionaries and ontologies.

2.3.3.3. Horizon Europe digital building permit projects

As was the case of many notable scientific advancements in the past, digital building permitting is also fuelled by funding from the European Union. To the best of this research knowledge, there are currently three related Horizon Europe projects aiming to develop, deploy and test digital building permitting platforms: ACCORD, CHEK and DigiChecks.

There are several notable European Union initiatives that interact with each other on the way towards a green & digital transition, such as EU Green Deal and the New European Bauhaus. These are the building blocks of the ACCORD project. The project, lasting 36 months starting September 2022, brings together 21 partners from 11 countries with the goal of helping European countries to automate and digitise their permitting and compliance processes on the way to more sustainable built environment. A semantic framework will be developed for processes, documents, standards, data, and tools. The framework should ease the design and implementation of building permit and compliance checking of construction, renovation, and demolition projects. Input data for the permitting processes must be in the form of BIM and GIS models ("ACCORD Project," 2023).

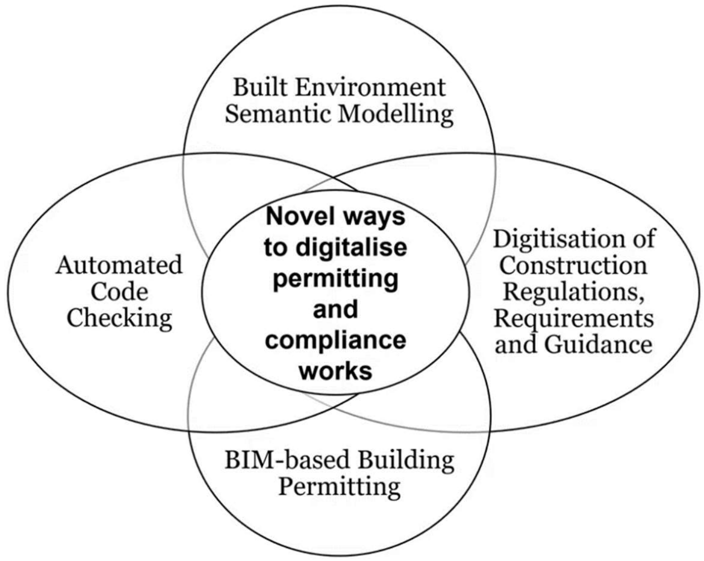

Figure 2.32 Building blocks of the ACCORD project (*Towards automated regulatory compliance in the EU*, 2023)

DigiChecks continues the findings of the DigiPLACE project, aiming to implement them in a new digital framework that would enable permit authorities to build permit solutions faster and more secure. Starting in June 2022, the project will last 36 months and connect 13 partners from 5 countries. Building on existing and developing new technologies, the project aims to develop a permit platform to facilitate interaction of the stakeholders of the permitting process. The technological solution itself will rely on linked data concepts to overcome interoperability constraints between the public rules and requirements and the data produced in the design stage (IFC, gbXML, CityGML).

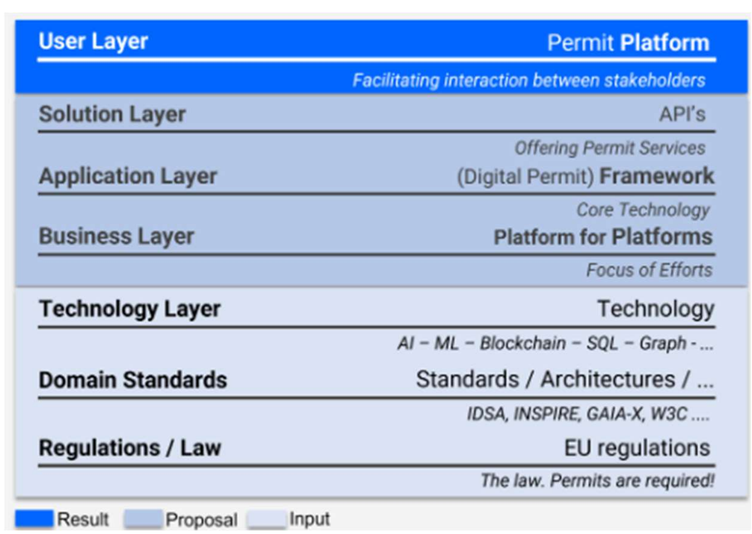

Figure 2.33 Layers of abstraction in the DigiChecks project ("DigiChecks," 2023)

The research in DigiChecks project is currently in the stage of design of a framework encompassing system architecture, interoperability, and data sovereignty. As a part of the research, three high-level requirements were formulated for Permit Requirements Management (PRM) (Bazuin et al., 2023):

- PRM must be able to process pre-existing document-based requirements
- Data exchange must be based on open and widely accepted structured data formats
- Non-technical users must be able to execute PRM utilizing current processes

DigiChecks research defines 4 levels of requirement explicitness and automation potential. Currently widespread document requirements (1), text-based requirements (2a), enriched text-based requirements and finally model-based requirements (3).

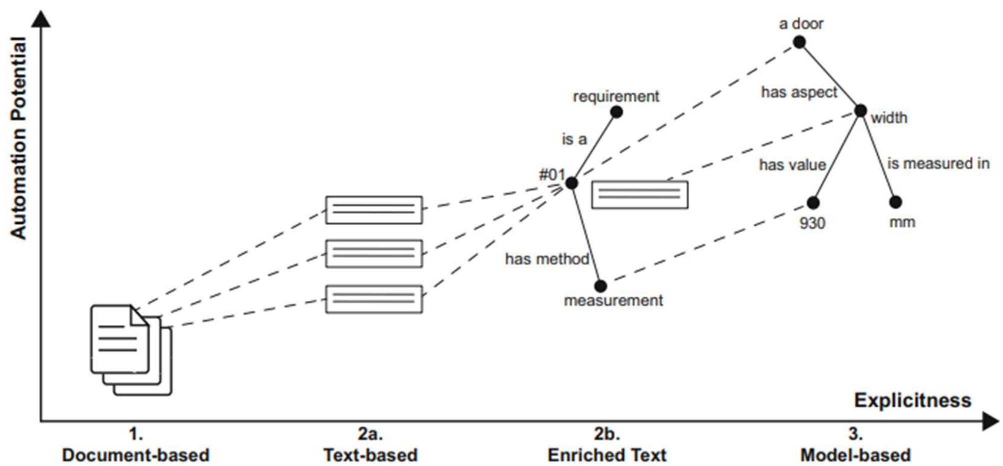

Figure 2.34 Explicitness versus automation potential in Permit Requirements Management (Bazuin et al., 2023)

Higher levels of explicitness are more suitable for process automation than lower levels. However, the engineering process often deals with imprecise steps and therefore a fully model based approach could be impractical. Therefore, a combination of text and model-based requirements is considered a promising direction to pursue.

The last of the three projects, CHEKdbp aims to provide a methodological and technological kit that will address barriers such as different regulatory context, knowledge and interoperability gaps, scalability of the solutions. The project aims to develop Regulatory Information Requirements that will be translated into IFC and CityGML specifications. The project, starting in October 2022 will last for 3 years and include 19 partners from different fields such as researchers, municipality officers, software developers, among others.

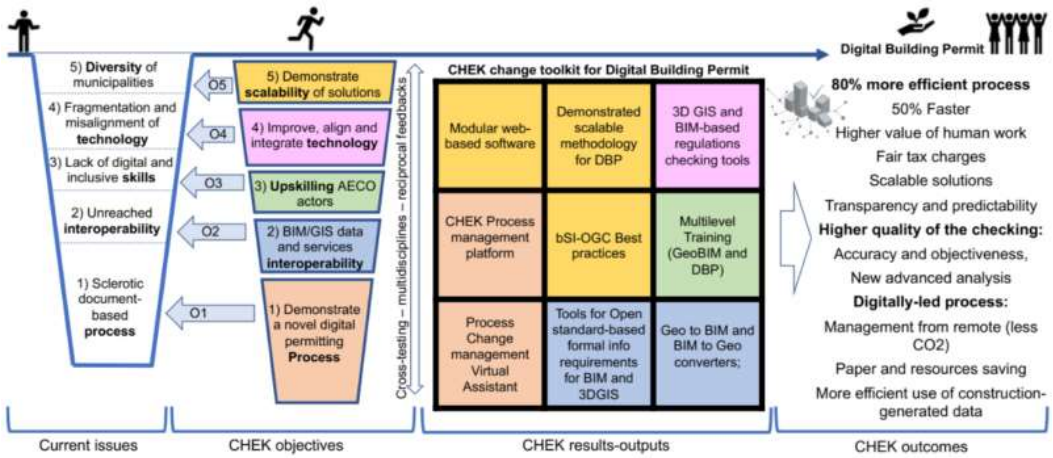

Figure 2.35 Outline of the CHEKdbp project ("CHEKdbp," 2023)

All the presented projects will coordinate their efforts to ensure alignment. Nearing the end of the projects, a testing phase on different use-cases will be done, to examine the developments made in the projects.

Table 2.6 Pilots of the relevant digital building permitting projects

Project	Country	Use Case
ACCORD	Finland & Estonia	BIM based building permit and environmental compliance
ACCORD	United Kingdom	Structural Integrity checking
ACCORD	Spain	Urban regulation checking
ACCORD	Germany	Land use permitting, Green building certification, architectural design compliance
DigiChecks	Austria	Office building
DigiChecks	Wales	Civil Engineering scenario
DigiChecks	Spain	Residential building

CHEKdbp	Italy	Ascoli Piceno
CHEKdbp	Portugal	Gaiurb EM
CHEKdbp	Portugal	Camara Municipal Lisboa
CHEKdbp	Czech Republic	IPR Prague

2.3.4. Relevant findings

The research of the state of the art in digital building permitting provided several important inputs to the developments of this research, both in the importance of the research direction in general, as well as specific facts that will shape it. Some were mentioned before but will be listed here again for clarity and emphasis of their importance in the development process.

The findings that show general increase of interest in research in digital building permitting:

1. Even in the most developed countries, the rate of BIM-based digital building permitting is extremely low, but the deployed projects prove significant possibilities of savings
2. The academic research in the field of digital building permitting is steadily growing since 2015
3. Pan European clusters are increasingly created and expanded with an increasing number of stakeholders willing to participate in the research of digital building permitting
4. Implementation through development and pilot deployment gained momentum through three related Horizon Europe projects starting in the second half of 2022

The findings that help shape directions of this research specifically:

1. Technological solutions should be based on the open standards
2. Technological solutions should be as easy to use as possible
3. It is unlikely that fully model-based solutions are feasible, a hybrid text and model-based approach seems to provide balance between automation potential and ease of use
4. Despite efforts of alignment, there is little chance that a one-fits-all solution will be possible. For mass deployment flexibility and customization is a key attribute
5. The technological ideas that shaped the outline of this research are well accepted in practice, and are currently being developed in at least two out of three presented European projects

3. THEORETICAL FRAMEWORK

3.1. High – level overview

Based on the findings presented above, a framework was devised that would connect the IFC data model and the NEN 2660 ontology. The framework follows common approach to building Linked Data ontologies, as shown in Fig. 3.1.

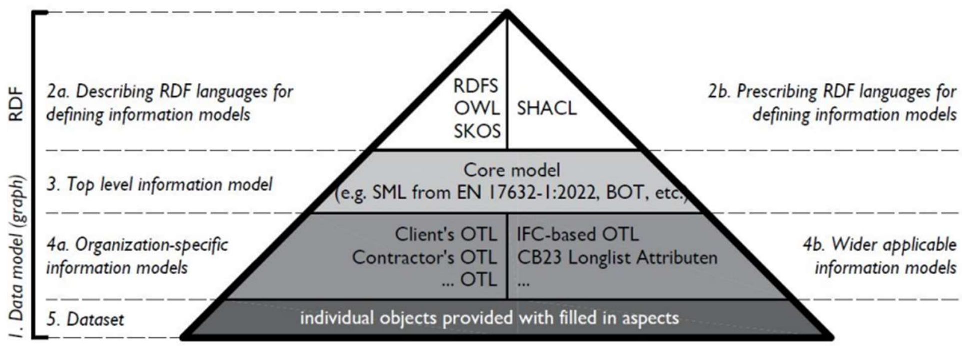

Figure 3.1 Building blocks of an RDF graph (Bonduel, 2023)

The graph starts with the creation of the descriptive top level information model (found under 3. in Fig 3.1.) expressed with descriptive RDF vocabularies (SKOS, RDFS, OWL). In this case, such information model is created by merging the core ontology from NEN 2660 (EN 17632) and the ifcOWL version of the IFC schema. An additional part of this top-level information model is to prescribe what should be contained in the dataset, done with the help of SHACL. While not a part of this research, it is useful for understanding the context to add that such information model can be further extended on an organizational level (4a. in Fig 3.1.), as shown on an example in Fig. 3.2. The value of this approach lies in the fact that each stakeholder has the possibility of extending concepts according to their specific workflow, while still maintaining the possibility of communication with other stakeholders by a common, top level information model that will be built as a part of this research.

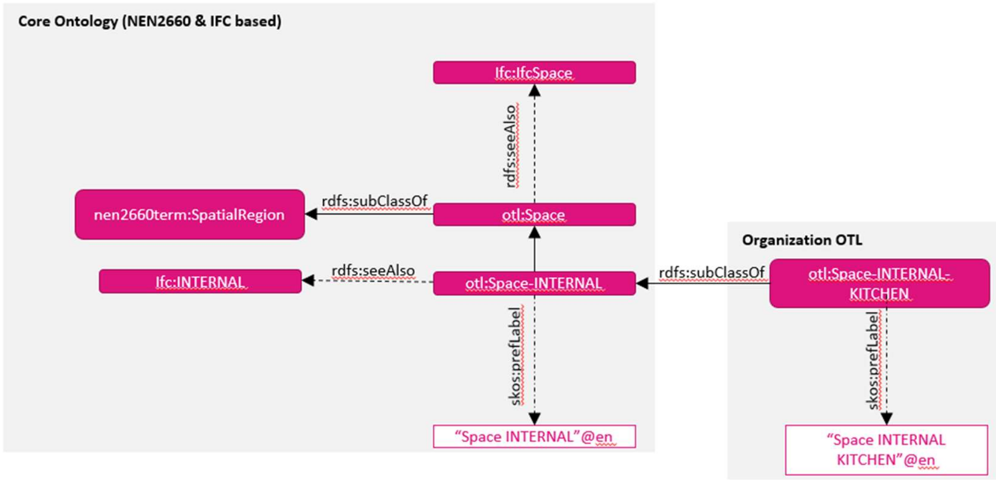

Figure 3.2 Example of extending the top-level information model with organization specific concepts

The next step is to instantiate a dataset of individuals and connect it to the OTL concepts as well as assert relations between these individuals, resulting in a data graph such as shown in an example in Fig. 3.3.

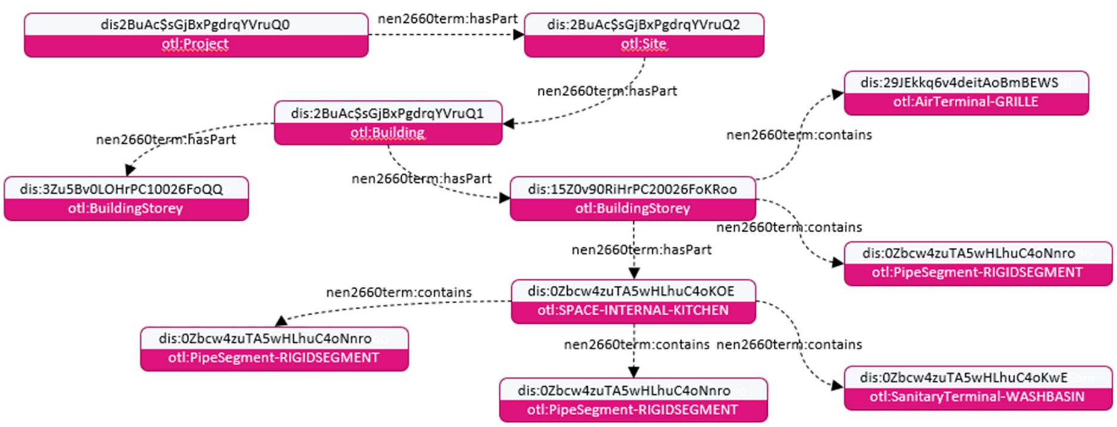

Figure 3.3 Example of relations between individual resources in a dataset

Finally, a separate shapes graph needs to be created to enable validation of the conformance of the dataset to the Object Type Library.

Having this in mind, the proposed workflow will have three separate parts, shown in interaction in Fig. 3.4.

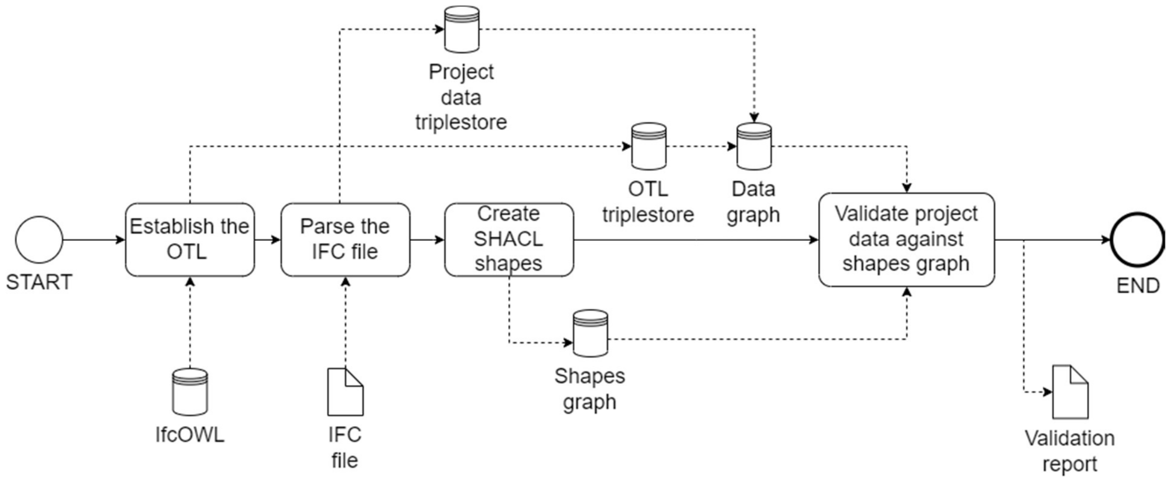

Figure 3.4 High level overview of the flow

Additionally, since the IFC schema is quite large and complex, for the sake of this research it will be simplified in the following ways:

- Only a subset of the schema will be considered – The current focus is on physical elements (subclasses of IfcBuildingElement, IfcDistributionElement, IfcFurnishingElement and IfcElementComponent), openings (subclasses of IfcOpeningElement) and spatial elements (subclasses of IfcSpatialElement). These will be considered and expressed in conformance with NEN2660 (EN 17632) while the rest will be disregarded. The part of the schema that will be analysed belongs to the Product Extension of the IFC.
- Modularization – The considered part of the IFC will be split, meaning the process will be repeated for each of the parts below
    - Taxonomy – Firstly, the taxonomy will be established by querying ifcOWL and connecting the found classes and enumerations with the NEN2660 concepts.
    - Relations – To establish the relations between different resources on the dataset level in conformity with NEN2660, IFC file will be parsed and certain IFC relations will be replaced with adequate NEN2660 concepts.
    - Attributes - The ifcOWL ontology will be queried for attributes of classes queried in step 1. These attributes will be appended to newly created classes but aligned to the EN 17632.
    - Properties – IFC properties are not directly connected to with an IFC instance (as is the case with IFC attributes), but through IfcRelationships on type (HasPropertySets) or instance (IfcRelDefinesByProperties) level. The IFC properties are not available directly in the ifcOWL, but in XML. The properties should first be converted to RDF with the help of RDF Mapping Language (RML) and appended to the OTL classes aligned to the EN 17632.

3.2. Taxonomy

The taxonomy of the OTL will connect NEN2660 and IFC by querying the ifcOWL and returning individual entity, all of its enumerations as well as the class which it is originating from. Three types of OTL classes will be created, as is shown in Fig. 3.5:

1. Generic IFC classes such as otl:BuildingElement, otl:SpatialElement. These will have a direct connection with the NEN2660 concepts

2. Specific IFC classes such as Window and Space will be created in the OTL and connected as subclasses to more generic OTL concepts created in step 1

3. Specializations of IFC classes, defined in IFC as enumerations of IFC PredefinedType attribute will be created as individual classes in the OTL and connected as subclasses with the classes created in step 2 (therefore indirectly being connected with the more generic concepts from step 1)

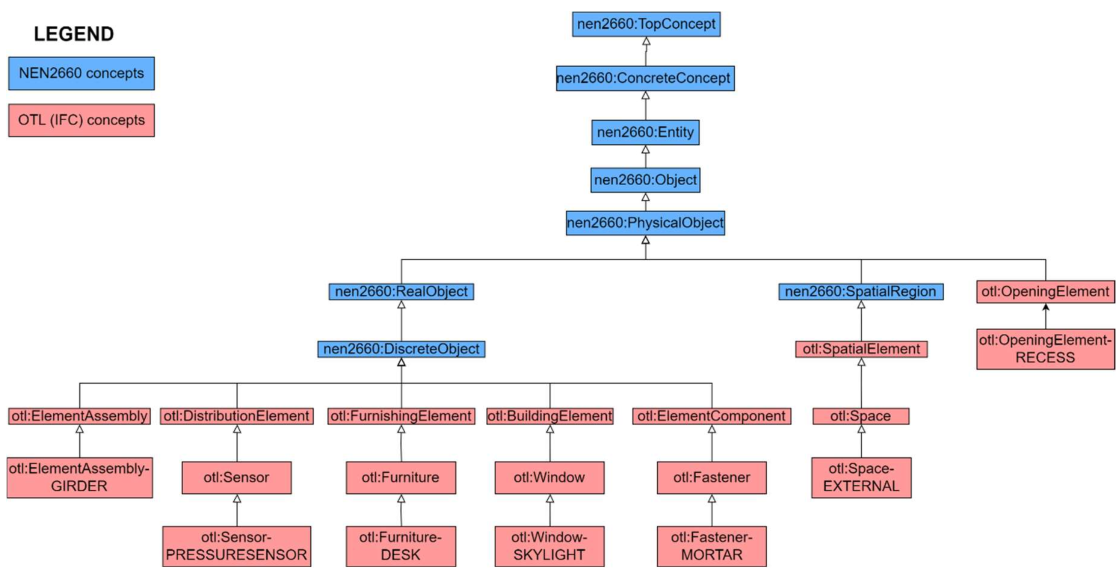

Figure 3.5 Connection of NEN2660 and IFC in the OTL taxonomy

3.3. IFC Attributes

3.3.1. Distinction between attributes and properties

After establishing the taxonomy of the object type library, the next step is to analyse the ifcOWL and the attributes of each of the classes. However, before this next step it is important to clarify the difference between attributes and properties. Some see the difference in the fact that attributes are ascribable, whereas properties are possessable (Bendiken, 2011). Similarly, some consider attributes a concept that it attributed to another, while properties can exist without any attribution (George, 2014). Often used interchangeably (even the Merriam-Webster dictionary lists both concepts as "something that sets apart an individual from the others of the same kind"(*Merriam-Webster*)) the distinction can often be quite unclear and context dependent, and the two words are often used as synonyms.

In the linked data context of EN 17632 properties can be both attributes and relations („There are many ways to model properties that is attributes (qualities and quantities) and relations." (Koehorst et al., 2022)), however, even in the standard both are used interchangeably („The most simple and direct modelling of an attribute in e.g. in OWL by an owl:DatatypeProperty" (Koehorst et al., 2022)).

Another important context for the research is the object-oriented context present in the IFC schema. In this context, the attributes are present on a class level, directly related to each IFC entity and inherited from the parent class by the children's classes. Properties, however, are present on instance level, they are not directly related to the IFC entities but through intermediary relationships (IfcRelDefinesByProperties) and are not inherited. Therefore, the modularization of the IFC considers this subtle distinction between attributes and properties (as described in chapter 3.1.) but will generally take the approach of EN 17632 where both terms are used interchangeably, with the term "property" being preferred.

There are 3 levels of property modelling commonly agreed on in the Linked Building Data community, dedicated datatype properties (level 1), objectified property nodes (level 2) and objectified property state nodes (level 3), as shown in Fig. 3.6 (Bonduel and Wagner, 2023).

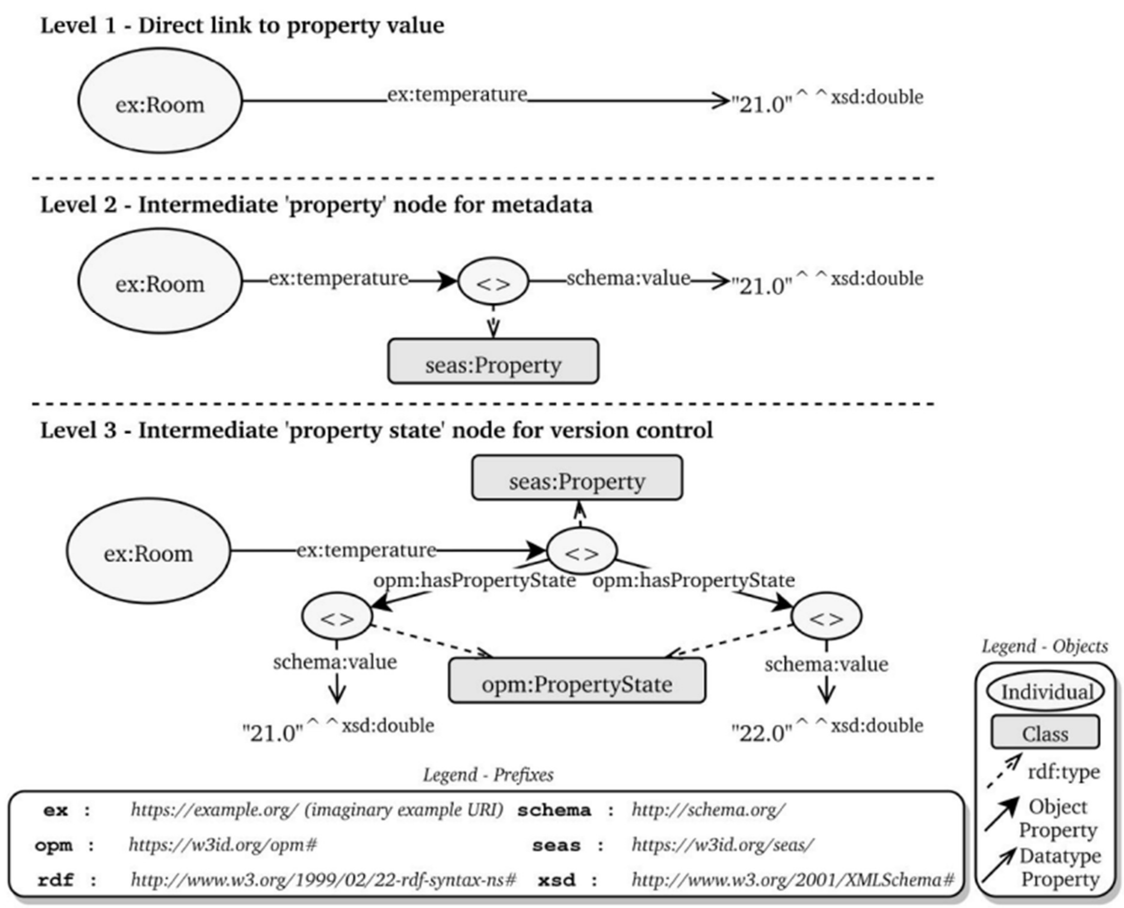

Figure 3.6 Levels of property modelling (Bonduel and Wagner, 2023)

3.3.2. Dedicated datatype properties – Level 1 property modelling pattern

In the simplest way of property modelling, there is a direct link to property value on Tbox level, these properties are modelled as owl:DatatypeProperty with an additional relation of rdfs:range that expresses the datatype that should be used for expressing such property. Such a property expressed with owl is descriptive and can only be used for inferencing, not for prescriptive, so an additional relation of sh:datatype can be added to ensure validation of the datatype. as shown in Fig. 3.6 on the TBox level.

Benefits of level 1 modelling are direct reusability of datatype properties from domain taxonomies and smaller graphs due to less nodes, resulting in high querying performance. However, additional information such as units cannot be added, retrieving all properties is more complex and properties cannot be referenced (Bonduel and Wagner, 2023). Simple IFC attributes that do not contain a unit of measurement, such as the name, description or GUID were modelled in this way.

3.3.3. Objectified properties with property node – Level 2 property modelling pattern

This, more complex way of property modelling introduces an intermediary node that objectifies the property and enables additional statements such as explicit introduction of quantity kind and units. Units can be explicitly mentioned or inferred if an already existing ontology is used to describe quantity kinds (such as QUDT). Level 2 modelling enables referencing of properties since they exist as individual nodes, while level 1 relations can be inferred. Naturally, the graph becomes larger, with more nodes and slower performance (Bonduel and Wagner, 2023). An example of object properties constructed with the level 2 modelling technique is shown in Fig. 3.9.

The third level is will not be described in detailed since it will not be used in the remainder of the research. Briefly, it adds additional intermediary node, called property state that can be used to describe value of a certain property at a certain point in time, thus enabling expression of evolving properties. This enables referencing of property values as well as inferencing of level 2 in addition to the already existing level 1. The drawback is decreasing of querying performance.

3.3.4. Property modelling according to EN 17632

Having the three levels in mind it is important to structure the OTL in compliance with the rules of EN 17632 shown in Fig. 3.7. The standard specifies a mix between two levels, level 1 should be used for qualitative properties and level 2 for quantitative properties, as shown in Fig. 3.8, while the datatypes should be assigned according to the xsd ontology and the units according to the qudt ontology.

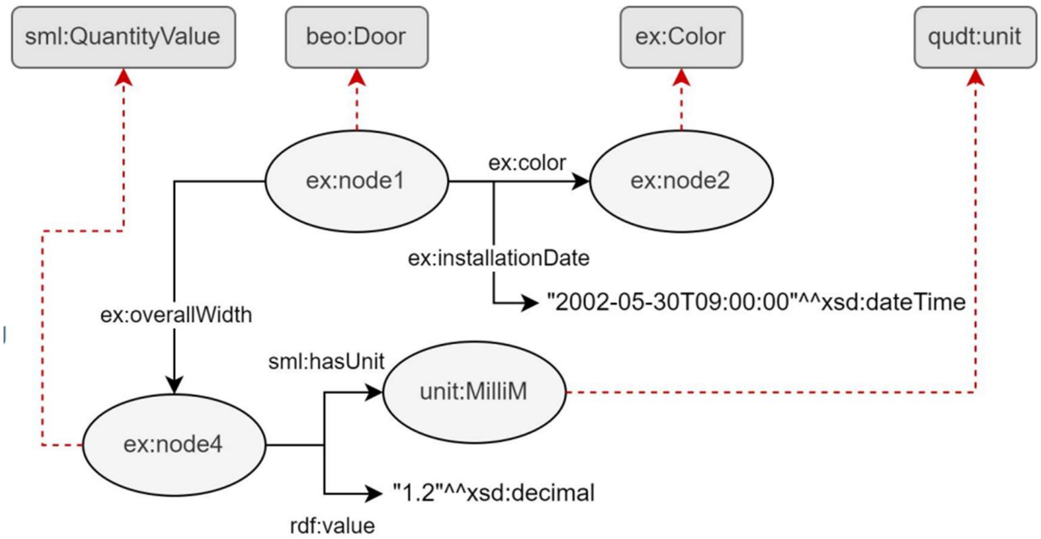

Figure 3.7 Example of qualitative (level 1) and quantitative (level 2) properties in compliance with EN 17632 (Bonduel & Wagner, 2023)

Qualitative properties of the OTL were therefore modelled using the level 1 pattern, while the quantitative properties were modelled using the level 2 pattern as shown in figures 3.9 and 3.10, respectively.

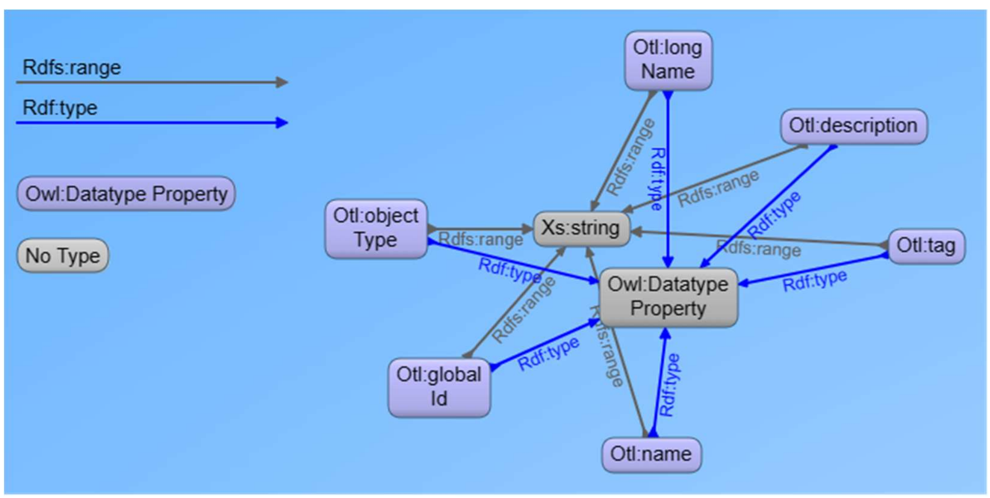

Figure 3.8 TBox of of quantitative IFC attributes (level 1 property modelling) in the newly created OTL

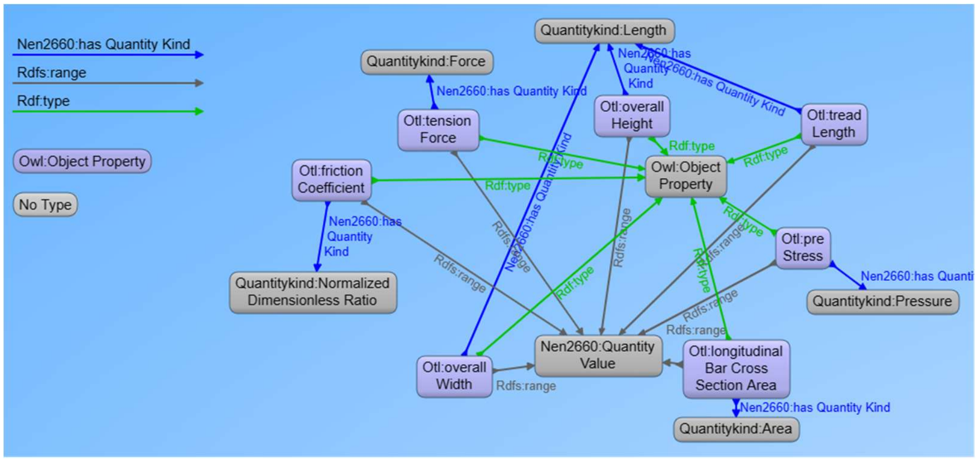

Figure 3.9 Tbox of qualitative IFC attributes (level 2 property modelling) in the newly created object type library

An additional issue that needs to be resolved is the attribute inheritance of the IFC Schema. IFC classes inherit attributes from their superclasses. Considering that this research deals only with a subset of the schema, there is a possibility that some attributes higher up the schema will not be inherited. To prevent this, a query needs to be made to find all the IFC classes between IfcRoot as a starting point and the classes that are the starting point of this research (stated in chapter 3.1.), find all the attributes that these classes have and finally make sure that the top level resources in the OTL contain these attributes. Such process resulted in creation of five owl:DatatypeProperty resources (otl:globalId, otl:name, otl:description, otl:objectType, otl:tag) that were connected with the highest OTL entities (otl:BuildingElement, otl:DistributionElement, otl:FurnishingElement, otl:ElementComponent, otl:OpeningElement, otl:SpatialElement, otl:ElementAssembly)

Some additional adjustments also need to be made. Firstly, considering that type enumerations were created as separate classes in the OTL (as shown in chapter 3.2.1), creating attributes that describe enumerations would be duplication. Therefore, all the attributed that contain predefinedType will be filtered out. Additionally, several attributes existing in the ifcOWL are actually subclasses of the IfcRelationship and are therefore not attributes as imagined in the OTL that should contain alphanumeric values. Instead, these attributes contain pointers to other IFC entities. This could be simply due to the current STEP-centricity of the IFC Schema (discussed in 2.3.1) or some other reason out of the scope of this research. These attributes that are subclasses of the IfcRelationship entity will be omitted from the scope of this research as well.

Considering that the IFC Schema is a closed system, it defines its own measure types as well, while EN 17632 prescribes expressing measurements with the QUDT ontology concepts. Therefore, a mapping needs to be made. The attributes from the IFC are given adequate quantity kind via hasQuantityKind relation from the EN 17632 as shown in the Table 3.1.

Table 3.1 Replacing concepts from the IFC with adequate concepts from quantitykind ontology

IFC measure type	Replacing concept
IfcPositiveLengthMeasure	quantitykind: Length
IfcAreaMeasure	quantitykind: Area
IfcForceMeasure	quantitykind: Force
IfcNormalizedRationMeasure	quantitykind: NormalizedDimensionlessRatio
IfcPressureMeasure	quantitykind: Pressure
IfcIdentifier	xsd:string (level 1 property modelling)
IfcLabel	xsd:string (level 1 property modelling)
IfcGloballyUniqueID	xsd:string (level 1 property modelling)

3.4. Prescribing constraints in the OTL

It was stated in chapter 3.1. that the workflow consists of three parts. The structure (expressed in the OTL), instantiation of actual project data (expressed in dataset) and constraints (expressed in the shapes graph). However, some changes are inherent to the IFC data model, and are therefore inherent to the IFC-based OTL. In practice, that means that these constraints will always be present, and can therefore be included in the OTL itself, instead of including it in a separate shapes graph.

Constraints consist of two important pieces. Firstly, the sh:PropertyShape expresses which property should be checked and for which constraints. In example, a simple property shape validates the existence of the globalId attribute in the IFC. The globalIdShape expresses following information. The property shape should find triples where globalId is the predicate (expressed by the sh:path), and will check that there is no more and no less that one object (making this attribute both required and unique for each object) and additionally, check that the datatype of the object is a string (using the xsd ontology, in compliance with the EN 17632 standard).

```
TERSE TRIPLE LANGUAGE (TTL)

otl:globalIdShape a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path otl:globalId ;
    sh:minCount 1 ;
    sh:maxCount 1 .
```

Figure 3.10 An example of a property shape node

Property shapes are reusable, i.e. the same property node can target multiple classes. Considering the issue of inheritance from IfcRoot (as discussed in chapter 3.3.3), the globalIdShape will be assigned to all the top-level OTL entities. This is done through an intermediary concept, a NodeShape that is then assigned multiple properties, as seen in an example in Fig 3.11.

```
TERSE TRIPLE LANGUAGE (TTL)

otl:BuildingElementShape a sh:NodeShape ;
    sh:property otl:globalIdShape, otl:nameShape,
    otl:descriptionShape, otl:objectTypeShape,
    otl:tagShape ;
    sh:targetClass otl:BuildingElement .
```

Figure 3.11 An example of a node shape node

A property shape usable for level 2 property modelling is more complex, i.e. property shape consists of two nested property shapes, expressing the existence of a unit node (via nen2660:hasUnit path) and the existence of a decimal value, as shown in Fig. 3.12.

```
TERSE TRIPLE LANGUAGE (TTL)

otl:overallWidth-Window a sh:PropertyShape ;
    sh:path otl:overallWidth ;
    sh:minCount 0 ;
    sh:maxCount 1 l
    sh:property otl:overallWidth-Window-unit,
                otl:overallWidth-Window-value.
;

otl:overallWidth-Window-unit a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path nen2660:hasUnit;
    sh:minCount 1 ;
    sh:maxCount 1 .

otl:overallWidth-Window-value a sh:PropertyShape ;
    sh:datatype xsd:integer ;
    sh:path rdf:value;
    sh:minCount 1 ;
    sh:maxCount 1 .
```

Figure 3.12 An example of a property shape node for level 2 property modelling

4. PROOF OF CONCEPT

4.1. Object Type Library Development

4.1.1. Creating the taxonomy in Object Type Library

As shown in Fig. 4.1, the taxonomy in the OTL is created by loading the ifcOWL to a triplestore, querying it for enumerations and creating these enumerations as individual resources of owl:Class type in the OTL.

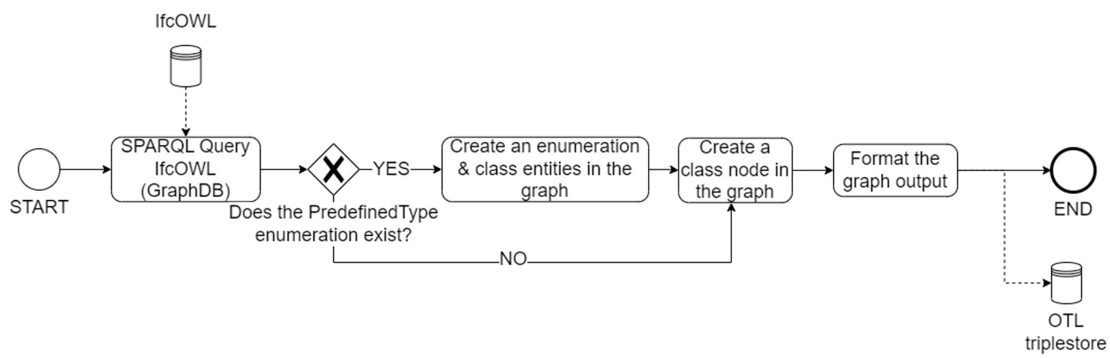

Figure 4.1 Process of establishing the taxonomy in the OTL

Finally, the top level IFC entity is mapped to the EN 17632 concept according to the mapping table shown in Table 4.1.

Table 4.1 Mapping table - IFC entities to EN 17632 concepts

IFC entity	EN 17632 concept
IfcBuildingElement	DiscreteObject
IfcDistributionElement	DiscreteObject
IfcFurnishingElement	DiscreteObject
IfcElementComponent	DiscreteObject
IfcOpeningElement	PhysicalObject
IfcSpatialElement	SpatialRegion
IfcElementAssembly	DiscreteObject

The IFC schema (in the ifcOWL format) then needs to be loaded to a triplestore to enable querying. Several triplestore solutions were investigated and the final version of the workflow opted for Ontotext

GraphDB because of the free version and simplicity of use. For visualization purposes, AllegroGraph and its Gruff tool was utilized, as well as Stardog with its Studio and Explorer modules. After loading the IFC schema to the triplestore, it was queried with the SPARQL query shown in Fig. 4.2.

```
SPARQL

 PREFIX nen2660: <https://w3id.org/nen2660/def#>
  PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD2#>
  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  PREFIX expr: <https://w3id.org/express#>
  PREFIX zh: <https://w3id.org/ziekenhuis/def#>
  PREFIX owl: <http://www.w3.org/2002/07/owl#>
  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
  SELECT DISTINCT ?subObject ?directParent ?enum ?type
    WHERE {
         ?subObject rdfs:subClassOf* ifc:IfcRoof .
         ?subObject rdfs:subClassOf ?directParent .
        ?directParent a ?type
      FILTER ( ?type != owl:Restriction)
        FILTER( !regex(str(?directParent), "node", "i") )
         OPTIONAL {
            ?predefinedTypeRelation rdfs:domain ?subObject ;
                                            rdfs:range
?subObjectEnumType .
            ?enum rdf:type ?subObjectEnumType .
            ?subObjectEnumType rdfs:subClassOf expr:ENUMERATION .

            FILTER(?enum != ifc:NOTDEFINED)
            FILTER(?enum != ifc:USERDEFINED)
         }
     }
```

Figure 4.2 SPARQL query used in creation of the Object Type Library

This query, as shown in Fig. 4.3 returns all PredefinedType enumerations, a top-level IFC class (e.g. IfcBuildingElement) which it stems from, as well as the direct IFC parent class (e.g. IfcRoof),. Aditionally, the type of relationship was used to filter for owl:Restriction, to prevent blank nodes to return in the query.

47	ifc:IfcShadingDevice	ifc:IfcBuildingElement	ifc:AWNING
48	ifc:IfcCurtainWall	ifc:IfcBuildingElement	
49	ifc:IfcPlate	ifc:IfcBuildingElement	ifc:CURTAIN_PANEL
50	ifc:IfcPlate	ifc:IfcBuildingElement	ifc:SHEET
51	ifc:IfcFooting	ifc:IfcBuildingElement	ifc:PAD_FOOTING

Figure 4.3 Values returned by the SPARQL query

The tool used for the implementation, GraphDB, allow querying directly in the user interface, but also allows returns in the form of Javascript Object Notation (JSON) objects. This enables direct use of SPARQL querying in the script that will be developed. The returned values (e.g. from Fig. 4.3) can then be used in the logic of the script to implement logic and create resources in the new OTL. The implementation is done with the JavaScript programming language, more specifically, in Node.js runtime environment and with utilization of several external libraries, listed in Appendix 1. The output of this part of the workflow is part of the newly created OTL, shown in Fig. 4.4. It contains all the IFC enumerations, their direct parents as well as top level IFC objects, with connections as illustrated in Fig.

3.5 These classes are connected to the ifcOWL schema by the rdfs:seeAlso relationship. The process was done for IFC4 Add2 TC1, resulting in around 3,500 triples. A snippet showing a part of the OTL is visible in the Fig. 4.4 below.

TERSE TRIPLE LANGUAGE (TTL)

```
otl:BuildingElement a owl:Class ;
    rdfs:subClassOf nen2660:DiscreteObject ;
    rdfs:seeAlso ifc:IfcBuildingElement ;
    skos:prefLabel "Building Element"@en .

otl:Covering a owl:Class ;
    rdfs:subClassOf otl:BuildingElement ;
    rdfs:seeAlso ifc:IfcCovering ;
    skos:prefLabel "Covering"@en .

otl:Covering-INSULATION a owl:Class ;
    rdfs:subClassOf otl:Covering ;
    rdfs:seeAlso ifc:INSULATION ;
    skos:prefLabel "Covering INSULATION"@en .

otl:SpatialElement a owl:Class ;
    rdfs:subClassOf nen2660:SpatialRegion ;
    rdfs:seeAlso ifc:IfcSpatialElement ;
    skos:prefLabel "Spatial Element"@en .

otl:Space a owl:Class ;
    rdfs:subClassOf otl:SpatialElement ;
    rdfs:seeAlso ifc:IfcSpace ;
    skos:prefLabel "Space"@en .

otl:Space-INTERNAL a owl:Class ;
    rdfs:subClassOf otl:Space ;
    rdfs:seeAlso ifc:INTERNAL ;
    skos:prefLabel "Space INTERNAL"@en .
```

Figure 4.4 A snippet from the Object Type Library

4.1.2. IFC Attributes

After establishing the taxonomy of the object type library, the next step is to develop a script that would analyse the ifcOWL and the attributes of each of the classes, considering important concepts of the IFC data model explained in chapter 3.3.

The first issue that needs to be resolved is the attribute inheritance. This will be done querying the schema as shown in Fig. 4.5. results in a list of all the attributes between IfcRoot and IfcBuildingElement filtered for attributes that are subclasses of IfcRelationship. These entities are listed in Fig. 4.6.

```
SPARQL

  SELECT DISTINCT ?subObject ?attribute ?upAtt
    WHERE {
         ?subObject rdfs:subClassOf* ifc:IfcRoot .
         ?attribute rdfs:domain ?subObject .
        ?attribute rdfs:range ?upAtt .
        ?upAtt rdfs:subClassOf ?upperClass .

    FILTER( !regex(str(?upperClass), "node", "i") )
    FILTER NOT EXISTS {
         ?upAtt rdfs:subClassOf*  ifc:IfcRelationship.
    }
    FILTER EXISTS {
         ifc:IfcBuildingElement rdfs:subClassOf*  ?subObject.
    }
    }
```

Figure 4.5 Querying for attributes of classes between IfcRoot and IfcBuildingElement

ifc:IfcRoot	ifc:globalId_IfcRoot
ifc:IfcRoot	ifc:name_IfcRoot
ifc:IfcRoot	ifc:description_IfcRoot
ifc:IfcObject	ifc:objectType_IfcObject
ifc:IfcElement	ifc:tag_IfcElement

Figure 4.6 All the attributes between IfcRoot and IfcBuildingElement and their classes

All the attributes found between IfcRoot and IfcBuildingElement are modelled according to Level 1 modelling pattern and appended to the highest class in the OTL, otl:BuildingElement. This is done by creating sh:PropertyShape resources for each individual attribute and assigning it to the adequate OTL classes via sh:NodeShape, as shown in Fig. 4.7. The same process will be repeated with the other considered top-level IFC classes (listed in chapter 3.1 and in the Fig. 3.5)

TERSE TRIPLE LANGUAGE (TTL)

```
otl:globalIdShape a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path otl:globalId ;
    sh:minCount 1 ;
    sh:maxCount 1 .

otl:nameShape a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path otl:name ;
    sh:minCount 0 ;
    sh:maxCount 1 .

otl:descriptionShape a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path otl:description ;
    sh:minCount 0 ;
    sh:maxCount 1 .

otl:objectTypeShape a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path otl:objectType ;
    sh:minCount 0 ;
    sh:maxCount 1 .

otl:tagShape a sh:PropertyShape ;
    sh:datatype xsd:string ;
    sh:path otl:tag ;
    sh:minCount 0 ;
    sh:maxCount 1 .

otl:BuildingElementShape a sh:NodeShape ;
    sh:property otl:globalIdShape,
                otl:nameShape,
                otl:descriptionShape,
                otl:objectTypeShape,
                otl:tagShape ;
    sh:targetClass otl:BuildingElement .
```

Figure 4.7 Attributes of the newly created OTL

Similarly, attributes of individual classes are found using the SPARQL Query shown in Fig. 4.8 and appended to their classes. PredefinedType attributes are ignored considering that the enumerations were modelled as classes in chapter 4.1.1.

SPARQL

```
SELECT DISTINCT ?subObject ?attribute ?upAtt
 WHERE {
      ?subObject rdfs:subClassOf* ifc:IfcElement .
      ?subObject rdfs:subClassOf ?directParent .
      ?directParent a ?type
   FILTER ( ?type != owl:Restriction)
     FILTER( !regex(str(?directParent), "node", "i") )
      OPTIONAL {
         ?predefinedTypeRelation rdfs:domain ?subObject ;
                          rdfs:range ?subObjectEnumType .
         ?enum rdf:type ?subObjectEnumType .
         ?subObjectEnumType rdfs:subClassOf expr:ENUMERATION

 }
     ?attribute rdfs:domain ?subObject .
     ?attribute rdfs:range ?upAtt .
     ?upAtt rdfs:subClassOf ?upperClass .


     FILTER( !regex(str(?upperClass), "node", "i") )
     FILTER NOT EXISTS {
      ?upAtt rdfs:subClassOf*  ifc:IfcRelationship.
     }
 FILTER (!regex(str(?attribute), "predefinedType", "i"))
 }
```

Figure 4.8 Querying for attributes of classes below IfcElement

These attributes are modelled according to level 2 modelling pattern. Therefore, each sh:PropertyShape contains nested property shapes that prescribe existence of a value and unit, as shown in the example in Fig. 4.9.

TERSE TRIPLE LANGUAGE (TTL)

```
otl:WindowShape a sh:NodeShape ;
    sh:property otl:overallWidth-Window,
                otl:overallHeight-Window,
                otl:partitioningType-Window ;
    sh:targetClass otl:Window .

otl:overallHeight-Window a sh:PropertyShape ;
    sh:path otl:overallHeight ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:property otl:overallHeight-Window-unit,
                otl:overallHeight-Window-value .

otl:overallHeight-Window-unit a sh:PropertyShape ;
    sh:path nen2660:hasUnit ;
    sh:minCount 1 ;
    sh:maxCount 1 .

otl:overallHeight-Window-value a sh:PropertyShape ;
    sh:datatype xsd:decimal ;
    sh:path rdf:value ;
    sh:minCount 1 ;
    sh:maxCount 1 .
```

Figure 4.9 SHACL constraints by level 2 property modelling

As mentioned, PredefinedType enumerations are not expressed as class properties but as individual clasess, but there are additional enumeration attributes that are modelled differently. An example (partitioningType attribute of IfcWindow class) is shown in Fig. 4.10 and Fig. 4.11.

```
TERSE TRIPLE LANGUAGE (TTL)

otl:partitioningType a owl:ObjectProperty ;
    rdfs:range otl:WindowTypePartitioningEnum ;
    rdfs:domain otl:Window .

otl:partitioningType-Window a sh:PropertyShape ;
    sh:path otl:partitioningType ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:class otl:WindowTypePartitioningEnum .

otl:WindowTypePartitioningEnum a owl:Class, sh:NodeShape ;
    rdfs:subClassOf nen2660:EnumerationType ;
    skos:definition "This is an enumeration list of the partitioningType attribute of the
otl:Window class"@en ;
    sh:in otl:WindowTypePartitioningEnum-list-0 .

otl:WindowTypePartitioningEnum-list-0 a rdf:List ;
    rdf:first otl:NOTDEFINED-Window ;
    <https://www.w3.org/1999/02/22-rdf-syntax-ns#rest> otl:WindowTypePartitioningEnum-list-1 .

    otl:NOTDEFINED-Window a otl:partitioningType-Window ;
    skos:prefLabel "NOTDEFINED"@en ;
    skos:definition "This is an enumeration of the partitioningType attribute of the otl:Window
class"@en .

otl:WindowTypePartitioningEnum-list-1 a rdf:List ;
    rdf:first otl:USERDEFINED-Window ;
    <https://www.w3.org/1999/02/22-rdf-syntax-ns#rest> otl:WindowTypePartitioningEnum-list-2 .

otl:USERDEFINED-Window a otl:partitioningType-Window ;
    skos:prefLabel "USERDEFINED"@en ;
    skos:definition "This is an enumeration of the partitioningType attribute of the otl:Window
class"@en .
```

Figure 4.10 Code snippet of the partitioningType enumeration attribute representation in linked data form

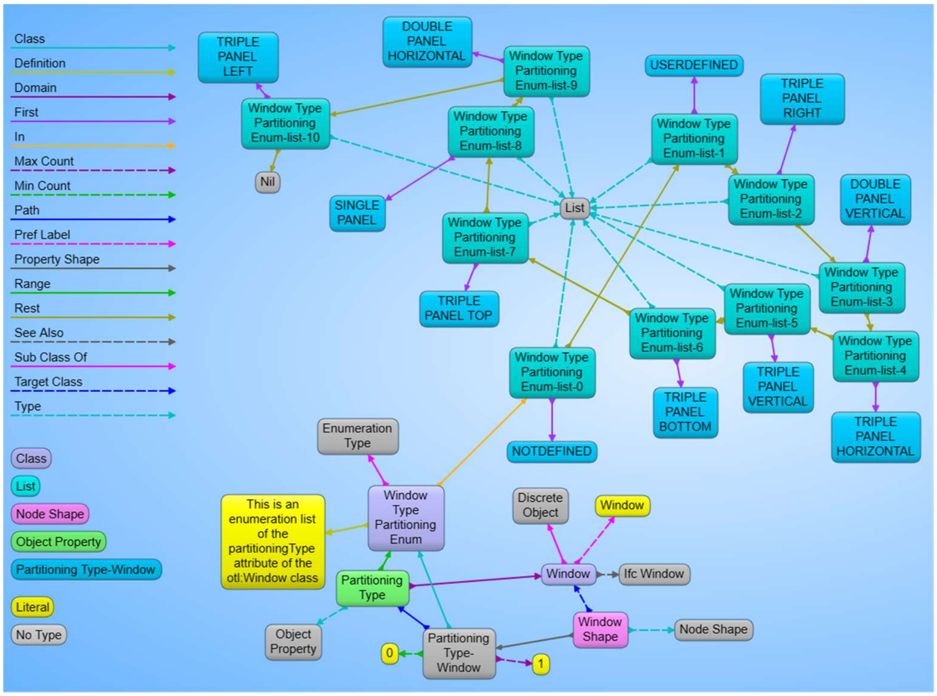

Figure 4.11 Visualization of the partitioningType enumeration attribute

The final OTL consists of more that 3500 nodes, as visualized in Fig. 4.12.

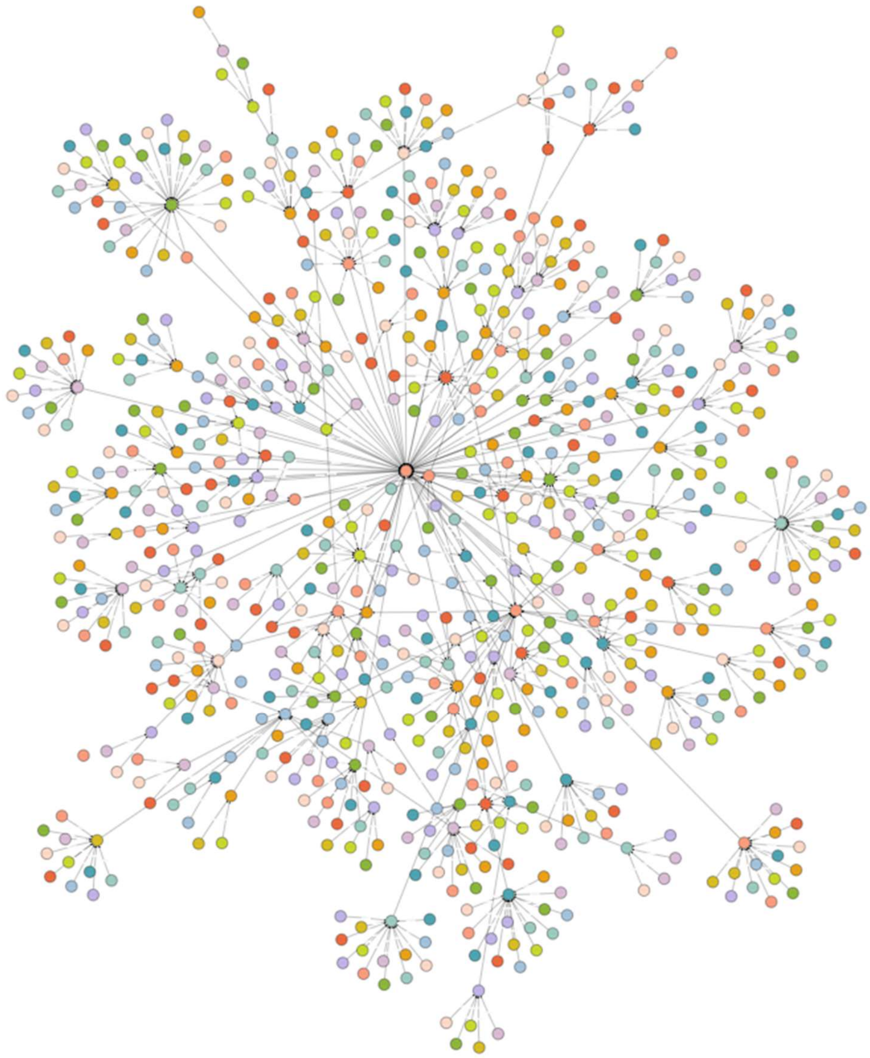

Figure 4.12 Visualization of the OTL in Stardog Explorer

4.2. Parsing the IFC-STEP file

After the OTL taxonomy was established, the next step was to develop a script that could take an IFC-STEP file, process it and produce an RDF graph (in JSON-LD and TTL formats) that complies with the OTL. The implementation is done in JavaScript with the utilization of web-ifc package developed as a part of the IFC.js project along with some additional packages, listed and described in Annex 1. The test data comes from the IFC models developed in the BIM A+2 – Modelling in Architecture and Engineering module. Three models are analysed, two of which are architectural models and an additional mechanical model. These models are deemed appropriate in terms of complexity and considering that special care was put into proper IFC Export of the entities and properties from the authoring tool (Revit). Some issues regarding IFC schema as well as the export from the authoring tool exist and will be briefly discussed later in the research.

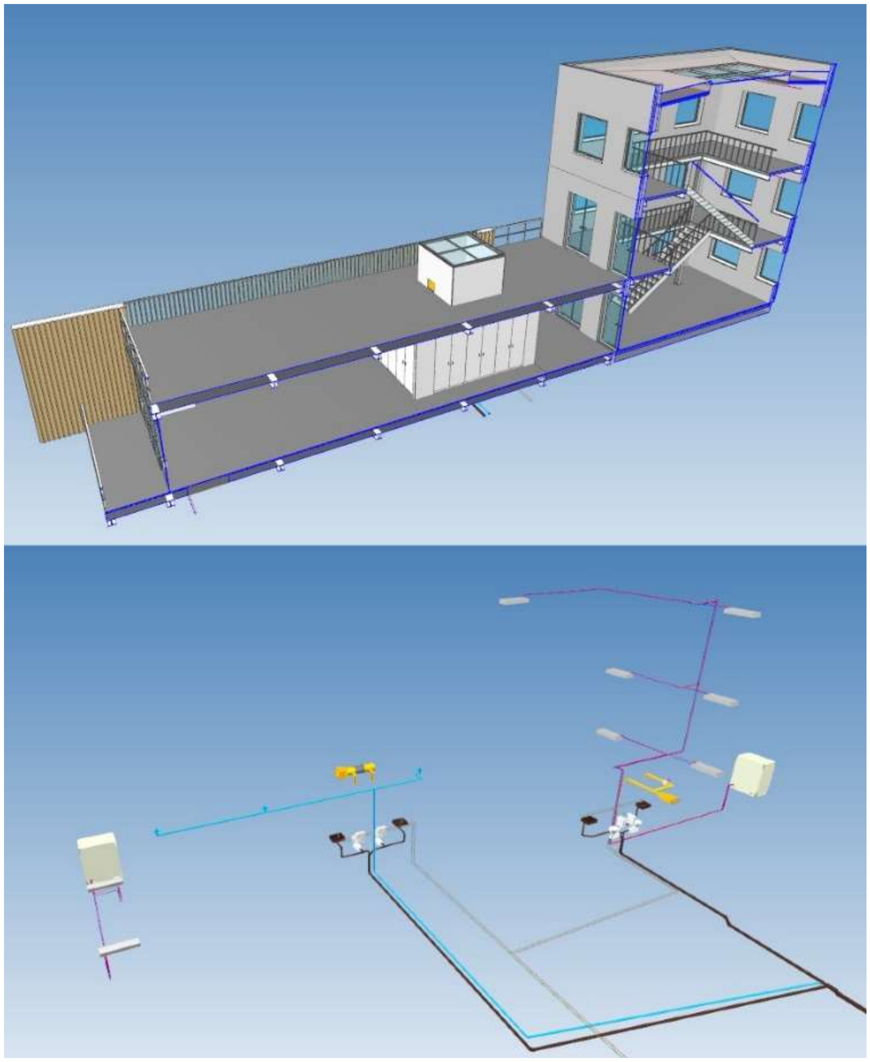

Figure 4.13 IFC-STEP models used, architectural (above) and mechanical (below)

4.2.1. Taxonomy

BPMN representation of the OTL taxonomy creation is shown on the Fig. 4.14. The web-ifc API enables retrieving all the elements depending on their IfcClass or the IfcRelationship. Therefore, the first step of the IFC-STEP parsing is to retrieve all the entities in the dataset, their classes and enumerations (if available) and utilize this information to create nodes in the RDF graph where the GUID serves as a unique identifier of the particular element, and link it via rdf:type relation to the classes existing in the OTL.

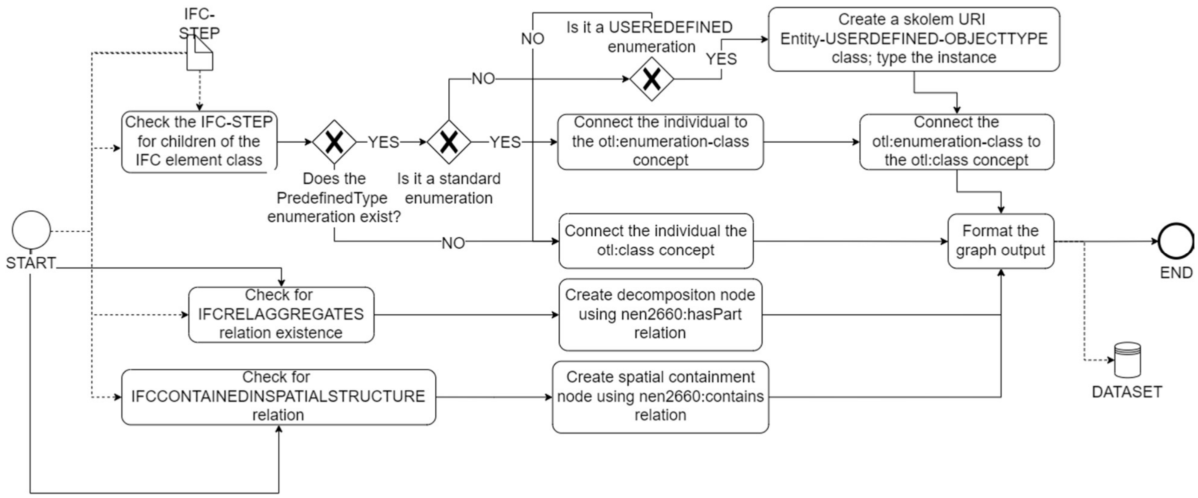

Figure 4.14 BPMN diagram of IFC-STEP taxonomy parsing workflow

A snippet of the output is presented in the Fig 4.15. below.

```
dis:2U2bEFdi9D29Y9qmkfORiT a otl:PipeFitting-BEND .

dis:29JEkkq6v4deitAoBmBEWS a otl:AirTerminal-GRILLE .

dis:29JEkkq6v4deitAoBmBEFu a otl:WasteTerminal-ROOFDRAIN .

dis:29JEkkq6v4deitAoBmBDmU a otl:DuctSegment-RIGIDSEGMENT .
```

Figure 4.15 A snippet from the dataset - connection of instances to OTL types

If an element is typed with the USERDEFINED value, a TBox node is created in the dataset, that will type the newly found object type and connect it to the higher class, as shown in Fig 4.16.

```
TERSE TRIPLE LANGUAGE (TTL)

dis:2gvPjy0ef0Le27qKCMgxYv a dis:CurtainWall-USERDEFINED-STEEL_COPING .

dis:CurtainWall-USERDEFINED-STEEL_COPING rdfs:subClassOf otl:CurtainWall .
```

Figure 4.16 A snippet from the dataset - creation userdefined object enumerations

Additionally, besides the connection of the dataset to the OTL, spatial decomposition was taken into account by mapping the RelatingObject and RelatedObjects attributes from the IfcRelAggregates entity as the subject and object of a triple, connected by the nen2660:hasPart relation. In the same way, spatial containment was taken into account by connecting the RelatingStructure and RelatedElements of the IfcRelContainedInSpatialStructure entity with the nen2660:contains relation. The snippet below shows the part of the dataset that shows how project can decompose in multiple or, as in our case, just one site. In the same way, site decomposes in one or more buildings, and each building has one or multiple building storeys. Building storey contain spaces but also individual elements, whereas spaces can only contain individual elements.

TERSE TRIPLE LANGUAGE (TTL)

```
<https://otl.buildingsmart.org/test-project/2BuAc$sGjBxPgdrqYVruQ0> a otl:Project ;
    nen2660:hasPart <https://otl.buildingsmart.org/test-project/2BuAc$sGjBxPgdrqYVruQ2> .

<https://otl.buildingsmart.org/test-project/2BuAc$sGjBxPgdrqYVruQ2> a otl:Site ;
    nen2660:hasPart <https://otl.buildingsmart.org/test-project/2BuAc$sGjBxPgdrqYVruQ1> .

<https://otl.buildingsmart.org/test-project/2BuAc$sGjBxPgdrqYVruQ1> a otl:Building ;
    nen2660:hasPart dis:3Zu5Bv0LOHrPC10026FoQQ, dis:15Z0v90RiHrPC20026FoKR,
<https://otl.buildingsmart.org/test-project/1VO3iV$$96$hFKjId3ToVq>,
<https://otl.buildingsmart.org/test-project/1VO3iV$$96$hFKjId3ToVt> .

dis:3Zu5Bv0LOHrPC10026FoQQ a otl:BuildingStorey ;
    nen2660:contains  dis:0Zbcw4zuTA5wHLhuC4oKO2, dis:0Zbcw4zuTA5wHLhuC4oKOT,
dis:1eqg1glbL1rABPeoupj2EA .

dis:1eqg1glbL1rABPeoupj2Ea a otl:Space-SPACE ;
    nen2660:contains dis:2U2bEFdi9D29Y9qmkfOT_v, dis:2U2bEFdi9D29Y9qmkfOTyZ .

dis:2U2bEFdi9D29Y9qmkfOTyZ a otl:PipeFitting-JUNCTION .
```

Figure 4.17 A snippet from dataset - showing decomposition and containment of spatial elements

4.2.2. Class parsing validation

To validate the output of the parsing script, the test files were additionally parsed with the IFCtoLBD converter and loaded into new GraphB repositories. Then the simple SPARQL query shown in Fig. 3.13 was used to count the number of distinct triples in each of the repositories.

SPARQL

```
  SELECT (COUNT (DISTINCT ?subject) AS ?COUNT)
  WHERE {
  ?s ?p ?o
}
```

Figure 4.18 SPARQL query to count the number of entities

The table 4.2 below shows the comparison between the parsing done by the IFCtoLBD and the parser developed in the thesis. As visible, the datasets from newly developed parsers always have one additional node. This node is the node of the IfcProject that does not get parsed in the IFCtoLBD converter since it is not a subclass of the IfcElement but was manually included in the newly developed script, as well as the OTL. Whether all the models should have different GUIDs and therefore different IfcProject entities, as it is the case now, or should they somehow point to the same IfcProject entity or some higher-level common entity should be discussed but is out of the scope of this research.

Table 4.2 Parsing results per model

File name	Number of entities in the LBD dataset	Number of entities in the thesis dataset
Arch1	2740	2741
Arch2	251	252
Mech	492	493

4.2.3. Attribute parsing

Flow diagram of the IFC attribute parsing is shown in Figure 4.19. Attributes are placed in separate array objects depending on their modelling patterns. During parsing the IFC file with the ifc.js library, each attribute is analyzed depending on where it should be placed.

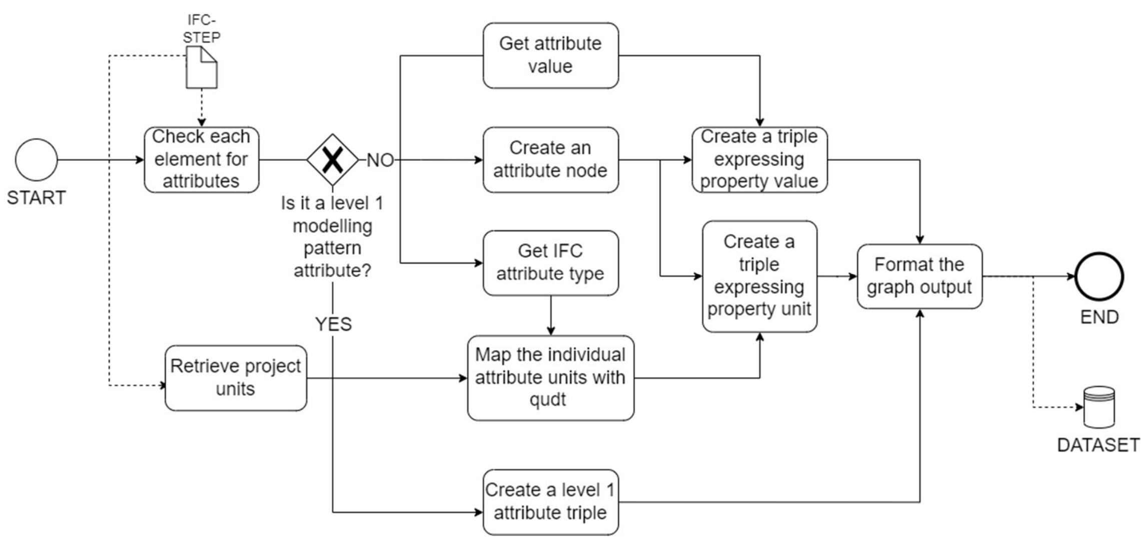

Figure 4.19 Flow diagram of IFC attribute parsing process

Attributes modelled according to level 1 modelling pattern maintain direct link with the IFC element, as shown in Fig. 4.20.

TERSE TRIPLE LANGUAGE (TTL)

```
dis:2vFKzyJSbEp8zuPoUlGXjO otl:globalId "2vFKzyJSbEp8zuPoUlGXjO" ;
    otl:name "Railing:BIMA+_Railing_Handrail_SteelRound25mm:337604" ;
    otl:objectType "Railing:BIMA+_Railing_Handrail_SteelRound25mm" ;
    otl:tag "337604" ;
    a otl:Railing-HANDRAIL .
```

Figure 4.20 TTL snippet from the dataset showing level 1 attributes for an IFC element

Attributes modelled according to the level 2 modelling pattern have an intermediary node and express the unit and the value separately. Units are retrieved from the project level and matched with the appropriate qudt units, as shown in Fig. 4.21.

TERSE TRIPLE LANGUAGE (TTL)

```
dis:1qHFr7JC9CTO5IS3NzkT0p otl:globalId "1qHFr7JC9CTO5IS3NzkT0p" ;
    otl:name "Doors_ExtDbl_Flush:BIMA+_Door_ExteriorGlassDoor_1510x2310mm:317917" ;
    otl:objectType "Doors_ExtDbl_Flush:BIMA+_Door_ExteriorGlassDoor_1510x2310mm" ;
    otl:tag "317917" ;
    otl:overallHeight dis:1qHFr7JC9CTO5IS3NzkT0p-overallHeight ;
    otl:overallWidth dis:1qHFr7JC9CTO5IS3NzkT0p-overallWidth ;
    otl:operationType otl:DOUBLE_DOOR_SINGLE_SWING-Door ;
    a otl:Door-DOOR .

dis:1qHFr7JC9CTO5IS3NzkT0p-overallHeight rdf:value 2310 ;
    nen2660:hasUnit <http://qudt.org/schema/qudt/MiliM> .

dis:1qHFr7JC9CTO5IS3NzkT0p-overallWidth rdf:value 1510 ;
    nen2660:hasUnit <http://qudt.org/schema/qudt/MiliM> .
```

Figure 4.21 TTL snippet from the dataset showing level 2 attributes for an IFC element

4.3. Dataset checking with SHACL

Finally, to ensure validity of the proposed framework as well as the data output of the proof-of-concept application, it is necessary to check the dataset with SHACL. The dataset consists of the TBOx part (contained in the OTL) and ABox instantiation created by parsing the IFC file, and SHACL validation will be done on two levels:

- The first level is checking the dataset for compliance with SHACL shapes that exist in the OTL. These SHACL shapes express the inherent nature of the IFC schema (such as that every subclass of the IfcRoot class should have a global unique identifier) and are therefore universally applied regardless of the project
- The second level is checking for project specific constraints contained in a separate shapes graph. These shapes differ from project to project or even depending on the discipline being checked

Both checks will firstly be done in Stardog Studio due to the ease of use, to check for validity of SHACL shapes. Afterwards, the same procedure will be carried out after development of a JavaScript script that enables SHACL validation, and the results will be compared.

4.3.1. OTL level compliance checking

Checking OTL Shapes in Stardog Studio returns errors, as shown in Fig. 4.22.

TERSE TRIPLE LANGUAGE (TTL)

```
_:r27 rdf:type sh:ValidationReport ;
    sh:conforms "false"^^xsd:boolean ;
    sh:result _:r0, _:r1, _:r10, _:r100, _:r101 .

_:r28 rdf:type sh:ValidationResult ;
    sh:focusNode <https://otl.buildingsmart.org/test-
project/0p2ZDGdpD7UPt8bD8w3L$P-overallHeight> ;
    sh:resultMessage "Value must have datatype xsd:decimal" ;
    sh:resultPath rdf:value ;
    sh:resultSeverity sh:Violation ;
    sh:sourceConstraintComponent sh:DatatypeConstraintComponent ;
    sh:sourceShape otl:overallHeight-Window-value ;
    sh:value "0"^^xsd:integer .

_:r93 rdf:type sh:ValidationResult ;
    sh:focusNode dis:3Zu5Bv0LOHrPC10026FoQQ ;
    sh:resultMessage "There must be >= 1 values" ;
    sh:resultPath otl:globalId ;
    sh:resultSeverity sh:Violation ;
    sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
    sh:sourceShape otl:globalIdShape .
```

Figure 4.22 Validation report of OTL shapes from Stardog Studio

This validation report shows two errors:

- Datatype of the overallHeight -Window-value is an integer, whereas it is prescribed in the OTL that it needs to be a decimal
- Each element needs to have a globalId property, whereas this specific element (an IFC space identified by the GUID, dis:3Zu5Bv0LOHrPC10026FoQQ) does not have that attribute expressed

Enumeration attributes return errors if their enumeration is not one of the predefined in the IFC schema, as shown in Fig. 4.23.

TERSE TRIPLE LANGUAGE (TTL)

```
_:r192 rdf:type sh:ValidationResult ;
   sh:focusNode <https://otl.buildingsmart.org/test-project/1qHFr7JC9CTO5IS3NzkT0p> ;
   sh:resultMessage "Value must have type
<https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/DoorTypeOperationEnum>" ;
   sh:resultPath <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/operationType> ;
   sh:resultSeverity sh:Violation ;
   sh:sourceConstraintComponent sh:ClassConstraintComponent ;
   sh:sourceShape <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/operationType-Door> ;
   sh:value <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/DOORERROR-Door> .

_:r121 rdf:type sh:ValidationResult ;
   sh:focusNode <https://otl.buildingsmart.org/test-project/1EFBNAS5XByhJ5rD1UPCsc> ;
   sh:resultMessage "Value must have type
<https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/WindowTypePartitioningEnum>" ;
   sh:resultPath <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/partitioningType> ;
   sh:resultSeverity sh:Violation ;
   sh:sourceConstraintComponent sh:ClassConstraintComponent ;
   sh:sourceShape <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/partitioningType-Window> ;
   sh:value <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/WINDOWERROR-Window> .
```

Figure 4.23 SHACL validation of enumeration attributes

And finally, level 2 attribute constraint do not prescribe attributes as required. If these attributes do exist however, they prescribe that each should have a unit and a value, along with the requirement that the value should be of certain datatype (seen in Fig. 4.22). In case that an attribute does not satisfy these requirements, SHACL validation returns messages as seen in Fig. 4.24 below.

TERSE TRIPLE LANGUAGE (TTL)

```
_:r213 rdf:type sh:ValidationResult ;
   sh:focusNode <https://otl.buildingsmart.org/test-project/3gD7tao8P3IujTtErDpEGD-
overallHeight> ;
   sh:resultMessage "There must be >= 1 values" ;
   sh:resultPath rdf:value ;
   sh:resultSeverity sh:Violation ;
   sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
   sh:sourceShape <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/overallHeight-Window-
value> .

_:r214 rdf:type sh:ValidationResult ;
   sh:focusNode <https://otl.buildingsmart.org/test-project/3gD7tao8P3IujTtErDpEU0-
overallHeight> ;
   sh:resultMessage "There must be >= 1 values" ;
   sh:resultPath <https://w3id.org/nen2660/def#hasUnit> ;
   sh:resultSeverity sh:Violation ;
   sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
   sh:sourceShape <https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/overallHeight-Window-unit>
```

Figure 4.24 SHACL validation of attribute value and unit existence

It is important to mention that OTL level SHACL shapes are not restrictive regarding the actual unit of the attribute to be used (e.g. meters, centimeters etc.), but only regarding the actual existence of the unit. This was intentionally done to ensure that OTL is applicable to datasets from multiple disciplines, using different project units. In case that units need to be validated for the content and not only for the existence, this can be done on the project level in a separate shapes graph.

4.4. Project level compliance checking

Project specific constraints can include anything that is important for the authority to assess the compliance of the project submitted. In example, metadata about the project management and application, as well as data coming from BIM and GIS systems can be merged in one data source and evaluated at the same place. The following examples will focus on simple constraints (value and relational constraints) dealing with the researched topic, the building elements coming from the IFC represented in linked data form. A more thorough investigation of types of SHACL constraints is available in (Nuyts et al., 2022), consulted during the process of SHACL shape creation which is presented below.

Firstly, the values of certain properties can be check for their value, and an issue can be spotted if a value of a property is more or less than the prescribed value. An example of a use case is accessibility constraint prescribing that door height should be at least 2400mm.

TERSE TRIPLE LANGUAGE (TTL)

```
shapes:doorAccessibilityShape rdf:type sh:NodeShape ;
    sh:property _:r0 ;
    sh:targetClass otl:Door .

shapes:valueShape rdf:type sh:PropertyShape ;
    sh:minInclusive "2400"^^xsd:integer ;
    sh:path rdf:value .

_:r0 sh:message "The door height is not at least 2400" ;
    sh:path otl:overallHeight ;
    sh:property shapes:valueShape .
```

Figure 4.25 Value constraints on project level

In case such constraints are not satisfied, the SHACL report lists issues as visible in Fig. 4.26 below.

TERSE TRIPLE LANGUAGE (TTL)

```
_:r10 rdf:type sh:ValidationReport ;
   sh:conforms "false"^^xsd:boolean ;
   sh:result _:r0, _:r1, _:r11, _:r12, _:r13, _:r14, _:r15, _:r2, _:r3, _:r4, _:r5, _:r6,
_:r7, _:r8, _:r9 .

_:r11 rdf:type sh:ValidationResult ;
   sh:focusNode dis:1ULospfXv7690gCF8h4237-overallHeight ;
   sh:resultMessage "Value must be >= 2400" ;
   sh:resultPath rdf:value ;
   sh:resultSeverity sh:Violation ;
   sh:sourceConstraintComponent sh:MinInclusiveConstraintComponent ;
   sh:sourceShape <https://otl.buildingsmart.org/shacl-shapes/valueShape> ;
   sh:value "2110"^^xsd:integer .
```

Figure 4.26 Validation report of value constraint checking

Besides the checking the values for existence and content, datasets can be checked for relations. An example of such a shape would be to check if a building has at least one building storey, as shown in Fig. 4.27.

TERSE TRIPLE LANGUAGE (TTL)

```
shapes:storeyExistenceShape rdf:type sh:NodeShape ;
    sh:property shapes:buildingExistenceShape ;
    sh:targetClass otl:Building .

shapes:buildingExistenceShape rdf:type sh:PropertyShape ;
    sh:path nen2660:hasPart ;
    sh:qualifiedValueShape otl:BuildingStorey;
    sh:qualifiedMinCount 1 .
```

Figure 4.27 Relational constraints on project level

In case of non – compliance, the validation report reports a violation, clearly stating which SHACL constraint was not fulfilled, as shown in Fig. 4.28.

TERSE TRIPLE LANGUAGE (TTL)

```
_:bnode_29a2037c_da08_44b5_acf9_bfb0e39a14c1_3454 a sh:ValidationReport ;
   sh:conforms false .
_:bnode_29a2037c_da08_44b5_acf9_bfb0e39a14c1_3455 a sh:ValidationResult ;
   sh:resultSeverity sh:Violation ;
   sh:sourceShape <https://otl.buildingsmart.org/shacl-shapes/buildingExistenceShape> ;
   sh:sourceConstraintComponent sh:QualifiedMinCountConstraintComponent ;
   sh:focusNode <https://otl.buildingsmart.org/test-project/2BuAc$sGjBxPgdrqYVruQ1> ;
   sh:resultPath <https://w3id.org/nen2660/def#hasPart> ;
   sh:resultMessage "There must be >= 1 values that satisfy the shape
<https://otl.buildingsmart.org/IFC4_ADD2_TC1/def/BuildingStorey>" .
_:bnode_29a2037c_da08_44b5_acf9_bfb0e39a14c1_3454 sh:result
_:bnode_29a2037c_da08_44b5_acf9_bfb0e39a14c1_3455 .
```

Figure 4.28 Violation of a relational constraint

More complex constraints can be made by combining relational and value constraints to check for fulfilment of project requirements. For example, sanitary spaces can be targeted by name and checked for containment of sanitary terminals, such as washbasins or water closets, stairs and ramps can be checked for accessibility by checking their width and slope and many, many more. This, however, necessarily raises the question of how to combine elements from multiple datasets from different disciplines, as is commonly the case in the design process. This question will be briefly touched upon in the next, discussion section.

5. DISCUSSION

The main objective of this work is to research possibilities of connecting standardised data models in the built environment in the context of digital building permitting processes. The state of the art literature overview was carried out with two main goals. Firstly, it introduces concepts necessary for understanding the rest of the research and secondly, establishes several takeaways that help shape the remainder of the research. On the relevancy side, the literature overview confirms that the research is in line with several projects currently under way. On the technology side, literature overview confirms that future deployed solutions should be based on open standards, easy to use and flexible. Having this in mind, the content of the standards (ISO 16739-1:2018 or IFC, and EN 17632-1:2022 or SML) is analysed through the lens of their interconnectivity. The research introduces theoretical means of connecting these standards for taxonomy, relations and properties, along with a proposal of SHACL constraints modelling for compliance checking.

After presenting the theoretical framework, it is applied in practice with utilization of already existing software tools (GraphDB, Stardog Studio, AllegroGraph), JavaScript programming language and several JavaScript libraries, with the most important being zazuko and IFC.js. The applications for OTL creation, model parsing and validation were successfully developed and tested by comparing the results with an existing tool, IFCtoLBD. The research done in applying the theoretical framework could be considered most important output of this work since it resulted in a workflow that enables connecting the two standards, as well as several remarks that will be presented below.

5.1. Remarks from the framework application process

The research done in applying the theoretical framework resulted in several remarks, presented below. These could prove useful in the future development of both standards and tools implementing these standards. These remarks only refer to the version of the IFC schema analysed in this research (IFC 4 TC1 Add2).

5.1.1. Standardization of enumeration naming in the IFC

There are no clear rules for enumeration naming. Some IFC classes, listed in the table below, have enumerations with clearly separated words by underscores (e.g. Window-TRIPLE_PANEL_RIGHT), making it easy to separate the words to create human-readable attributes (such as skos:prefLabel value). However, most enumerations do not follow any particular naming convention (e.g. Covering-SKIRTINGBOARD), making it very hard to separate the words in any kind of a way except manually going through the whole schema. Ideally, all such enumerations would be assigned word separators to enable easier parsing in human readable forms. If such a task is too complex or demanding, at least the separators should be omitted from the table below to achieve consistent naming. Classes such as IfcElectricAppliance, IfcBeam and IfcPile are especially interesting because they contain both enumerations with and without underscore separation.

Table 5.1 Enumerations with underscores

Property	Number of enumerations with underscores (Number of total enumerations)
IfcWindow	9 (12)
IfcPlate	2 (3)
IfcFooting	5 (5)
IfcRoof	13 (14)
IfcRamp	6 (6)
IfcPile	3 (10)
IfcBeam	1 (6)
IfcDoor	17 (19)
IfcStair	14 (14)
IfcFlowInstrument	2 (8)
IfcElectricAppliance	1 (16)
IfcTendonAnchor	2 (3)
IfcExternalSpatialElement	3 (4)

While from the parsing point of view it would be easier to deal with enumerations which have underscores, recommendations for the IFC extensions seem to be going in the other direction (*Whitepaper der buildingSMART - BIM in der Energiewirtschaft*, 2023). Whichever of the strategies is adopted, normalization needs to be done and clear naming guidelines should be present to ensure uniform approach from every interested stakeholder in the future.

5.1.2. Containment in the IFC

IFC Schema is as a data schema that encompasses multiple disciplines and multiple use cases of a built asset. As presented in chapter 2.1.2.4 and applied in chapter 4.2.1, IFC enables retrieval of information which elements contain other elements (e.g. which spaces contain building elements such as furniture, plumbing fixtures etc.), important information for automated compliance checking. In practice, however current IFC file-based workflows do not enable representation of such information. A federated building model consists of multiple discipline models, without a clear mean of communication on data level. If a building model (e.g. mechanical) contains a space and a plumbing fixture (e.g. accessible toilet), another model (e.g. architectural) will contain information about the door of such a space, and there is no way to do compliance check on this object to assess whether it complies with accessibility requirements for both the mechanical and architectural elements. There are different approaches to resolve this issue. There is an approach based on computing the information directly in the IFC, after connecting all discipline models in a federated model (Teclaw et al., 2023). This approach has already seen some industrial application (*Data Soluce*). Another one is to move the connection of the elements to the linked data part of information modelling, by creating space objects in each of the elements and then using a relation such as owl:sameAs to state that the spaces are actually the same (Teclaw et al., 2023). Each of these approaches has certain drawbacks. The first approach is computationally heavy, whereas the second approach promotes duplication of IFC elements and could add additional workload for the BIM modeler.

5.1.3. Implementation of the IFC schema in authoring tools

Besides remarks about the schema itself, additional issues arise in testing the implementation of the IFC schema in authoring tools. Usually, modelling tools use proprietary, internal data models to develop the model. If such a model needs to be shared outside the proprietary ecosystem, the proprietary data model gets converted in the IFC and serialized in one of the formats, most commonly IFC-STEP. The IFC output is evaluated in the process of IFC software certification carried out by buildingSMART ("IFC Certified Software"). This however does not mean that the schema is implemented in its entirety and software implementation can still have errors while maintaining a passing grade. One such example, relevant for this research, is the implementation of the objectType IFC attribute used in chapter 4.2.1. The formal propositions of the IFC schema state that an ObjectType attribute shall be provided in case the PredefinedType attribute is set to USERDEFINED, as seen in Fig. 5.1. Object type in this sense serves as an extension of the PredefinedType property in case the IFC schema does not satisfy the requirements with its predefined enumerations and should be used only when the PredefinedType is set to USERDEFINED.

Formal Propositions

Rule	Description
CorrectPredefinedType	Either the *PredefinedType* attribute is unset (e.g. because an IfcCurtainWallType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.
CorrectTypeAssigned	Either there is no curtain wall type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcCurtainWallType.

Figure 5.1 Formal IFC propositions regarding the IFC PredefinedType and Type attributes

An authoring tool, in this example Autodesk Revit, automatically assigns its internal types which are much narrower in scope that the ObjectType attribute should be while not maintaining the same naming conventions as the PredefinedType property has, as seen in Fig. 5.2 and reported in (*Issue #502 · Autodesk/Revit-Ifc*). Such naming pattern, implemented in Autodesk Revit would make processing of the attribute to Linked Data Format impossible, due to the special characters it uses.

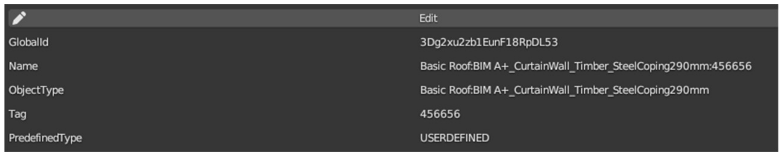

Figure 5.2 Example of the non-compliant ObjectType attribute

Revit was the only proprietary tool used in this research so no extensive research I made about IFC export on this issue. What was observed however, is that this issue does not exist when working with the BlenderBIM Add-on (*BlenderBIM Add-on*) which combines blender geometry engine and the IFC schema to provide a free, open-source and IFC native authoring platform.

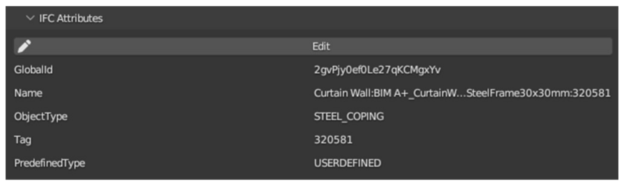

Figure 5.3 Example of a compliant ObjectType attribute

5.2. Research limitations

The research showed how two standardised data models for built environment can be connected, first theoretically, then through proof-of-concept applications. There are however certain limitations that apply.

The first limitation is data quality in the delivered IFC models persists. The proof-of-concept applications were tested on IFC models that were modelled for interoperability in the first place and therefore had a high degree of quality, such as all IFC classes and enumerations assigned. A lot of work is being done on standardization of IFC deliverables, from development of modelling guidelines and product data templates adapted to regional markets, to developing software tools that enable definition of information requirements as well as validation, e.g. (*Anker*). The appointing party should provide Exchange Information Requirements that define how data should be structured to use applications such as the ones developed in this research.

The second limitation is the IFC schema itself. While IFC schema is quite broad, it still lacks in some important fields such as tunnelling (being developed in IFC 4.4.0) or renewable energy (developments by buildingSMART Germany (*Whitepaper der buildingSMART - BIM in der Energiewirtschaft*, 2023)). While in the current workflow it would be possible to map objects under IfcBuildingElementProxy with a USERDEFINED enumeration and an objectType, such workflow creates bespoke solutions and prevents interoperability in the long term.

5.3. Further research

Considering the novelty of SWT application in the industry, as well as its academic potential, there are many possible directions future research could go in. The first, so called horizontal approach would deal with the same issue, connection of IFC with top-level, as well as other more specific data models. Firstly, a need to express the entirety of the IFC schema in semantic web stack exists, since the ifcOWL currently does not contain properties. While some research was done in translating current, XML format of IFC property sets into an OWL specification (van Berlo et al., 2019) (Lefrançois, 2020) this has not been achieved in the entirety of the schema and published as the integral part of ifcOWL, therefore any SWT-based workflows with the IFC are confined to a small subset of the schema. Moreover, this research took only a subset of the IFC schema and connected it to the EN 17632. In the future, research could be done aiming to answer how additional parts of the schema could be connected (e.g. IfcPort or IfcMaterial) as well as how to handle the objectified relationships that involve these classes (IfcRelConnectsPortToElement, IfcRelAssociatesMaterial).

The other approach would be the vertical approach which would take a subset of the IFC schema (e.g. the one developed in this research) and aim to investigate how it can be applied in practice. Examples include development of EIR and BIM modelling guidelines that enable utilization of SWT-based workflow, development of robust IFC to EN 17 632 parser with a user interface that would be usable by a wider audience (inspired by Oraskari, (2023)). Moreover, research should also be done on how, after constructing such OTLs these can be utilized in the latter part of building lifecycle. Possible use -cases are very broad. Linked data, additional standards and building permit metadata can be researched to build proof of concept applications for digital building permitting (Zentgraf et al., 2023). Linked data building models can be used to enable robot movement in construction (de Koning et al., 2021) and building element prefabrication (Kirner et al., 2023). An important use case is also to investigate how linked data could be used in asset management (Hagedorn et al., 2023) and how it could be connected to internet of things to enable streaming of sensor data for building performance monitoring (Donkers et al., 2021). These represent only a small subset of possible use cases that still need to be explored.

This page is intentionally left blank

6. CONCLUSION

The primary goal of this research was to investigate how two important standards (the ISO 16739-1:2018 or IFC, and the EN 17632-1:2022 or SML) can be interconnected with the goal of achieving semantic interoperability, required for instance in digital building permitting processes. Along with this primary goal, the secondary goal was to introduce main concepts and principles to researchers that would like to introduce themselves to the application of semantic web technologies in the built environment. This secondary goal will be considered achieved if this research inspires more young researches to take their research in the linked data direction.

The thesis begins with a detailed state of the art that explains the basics of the technologies applied. It continues by applying these technologies, firstly on a theoretical level through analysis of the standards and development of a framework that would enable their interconnection. Later, the theoretical framework was put into practice with utilization of several existing tools (GraphDB, Stardog Studio, AllegroGraph) as well the JavaScript programming language. The practical application resulted in development of scripts that enable creation of OTLs, parsing IFC files, and validation of both generic as well as project specific requirements. Several smaller contributions were made, such as practical remarks when working with existing tools as well as remarks regarding the structure of the IFC. Such organization of thesis aimed to provide a strong theoretical background for the user that could then be shown in practice with actual development, thus fulfilling both the primary and the secondary goal of the research.

This research addresses the stated research gap between IFC and SML with proof-of-concept applications while noting relevant findings along the way. It therefore adds to the body of knowledge in an increasingly popular and rapidly developing field. The main conclusion of the research is that it is indeed possible to connect the two analysed standards on taxonomical level, as well as it is possible to express IFC characteristics (currently only IFC attributes) in compliance with the modelling patterns prescribed by the SML standard). Afterwards, remarks about the IFC data model structure were presented and proposals about possible future research directions are made.

Several questions remain that lie outside the scope of this research, e.g. how should the IFC be structured in the future to enable interoperability with Linked Data concepts and how should the issue of multidisciplinarity of BIM models be addressed to prevent duplication of work and model elements. On the implementation side, questions that remain to be answered are how such information models should be utilized to bring real value to different stakeholders in the best way possible, from streamlining building permitting processes, optimising construction works to achieving better outcomes in the operations and maintenance phase.

While promising with its flexibility, the high threshold of entry construction is still hindering application of SWT in construction. Application is getting less complex with development of existing tools, application of SWT in construction still requires programming knowledge, knowledge about data structures research and knowledge about SWT. One of the focal points of future research should therefore be to make toolkits or platforms that will enable application of SWT in construction possible for a wider set of challenges, including even non-technical stakeholders.

This page is intentionally left blank

7. REFERENCES

Journal article

Borgo, S., Sanfilippo, E. M., Sˇojic, A., & Terkaj, W. (2015). Towards an Ontological Grounding of IFC.

Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., & Klein, R. (2018). The IFC to Linked Building Data Converter—Current Status.

Donkers, S., Ledoux, H., Zhao, J., & Stoter, J. (2016). Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings: Automatic conversion of IFC datasets to CityGML LOD3 buildings. Transactions in GIS, 20(4), 547–569. https://doi.org/10.1111/tgis.12162

Fauth, J., & Soibelman, L. (2022). Conceptual Framework for Building Permit Process Modeling: Lessons Learned from a Comparison between Germany and the United States regarding the As-Is Building Permit Processes. Buildings, 12(5). https://doi.org/10.3390/buildings12050638

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

Kovacs, A. T., & Micsik, A. (2021). BIM quality control based on requirement linked data. International Journal of Architectural Computing, 19(3), 431–448. https://doi.org/10.1177/14780771211012175

Laakso, M., & Kiviniemi, A. (2012). Laakso & Kiviniemi, pg. 134 www.itcon.org-Journal of Information Technology in Construction. In Journal of Information Technology in Construction (ITcon) (Vol. 17, pp. 134–161). http://www.itcon.org/2012/9

Noardo, F., Guler, D., Fauth, J., Malacarne, G., Mastrolembo Ventura, S., Azenha, M., Olsson, P.-O., & Senger, L. (2022). Unveiling the actual progress of Digital Building Permit: Getting awareness through a critical state of the art review. Building and Environment, 213, 108854. https://doi.org/10.1016/j.buildenv.2022.108854

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction, 63, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003

Petrova, E., & Pauwels, P. (2021). On the meaning of all things built: Or how the Web changes the way we design, build and operate the built environment. Intervisie, 16–19.

Ruiz-Zafra, A., Benghazi, K., & Noguera, M. (2022). IFC+: Towards the integration of IoT into early stages of building design. Automation in Construction, 136, 104129. https://doi.org/10.1016/j.autcon.2022.104129

Shkundalov, D., & Vilutienė, T. (2021). Bibliometric analysis of Building Information Modeling, Geographic Information Systems and Web environment integration. Automation in Construction, 128, 103757. https://doi.org/10.1016/j.autcon.2021.103757

Studer, R., B'c, V. R. B., & Fensel, D. (1998). Knowledge Engineering: Principles and methods. In Data & Knowledge Engineering (Vol. 25, pp. 161–197).

Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P., & Gao, X. (2019). A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. Automation in Construction, 101, 127–139. https://doi.org/10.1016/j.autcon.2019.01.020

Ullah, K., Raitviir, C., Lill, I., & Witt, E. (2020). BIM adoption in the AEC/FM industry – The case for issuing building permits. International Journal of Strategic Property Management, 26(4), 400–413. https://doi.org/10.3846/ijspm.2020.13676

Ullah, K., Witt, E., & Lill, I. (2022). The BIM-Based Building Permit Process: Factors Affecting Adoption. Buildings, 12(1). https://doi.org/10.3390/buildings12010045

Yang, Y., Pan, Y., Zeng, F., Lin, Z., & Li, C. (2022). A gbXML Reconstruction Workflow and Tool Development to Improve the Geometric Interoperability between BIM and BEM. Buildings, 12(2), 221. https://doi.org/10.3390/buildings12020221

Book

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building information modeling: Technology foundations and industry practice. In Building Information Modeling: Technology Foundations and Industry Practice. Springer International Publishing. https://doi.org/10.1007/978-3-319-92862-3

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. de, Gutierrez, C., Gayo, J. E. L., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngomo, A.-C. N., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2020). Knowledge Graphs. https://doi.org/10.1145/3447772

Book chapter

Antoniou, G., Franconi, E., & Harmelen, F. V. (2005). Introduction to Semantic Web ontology languages. *Lecture Notes in Computer Science*, *3564*, 1–21. https://doi.org/10.1007/11526988_1

Daum, S., Borrmann, A., & Kolbe, T. H. (2017). A Spatio-Semantic Query Language for the Integrated Analysis of City Models and Building Information Models. In A. Abdul-Rahman (Ed.), *Advances in 3D Geoinformation* (pp. 79–93). Springer International Publishing. https://doi.org/10.1007/978-3-319-25691-7_5

E-book

Gayo, J. E. L., Prud'hommeaux, E., Boneva, I., & Kontokostas, D. (2018). *Validating RDF Data*. https://book.validatingrdf.com/

Conference proceedings

Bazuin, B., Stolk, S., & Stevens, M. (2023, June). Towards Recommendations for Knowledge-Graph-Based Requirements Management in Construction: A Report on the DigiChecks Project.

Chamari, L., Petrova, E., & Pauwels, P. (2022). A web-based approach to BMS, BIM and IoT integration: A case study Holistic BIM-Based Sustainable Building Design and Performance Assessment View project Flemish Cities in Transition-A Framework for a Web-based Built Heritage Platform Using BIM, GIS and Linked Data View project A web-based approach to BMS, BIM and IoT integration: A case study. https://doi.org/10.34641/clima.2022.228

Donkers, A., Yang, D., de Vries, B., & Baken, N. (2021). Real-Time Building Performance Monitoring using Semantic Digital Twins.

Kirner, L., Oraskari, J., Jung, V., & Brell-Cokcan, S. (2023). dstv: An ontology-based extension of the DSTV-NC standard for the use of linked data in the automation of steel construction.

Noardo, F., Malacarne, G., Ventura, S. M., Tagliabue, L. C., Ciribini, A. L. C., Ellul, C., Guler, D., Harrie, L., Senger, L., Waha, A., & Stoter, J. (2020). Integrating expertises and ambitions for data driven digital building permits—The EUnet4DBP. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives.

Pauwels, P., & Deursen, D. V. (2012). IFC/RDF: Adaptation, Aggregation and Enrichment.

Rasmussen, M. H., Pauwels, P., Hviid, C. A., & Karlshøj, J. (2017). Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. 237–244. https://doi.org/10.24928/jc3-2017/0153

Salheb, N., Arroyo Ohori, K., & Stoter, J. (2020, September 7). Automatic Conversion of CityGML to IFC. 2020 3rd BIM/GIS Integration Workshop and 15th 3D GeoInfo Conference.

Terkaj, W., & Pauwels, P. (2017). A Method to generate a Modular ifcOWL Ontology. https://www.researchgate.net/publication/320077313

Teclaw, W., Rasmussen, M. H., Labonnote, N., & Oraskari, J. (2023). The semantic link between domain-based BIM models.

van Berlo, L., Willems, P., & Pauwels, P. (2019). Creating Information Delivery Specifications using Linked Data.

Zentgraf, S., Fauth, J., Hagedorn, P., Seiss, S., Smarsly, K., König, M., & Melzner, J. (2023). OntoBPR: Ontology-based workflow and concept for building permit reviews.

Zhang, Z., Nisbet, N., Ma, L., & Broyd, T. (2022). A multi-representation method of building rules for automatic code compliance checking.

Dissertations

Bonduel, M. (2021). Faculty of Engineering Technology A Framework for a Linked Data-based Heritage BIM.

Nuyts, E., Present, P., & Werbrouck, J. (2022). Automated validation of building models against legislation using Linked Data. http://lib.ugent.be/catalog/rug01:003063643

Schlachter, A. (2020). Unlocking the value of Linked Building Data (LBD) A lean and integrated management process of temporary construction items (TCIs).

Werbrouck, J. (2017). Linking Data: Semantic enrichment of the existing building geometry.

Standards

Koehorst, B., Bohms, M., Asmussen, S., Jackson, P., Kantor, D., Klein, R., Schroeder, M., Avalos, H., & Jansweijer, S. (2022, July 31). *ISO 17632:2021—BIM Semantic modelling and linking*. https://repozitorij.hzn.hr/norm/HRN+EN+17632-1%3A2022

Millais, J.E. (1851-1852) Ophelia [online]. Available at: www.tate.org.uk/art/artworks/millais-ophelia-n01506 (Accessed: 21 June 2014)

Reports

ACCA Software. (2023). IFC file whitepaper. ACCA Software. https://s3-eu-west-1.amazonaws.com/acca.biblus.downloads/International/IFC/IFC-file_Whitepaper_ENG.pdf

Chakaroun, K., Onofri, L., Orellana, E., Jayashree, S., & Umarov, F. (2023). Doing Business Case Studies From Paper to the Cloud-Improving Building Control through E-permitting. World Bank.

Dealing with construction permits. (2020). 18/02/2023https://archive.doingbusiness.org/en/data/exploretopics/dealing-with-construction-permits

Harrie, L., & Jensen, A. B. O. (2018). Description of geodata quality with focus on integration of BIM-data and geodata. https://doi.org/10.13140/RG.2.2.28011.64806

Technical Roadmap. (2020). buildingSMART International.

The Information Economy: A Study of Five Industries. (2014). box. https://img.forconstructionpros.com/files/base/acbm/fcp/document/2014/06/box-cloud-study_11535206.pdf

Whitepaper der buildingSMART - BIM in der Energiewirtschaft. (2023). buildingSMART.

Presentations

Bizer, C., Heath, T., & Berners-Lee, T. (2008). Linked Data: Principles and State of the Art.

Bonduel, M. (2023, December 6). Summer School of Linked Data in Architecture and Construction Lecture—Ontologies. https://linkedbuildingdata.net/ldac2023/summerschool/files/presentations/SSoLDAC2023_lecture2_ontologies.pdf

Bonduel, M., & Wagner, A. (2023, June). Describing Products in a Linked Data World. LDAC 2023, Matera.

buildingSMART International (Director). (2023, January 26). Towards automated regulatory compliance in the EU. https://www.youtube.com/watch?v=4Qiyx61TSSk

Pauwels, P. (2023). BIM, Linked Data, and Digital Twinning BIM A+ Guest Lecture.

Sack, H. (2016). Linked Data Engineering. OpenHPI. https://open.hpi.de/courses/semanticweb2016

Turk, Ž. (2023). BIM A+4: Advanced BIM data systems and interoperability—Connective Interoperability, BIM A+ Lectures.

Turk, Ž. (2023c). BIM A+4 Lecture Slides: IFC.

Van Berlo, L. (2022). Advancements in Building Data Exchange with IFC and Semantic Web Technologies. https://www.youtube.com/watch?v=Yt6kG_Wnx6M&t=278s&ab_channel=IBPSAUSAWebsite

Websites

ACCORD Project. (2023, May 10). ACCORD Project. https://accordproject.eu/

Anker, Retrieved August 7, 2023, from https://ankerdb.com

BlenderBIM Add-on—Beautiful, detailed, and data-rich OpenBIM. Retrieved August 6, 2023, from https://blenderbim.org/

Britannica. (2023, May 23). World Wide Web | History, Uses & Benefits | Britannica. https://www.britannica.com/topic/World-Wide-Web

buildingSMART Data Dictionary. (2023). BuildingSMART International. https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/

buildingSMART. (2023a). IFC Formats. BuildingSMART Technical. https://technical.buildingsmart.org/standards/ifc/ifc-formats/

buildingSMART. (2023b). IFC Schema Specifications. BuildingSMART Technical. https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/

buildingSMART. (2023c). IfcProductExtension. https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/annex/annex-d/ifcproductextension/diagram_0002.htm

buildingSMART. (2023d). IfcRelationship. https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/schema/ifckernel/lexical/ifcrelationship.htm

buildingSMART. (2023e). IfcRelContainedInSpatialStructure. https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/schema/ifcproductextension/lexical/ifcrelcontainedinspatialstructure.htm

buildingSMART. (2023f). Model View Definitions (MVD). BuildingSMART Technical. https://technical.buildingsmart.org/standards/ifc/mvd/

CHEKdbp. (2023). https://chekdbp.eu/

Data soluce Retrieved August 3, 2023, from https://www.datasoluce.com/

Davies, T. (2022, September 6). Breaking down the IFC 4.3 schema. AEC Magazine. https://aecmag.com/collaboration/breaking-down-the-ifc-4-3-schema/

DigiChecks. (2023, June 13). DigiChecks. https://digichecks.eu/

DigiPLACE, Retrieved June 26, 2023, from https://digiplaceproject.eu/

George. (2014, February 5). StackOverflow. Stack Overflow. https://stackoverflow.com/a/21566583

History and versions of IFC – BIM Supporters.. Retrieved September 8, 2023, from https://app.bimsupporters.com/courses/ifc/lessons/history-and-versions-of-ifc/

IFC Certified Software, BuildingSMART International. Retrieved August 3, 2023, from https://www.buildingsmart.org/compliance/software-certification/certified-software/

IFCtoRDF-Desktop [Java]. https://github.com/jyrkioraskari/IFCtoRDF-Desktop (Original work published 2017)

LD-BIM - BIM meets Linked Data. Retrieved June 26, 2023, from https://ld-bim.web.app/

Linked Data—Design Issues. (2023). https://www.w3.org/DesignIssues/LinkedData.html

Merriam-Webster Retrieved July 13, 2023, from https://www.merriam-webster.com/

Oraskari, J. (2023). IFCtoRDF-Desktop [Java]. https://github.com/jyrkioraskari/IFCtoRDF-Desktop (Original work published 2017)

RDF 1.2 Schema., Retrieved September 5, 2023, from https://www.w3.org/TR/rdf12-schema/

Revit incorrectly uses the ObjectType attribute · Issue #502 · Autodesk/revit-ifc. GitHub. Retrieved August 6, 2023, from https://github.com/Autodesk/revit-ifc/issues/502

Rasmussen, M. H., LD-BIM - BIM meets Linked Data. Retrieved June 26, 2023, from https://ld-bim.web.app/

SKOS Simple Knowledge Organization System Reference. Retrieved September 5, 2023, from https://www.w3.org/TR/skos-reference/

The Linked Open Data Cloud. (2020). http://cas.lod-cloud.net/

The status of IFC 4.3 and the benefit of further extensions as IFC 4.4—buildingSMART International. (2022, June 7). https://www.buildingsmart.org/the-status-of-ifc-4-3-and-the-benefit-of-further-extensions-as-ifc-4-4/

Wallin, E., & Fierro, G. (2022, July 1). Brick Schema and RealEstateCore announce a major harmonization effort between two smart building metadata standards. RealEstateCore. https://www.realestatecore.io/brickrec/

OWL 2 Web Ontology Language Document Overview (Second Edition). Retrieved September 5, 2023, from https://www.w3.org/TR/owl2-overview/

LIST OF ACRONYMS AND ABBREVIATIONS

BIM	Building Information Modelling
OECD	Organization for Economic Cooperation and Development
DBP	Digital building permitting
ACC	Automated compliance checking
GIS	Global Information System
SWT	Semantic Web Technologies
IFC	Industry Foundation Classes
SHACL	Shapes Constraint Language
LDAC	Linked Data in Architecture and Construction
W3C LBD CG	World Wide Web Consortium Linked Building Data Community Group
STEP	Standard for the Exchange of Product Model
IAI	International Alliance of Interoperability
XML	Extensible Markup Language
TURTLE	Terse RDF Triple Language
RDF	Resource Description Framework
JSON	JavaScript Object Notation
HDF	Hierarchical Data Format
CSG	Computational Solid Geometry
bSDD	buildingSMART Data Dictionary
MVD	Model View Definition
ER	Exchange Requirements
CDE	Common Data Environment
BEM	Building Energy Modelling
IoT	Internet of Things

DEL	Direct Edge Labelled
URI	Universal Resource Identifier
URL	Universal Resource Location
CWA	Closed World Assumption
OWA	Open World Assumption
UNA	Unique Name Assumption
NUNA	No Unique Name Assumption
SKOS	Simple Knowledge Organization System
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
NLP	Natural Language Processing
BOT	Building Topology Ontology
OPM	Ontology for Property Management
UPonto	Uniform Project Ontology
OMG	Ontology for Managing Geometry
FOG	File Ontology for Geometry formats
EDM	Electronic Document Management
OTL	Object Type Library
AEC	Architecture, Engineering and Construction
PRM	Permit Requirements Management

APPENDICES

APPENDIX 1: LIST OF EXTERNAL LIBRARIES AND FUNCTIONS

```
import _ from "lodash";
import namespace from "@rdfjs/namespace";
import SparqlClient from "sparql-http-client";
import cf from "clownface";
import rdf from "rdf-ext";
import { skos, owl, rdfs } from "@tpluscode/rdf-ns-builders";
import { rdf as rdff } from "@tpluscode/rdf-ns-builders";
import { Readable } from "readable-stream";
import getStream from "get-stream";
import {
  TurtleSerializer,
  JsonLdSerializer,
} from "@rdfjs-elements/formats-pretty";
import fs from "fs";
import {
  PredefinedTypeEnumerationQuery,
  ifcRootAttributeQuery,
  belowAttributesNoEnumerationsQuery,
  upperCaseFirst,
  lowerCaseFirst,
  enumerationAttributeQuery,
  classAtrributeEnumerationQuery,
  fullEnumerationsInAttributesQuery,
  queriedElements,
  ifcToDiscreteObjectArray,
  createIfcEntityClassNode,
  createEnumerationClassNode,
  createOpeningNode,
  otlGraph,
  ns,
  createNode,
  createBlankNode,
} from "./utils.js";
import {
  IfcAPI,
  IFCRELCONTAINEDINSPATIALSTRUCTURE,
  IFCRELAGGREGATES,
  IFCELEMENT,
  IFCDOOR,
  IFCSPACE,
```

```
  IFCPROJECT,
  IFCUNITASSIGNMENT,
  IFCWINDOW,
  IFCPRODUCT,
  IFCSPATIALELEMENT,
} from "web-ifc";
import namespace from "@rdfjs/namespace";
import cf from "clownface";
import rdf from "rdf-ext";
import _ from "lodash";
import { rdf as rdff, rdfs, xsd } from "@tpluscode/rdf-ns-builders";
import {
  TurtleSerializer,
  JsonLdSerializer,
} from "@rdfjs-elements/formats-pretty";
import getStream from "get-stream";
import { Readable } from "readable-stream";
import { get } from "http";
import { create } from "domain";
import fs from "fs";
import factory from "rdf-ext";
import ParserN3 from "@rdfjs/parser-n3";
import SHACLValidator from "rdf-validate-shacl";
import {
  TurtleSerializer,
  JsonLdSerializer,
} from "@rdfjs-elements/formats-pretty";
import getStream from "get-stream";
import { Readable } from "readable-stream";
```