Data Representation in Steel Pre-fabrication: A Multi-Modal Approach for Interoperability, Automation and Reuse

Datenrepräsentation in der Stahlvorfertigung: Ein multimodaler Ansatz für Interoperabilität, Automatisierung und Wiederverwendung

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

Vorgelegt von

Victoria Jung

geboren in Wuppertal, Deutschland

Berichtende:

Univ.-Prof. Dr. techn. Sigrid Brell-Cokcan

Univ.-Prof. Dr.-Ing. Markus Feldmann

Tag der mündlichen Prüfung: 16.12.2025

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Danksagung

Bevor ich auf die inhaltlichen Beiträge dieser Arbeit eingehe, möchte ich zunächst all jenen danken, die mich auf dieser Forschungsreise begleitet und wesentlich unterstützt haben.

Mein erster Dank gilt meiner Professorin und Betreuerin, Univ.-Prof. Dr. techn. Sigrid Brell-Cokcan. Durch ihre visionäre Perspektive, ihre Offenheit für neue Ideen und ihre stetige Unterstützung hat sie den Rahmen geschaffen, in dem diese Arbeit entstehen konnte. Ihr Vertrauen in meine wissenschaftliche und persönliche Entwicklung hat mich über die gesamte Promotionszeit hinweg begleitet und motiviert. Ebenso danke ich Herrn Univ.-Prof. Dr.-Ing. Markus Feldmann, der als Zweitgutachter diese Arbeit begleitet hat.

Für die inspirierende Zusammenarbeit, den fachlichen Austausch und die vielen Stunden auf der Referenzbaustelle danke ich meinen Kolleginnen und Kollegen vom Lehrstuhl für Individualisierte Bauproduktion und Center Construction Robotics. An dieser Stelle möchte ich Christoph Heuer, Lukas Kirner, und Jyrki Oraskari hervorheben. Mein Dank gilt Heinrich Knitt und Emre Ergin für ihre tatkräftige Unterstützung bei den robotischen Versuchen sowie allen anderen, die meine Zeit sehr besonders gemacht haben: Thomas Adams, Iremnur Tokac Celikyay, Wu Han Chu, Baris Cokcan, Görkem Ertemli, Luca Fahrendholz-Heiermann, Philippe Gülpen, Martin von Hilchen, Nico Horsch, Carsten Kamp, Ethan Kerber, Ajith Krishnan, Joo Lee, Davide Picchi, Pranav Shevkar, Sven Stumm, Julia Weishäupl, Peter Wildemann, Burak Yirmibes und Marit Zöcklein. Ein ganz besonderer Dank gilt Stefanie Steinfeld, die in jeder Situation mit Rat und Tat zur Seite stand und stets ein offenes Ohr hatte. Ein großer Dank geht auch an unsere studentischen Hilfskräfte, deren engagierter Einsatz zum Gelingen dieser Arbeit beigetragen hat. Besonders erwähnen möchte ich Simon Laprell.

Nicht zuletzt danke ich den Partnern der Ad-hoc-Gruppe des IT-Ausschusses von bauforumstahl e.V. für die konstruktive Zusammenarbeit und die praxisnahen Einblicke in die industrielle Prozesse, die diese Arbeit entscheidend mitgeprägt haben.

Mein besonderer Dank gilt schließlich meiner Familie sowie meinen Freundinnen und Freunden, die mir in all den Jahren mit Geduld, Optimismus und Rückhalt zur Seite standen.

Abstract

In view of the growing demand for sustainable construction methods, building materials with high reusability and deconstructable connection techniques are increasingly coming to the fore. In this particular context, steel has been identified as having the potential to positively contribute to the reduction of emissions and waste. One of the key challenges for the widespread implementation of steel reuse is the lack of a consistent, digital and interoperable data basis. This not only affects the deconstruction phase, but is also important along the entire value chain in the construction industry. The upstream production of steel requires both horizontal digitalisation strategies that enable standardised data exchange between different actors and vertical strategies are important to connect internal processes, systems and stakeholders. However, the presence of inconsistent data formats and proprietary software solutions increase the complexity of such data exchange.

While the DSTV-NC format has become an established standard for representing geometric data and enabling conventional steel fabrication, it no longer meets the requirements for modern interoperability or supports emerging fabrication methods such as robotic processing. Advancing digital fabrication and facilitating material reuse in line with circular construction principles therefore requires a more flexible and semantically enriched data model.

This dissertation proposes a multi-modal approach to data representation in steel pre-fabrication. Both the Industry Foundation Classes (IFC) schema and ontology-based frameworks are investigated as potential successors to the DSTV-NC format. A detailed analysis of the data structures required for novel fabrication and reuse processes form the basis for the development of both models. Their practical applicability is evaluated through prototype implementations in robotic steel fabrication. Based on the findings, a combined approach is recommended. The IFC schema serves as a standardised data model that allows cross-project information to be stored within a Common Data Environment (CDE), ensuring accessibility for all relevant parties. Ontology-based models can complement this by enabling flexible, domain-specific modelling of detailed data. This interoperable data architecture supports both horizontal and vertical digitalisation. The dissertation also examines the practical challenges of integrating the proposed data model into existing planning and fabrication workflows, and identifies specific opportunities for standardisation. The outcomes contribute to the advancement of digital transformation in steel construction and simultaneously promote circular economy principles through enhanced data interoperability and process automation.

Kurzfassung

Angesichts des steigenden Bedarfs an nachhaltigen Bauweisen rücken Baustoffe mit hoher Wiederverwendbarkeit und rückbaufähigen Verbindungstechniken zunehmend in den Fokus. Insbesondere Stahl verfügt in diesem Zusammenhang über vielversprechende Eigenschaften, die einen positiven Beitrag zur Reduzierung von Emissionen und Müll leisten können. Eine der zentralen Herausforderungen für die breite Umsetzung der Wiederverwendung von Stahl ist das Fehlen einer durchgängigen, digitalen und interoperablen Datengrundlage. Dies betrifft nicht nur die Rückbauphase, sondern ist entlang der gesamten Wertschöpfungskette im Bauwesen von Bedeutung. Durch die vorgelagerte Fertigung von Stahl sind sowohl horizontale Digitalisierungsstrategien, die einen standardisierten Datenaustausch zwischen verschiedenen Akteuren ermöglichen, als auch vertikale Strategien gefragt, um innerbetriebliche Prozesse und Stakeholder zu vernetzen. Derzeit erschweren jedoch uneinheitliche Datenformate und proprietäre Softwarelösungen einen solchen Datenaustausch erheblich.

In der Stahlfertigung hat sich das DSTV-NC Format als gängiger Standard zur Beschreibung von Geometriedaten und zur Ansteuerung konventioneller Fertigungsmaschinen etabliert. Es erreicht jedoch seine Grenzen bei der Erfüllung moderner Interoperabilitätsanforderungen und der Unterstützung neuer automatisierter Fertigungsprozesse, wie der robotischen Bearbeitung. Um die Digitalisierung der Fertigung und die Wiederverwendung von Bauteilen im Sinne einer zirkulären Bauwirtschaft voranzutreiben, ist ein flexibleres und semantisch reichhaltigeres Datenmodell erforderlich.

Im Rahmen dieser Forschungsarbeit wird ein multi-modaler Ansatz zur Datenrepräsentation in der Stahlvorverarbeitung untersucht. Dabei werden sowohl das Industry Foundation Classes (IFC)-Schema als auch ontologiebasierte Frameworks als mögliche Nachfolger zum DSTV-NC Format betrachtet. Eine detaillierte Analyse der für neue Fertigungs- und Wiederverwendungsprozesse erforderlichen Datenstrukturen bildet die Grundlage für die Entwicklung. Ihre Praxistauglichkeit wird anhand prototypischer robotergestützter Stahlfertigung evaluiert. Angesichts der Ergebnisse wird ein kombinierter Ansatz empfohlen. Das IFC-Schema dient als standardisiertes Datenmodell, mit dem projektübergreifende Informationen innerhalb einer Common Data Environment (CDE) gespeichert und zugänglich gemacht werden. Ergänzend ermöglichen ontologiebasierte Modelle die flexible und domänenspezifische Abbildung von Daten. Diese kompatible Datenarchitektur unterstützt horizontale und vertikale Digitalisierung. Die Dissertation analysiert praktische Herausforderungen bei der Integration des Datenmodells in bestehende Planungs- und Fertigungsprozesse, zeigt Standardisierungsansätze auf und trägt so zur digitalen Transformation im Stahlbau sowie zur Förderung zirkulärer Prinzipien durch verbesserte Dateninteroperabilität bei.

Contents

Acronyms

AEC Architecture, Engineering, and Construction

AI Artificial Intelligence

AISC American Institute of Steel Construction

ASCII American Standard Code for Information Interchange

API Application Programming Interface

bSDD buildingSMART Data Dictionary

bSI buildingSMART International

bSUCM buildingSMART Use Case Management

BPMN Business Process Model and Notation

CAD Computer Aided Design

CAM Computer Aided Manufacturing

CDE Common Data Environment

CNC Computer Numerical Control

CQ Competency Question

CRC Cloud Remote Control

DFReu Design for Reuse

Dfa Design for disassembly

Dfd Design for deconstruction

DIHK Deutsche Industrie- und Handelskammer

DIN Deutsches Institut für Normung

DSRP Design Science Resource Process

DSTV Deutscher Stahlbau Verband

EM Exchange Modules

EOL End of Life

ER Entity-Relationship

ERD Entity-Relationship Diagram

HRI Human Robot Interaction

IAI Industry Alliance for Interoperability

ICDD Information Container for linked Document Delivery

IDM Information Delivery Manual

IDS Information Delivery Specification

IFC Industry Foundation Classes

IoC Internet of Construction

IoT Internet of Things

IRI Internationalised Resource Identifier

ISO International Organisation for Standardisation

JSON Java Script Object Notation

KUKA | prc KUKA | Parametric Robot Control

LBD Linked Building Data

LBDCG Linked Building Data Community Group

LCA Life Cycle Assessment

LOD Level of Development

LOIN Level of Information Need

MQTT Message Queuing Telemetry Transport

MVD Model View Definition

mvdXML Model View Definition XML

NC Numerical Control

OO Object-Oriented

OOM OO Modelling

OWL Ontology Web Language

Pset Property Set

RDF Resource Description Framework

RFDS Resource Description Framework Schema

SME Small and Medium Enterprises

SPARQL The SPARQL Protocol And RDF Query Language

STEP Standard for the Exchange of Product Model Data

TTL Terse RDF Triple Language

UML Unified Modelling Language

URI Uniform Resource Identifiers

WLAN Wireless Local Area Network

W3C World Wide Web Consortium

WWW World Wide Web

XML Extensible Markup Language

List of Figures

List of Tables

Chapter 1

Introduction

The first chapter of this thesis introduces the research conducted by highlighting the motivation (see Section 1.1), the context and collaborative framework in which the research originated (see Section 1.2), and outlining the hypothesis and respective research questions (see Section 1.3). Section 1.4 summarises the specific contribution of this thesis. The final Section 1.5 focuses on the applied research methodology and provides an overview of the subsequent manuscript.

1.1 Motivation

The concepts of digitalisation, automation, and the circular economy have significantly influenced industrial development and research agendas in recent years [7]. By developing scalable and interoperable solutions, research plays a crucial role in bridging the gap between theoretical advances and practical industrial applications. This thesis is therefore based on various research projects and close collaboration with industry partners, as detailed in Section 1.2. It contributes to concise data representation in steel pre-fabrication through a multi-modal approach for reuse and automation.

While manufacturing and automotive sectors have long embraced Industry 4.0 principles integrating the Internet of Things (IoT), artificial intelligence (AI), and big data to enhance productivity and resource efficiency, the construction industry lags behind [8]. This delay is due to the sector's fragmented stakeholder landscape, dynamic site conditions, and continued reliance on manual labour [9]. Steel fabrication exemplifies these challenges, particularly regarding digital integration, automation, and continuity of data [10], each crucial for enabling circular construction practices [11].

The construction sector is responsible for 37% of global operational and process-related CO_2 emissions [12], underscoring its central role in global sustainability efforts. To meet the goals articulated in EU climate policy [13], the sector must adopt more sustainable practices. Implementing circular economy strategies can help minimise resource consumption and reduce environmental impacts [13]. Steel has particular potential in this regard [14], as it can be manufactured entirely from recycled scrap (secondary steel) or from a mix of recycled and virgin material (primary steel). However, the reuse of steel elements is currently limited by several barriers [15]. These include a lack of standardised information on component condition, provenance, and material properties, as well as variability in geometric tolerances and the absence of consistent certification mechanisms. Although some data can be collected during prefabrication, it is typically not preserved in a structured format and is thus unavailable when components are deconstructed.

Digitalisation in steel fabrication requires a comprehensive strategy that addresses both vertical and horizontal integration of data and processes within the construction value chain [1]. Figure 1.1 illustrates this dual approach, highlighting the necessity of vertical integration, linking material import and design through field-level fabrication, operational management, and client requirements, as well as horizontal integration, which ensures data exchange and interoperability across various stakeholders, including pre-fabrication, successive trades, and onsite activities.

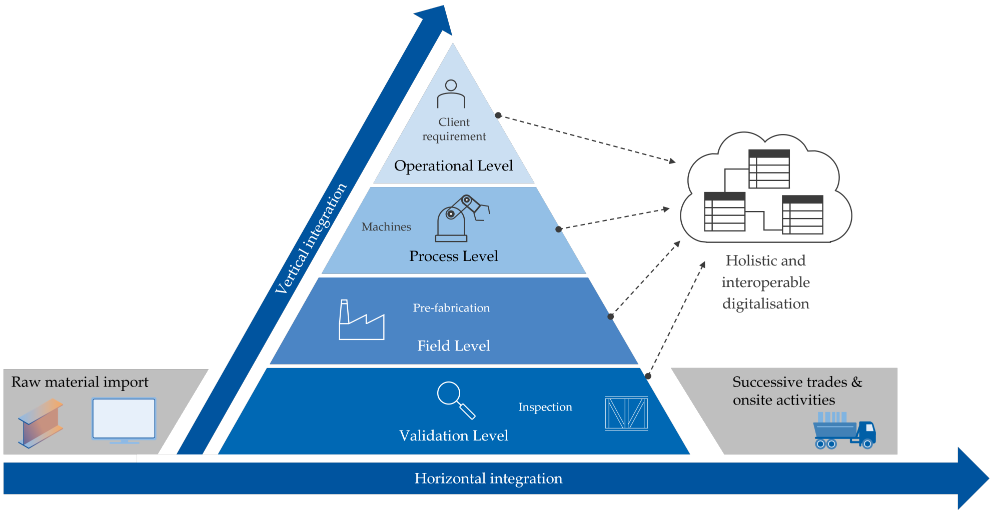


FIGURE 1.1: Concept of achieving horizontal and vertical integration of data between stakeholders in the steel industry, based on You et al. [1]

Vertical integration focuses on consolidating data flows inside an organisation, enabling a multi-modal digital ecosystem where information is continuously captured, processed, and utilised across robotic fabrication, automated production, and quality inspection. This internal connectivity is essential to enhance efficiency, maintain data continuity, and enable real-time feedback loops that optimise manufacturing workflows.

In parallel, horizontal integration facilitates collaboration across the entire construction supply chain by establishing interoperable data standards that transcend company boundaries. This ensures that data generated at early stages, such as design and pre-fabrication, remains accessible and meaningful during later phases, including inspection, assembly, and eventual reuse or deconstruction. Such interoperability is crucial for enabling circular economy principles, where the lifecycle of steel components is extended through reliable data provenance and reuse potential.

The combined vertical and horizontal integration strategy underpins the development of advanced data models that support both automation and sustainability goals in steel fabrication.

Currently, the numerical control (NC) file format established by the Deutscher Stahlbau Verband (DSTV) is used to describe the geometry and fabrication features of steel parts, such as beams, columns and plates [16, 17]. The DSTV-NC serves as an interface between CAD systems and fabrication machines, and is specifically designed to describe different processes in a standardised way. However, it is only used in steel pre-fabrication and is not shared along the value chain. Although the DSTV-NC format is widely used, it no longer meets new fabrication and modelling requirements. It falls short of providing the extensibility and bidirectional data flows necessary for an integrated approach.

The Industry Foundation Classes (IFC) and the principles of Linked Data are increasingly being combined to facilitate data interoperability throughout the entire lifecycle of the construction value chain [18], providing a foundation for an successor to the DSTV-NC format. Therefore, this thesis investigates a multi-modal data representation framework that addresses these challenges by combining an IFC-based model view and a domain ontology to enable data exchange, robotic fabrication, and lifecycle reuse of steel components.

1.2 Research context and collaborative framework

The outlined challenges have been identified over time through various research projects and close collaboration with industry partners. This setting provides a unique environment in which to conduct the investigations and development work that form the focus of this thesis.

The Internet of Construction (IoC) project [19] conducted foundational work investigating the digital integration of value chains in the construction sector. The project developed initial approaches for cross-stakeholder and cross-platform data interoperability. Particular attention was paid to existing practices and potentials within steel construction, with a focus on implementing robotic technologies and Industry 4.0 principles. Rather than focusing on standardised data exchange formats, the project prioritised remote control of production processes via cloud technologies, as well as the integration of haptic robot programming and human–machine interfaces [20]. A significant challenge identified during the IoC project was how to handle and document manufacturing tolerances, especially with regard to storing and reusing measured data. Building on these insights, the subsequent BauFeSt 4.0 project focused on overcoming limitations in the DSTV-NC format relating to the documentation of tolerances for steel base materials and plates. The project provided the foundation for developing a model for storing measured values, as well as exploring ways to digitise existing tolerances [21].

In parallel, the EU-funded TARGET-X project is investigating robotic deconstruction processes in a construction-specific context. The project aims to identify the necessary data structures and machine developments to facilitate circular construction workflows. Using a multi-material demonstrator named ReStage, the project examines steel components to determine the required data foundations for deconstruction and how such data can be generated during the pre-fabrication phase [22].

Alongside the research efforts at the Chair of Individualized Production in Architecture at the RWTH Aachen University, which focuses on interoperability and digital data formats in construction, the ad hoc group *NC* of the Information Technology Working Committee of bauforumstahl e.V. is actively working on mapping the DSTV-NC format into the IFC schema [23, 24]. IFC has been identified in industry as a more appropriate and capable data format to overcome the limitations of DSTV-NC, particularly with regard to extensibility, semantic richness, and its suitability for integrating fabrication data into broader digital workflows. In collaboration with the group the aim is to both define technical requirements and drive forward the development and implementation of IFC in steel pre-fabrication. The requirements and the analysis of existing standards (DSTV-NC & EM.11) were defined and conducted in close conjunction with key industrial stakeholders in the sector. These include a steel fabrication machinery supplier (Kaltenbach GmbH & Co. KG), two steel fabrication companies (Wurst Stahlbau GmbH and Lamparter Stahlbau GmbH & Co. KG), a steel detailing and work preparation software provider (Huhn EDV Beratung) and a steel construction industry association (bauforumstahl e.V.). Individual representatives from these partners actively contribute to the ad hoc group.

The projects and industrial collaborations outlined above form the basis of the research questions addressed in this dissertation. These questions aim to systematically explore the scientific and practical implications of digital fabrication, data integration and robotics in the context of steel pre-fabrication.

1.3 Hypothesis and research questions

Hypothesis

To overcome the limitations of isolated, non interoperable, and non extensible file formats currently used in steel prefabrication, this thesis proposes a multimodal data representation approach that enables both vertical integration, within fabrication workflows such as robotic automation and quality assurance, and horizontal integration, across planning, execution, and reuse stakeholders. A single, universally valid data model is insufficient due to the diversity of stakeholder requirements and use cases. Therefore, this thesis investigates a dual approach: the development of an IFC-based Model View Definition tailored to steel fabrication processes, and the creation of a domain ontology to structure and link relevant data across different contexts. This approach aims to support three key application areas: robotic fabrication, the integration of planning and production data, and data-driven reuse of steel components.

The hypothesis is that these two complementary representations can together serve as a viable and more flexible successor to the DSTV-NC format, enabling automation, lifecycle data continuity, and reuse in the digital steel construction industry.

Research questions

The hypothesis is that a multi-modal data model, as a successor to the DSTV-NC format, can overcome the current limitations. This raises the question of what data and how such a model could be structured to work in practice in a standardised way as a comprehensive data exchange format. This leads to four research questions, which are described below:

1. *What categories of data are essential to connect planning, pre-fabrication, and automated fabrication processes to optimise the lifecycle and enable the reuse of structural steel components in construction?*
   This research aims at analysing the different requirements and structuring them into a conceptual model as a basis for developing the data model.

2. *How can a data model be designed to effectively support both the reuse of steel components and the data requirements of automated fabrication processes in steel construction?*
   The question is addressed by a twofold approach to develop and evaluate two data models to meet the requirements and serve as a successor to the DSTV-NC standard.

3. *How can the developed data models be implemented to successfully run robotic fabrication processes and store processed data, and what are the differences regarding implementation complexity, data consistency, and scalability?*
   To ensure that the models developed meet their requirements, this question is focused on demonstrating their applicability to robotic steel fabrication processes and providing a holistic evaluation of both models.

4. *What fundamental steps and challenges arise in integrating the data model into existing planning and steel fabrication workflows to promote standardisation within the industry?*
   As it is important not only to develop a data model, but more importantly to promote standardisation and implementation into industrial practice, the final research question analyses the existing entities that maintain each technology, but also identifies potential barriers in the steel industry and how to overcome them to provide a guide to follow.

1.4 Contribution

This thesis contributes to the standardisation of steel-specific data exchange and the integration of robotic fabrication requirements into open data models. As a result, the first steel-related property sets have been published in the buildingSMART Data Dictionary (bSDD) in collaboration with buildingSMART and bauforumstahl e.V., and the updated DSTV ontology is now publicly available [25] through a dedicated ontology search platform [26]. In addition to the development of modelling standards, the thesis systematically analyses the information requirements of robotic steel fabrication workflows, including data relevant for reuse-oriented processes. Importantly, the work highlights the role of close collaboration between research and industry as a driver for innovation, ensuring that technical developments are grounded in practical needs while also embracing new technologies and evolving requirements emerging from the research domain. While a dedicated IFC-based Fabrication View was developed and tested, it does not yet include supporting IDS or mvdXML documents, which underscores that it is not a complete standard within the buildingSMART framework.

Finally, the thesis addresses emerging challenges in data governance, particularly regarding lifecycle traceability and component reuse in line with circular economy goals. It provides recommendations for how these developments can be practically integrated into industry. It is highlighted that full adoption will require shared responsibility, standardisation and cross-sector collaboration among software vendors, machinery manufacturers and end-users.

1.5 Methodology and structure

Overall methodology

The research method employed in this thesis is a variant of the Design Science Research (DSR) methodology. The DSR is used to generate an artifact to address a specific issue and the subsequent examination of its functionality. In this context an artifact is a (technical) solution for an existing problem within a specific environment [27]. Hevner et al. [27] established the concept in 2004 with the objective of driving innovation by developing ideas, methodologies, technical capabilities and tools that enable the effective and efficient analysis, design, implementation and management of information systems. The present methodology was selected for this research with the intention of addressing a real problem and analysing existing solutions (knowledge base). Following the successful design, development and evaluation of the solution, it will be integrated into the existing knowledge base and made available to all. The methodology is ideal for addressing existing challenges in the steel industry, as it is grounded in empirical evidence and a systematic approach.

Building upon Hevner et al. [27], the applied methodology by Peffers et al. [2] comprises six steps (see Figure 1.2).

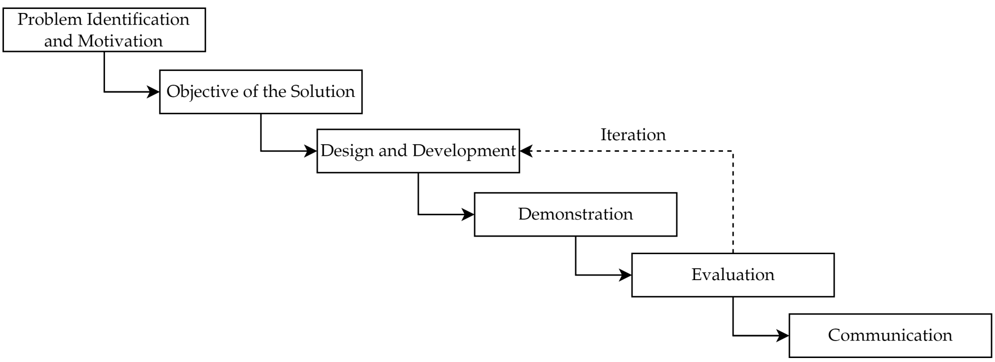


FIGURE 1.2: DSR framework according to Peffers et al. [2]

As applied in this chapter, the first step of the methodology, **Problem Identification and Motivation**, provides a clear argument for the relevance of the research and highlights the specific challenges it seeks to address. This is followed by the description of the **Objectives** and the approach to the **Solution**, which forms the basis for the subsequent implementation steps.

The **Design and Development** phase constitutes the core of the process, where the desired functionality, requirements, and architecture of the solution are first designed. During the development process, the design is then implemented as a solution (artifact) to the identified problem. The fourth step represents the **Demonstration** phase, in which the solution is showcased. This may take the form of an experiment, simulation, or case study, with the latter being applied in this thesis. Building upon this, an **Evaluation** is conducted to observe and measure how effectively the solution addresses the problem. This involves a detailed analysis and a subsequent decision on whether the solution sufficiently meets the requirements and resolves the problem, or whether an iteration of Step three is necessary to improve the design (see dashed line) . If no further iteration is required, the final step, **Communication**, emphasises the importance of disseminating the problem and proposed solution to researchers and other audiences, such as professionals.

The research methodology applied in this thesis follows a problem-centred approach, and therefore the order of the six steps is described in Figure 1.2. However, it is also possible to begin at any other step, should alternative approaches, such as an objective-centred approach (starting from step two) or a development-centred approach (starting from step three), be more appropriate for the specific case.

Thesis outline

The thesis is divided into chapters that contribute to the objective of a multi-modal approach for reuse and automation in steel fabrication. Figure 1.3 shows the structure of the applied DSR methodology, the corresponding chapters and the combined modelling methodology applied in the **Design** and **Development** stage. A short summary of each chapter is given below:

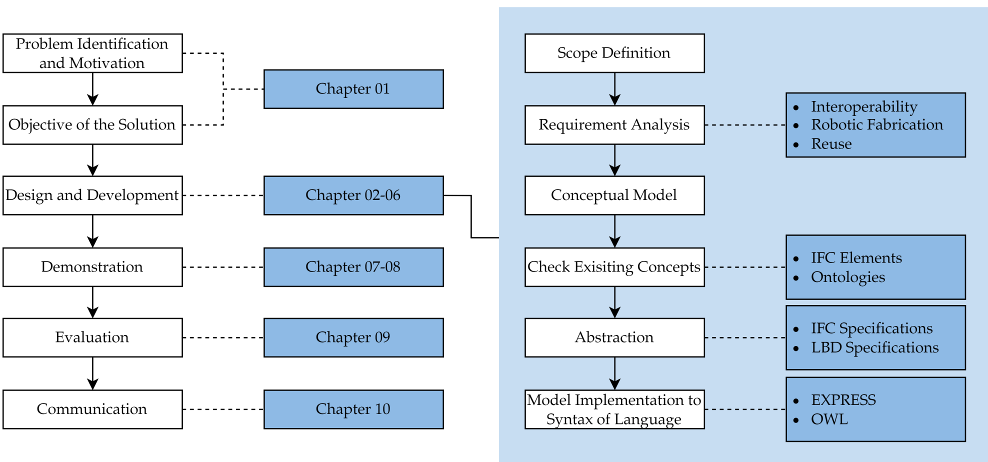


FIGURE 1.3: Overview of the applied DSR methodology, corresponding chapter references and the combined modelling methodology applied in the Design and Development stage

chapter 2 - Theoretical Framework This chapter establishes the basic concepts necessary to understand the research context of this thesis. It introduces key terminology related to data and information, as well as basic principles of data modelling. An overview of existing data and information models in the construction industry is provided, with particular emphasis on the Industry Foundation Classes (IFC) and ontologies within Linked Building Data. Finally, the methodologies used to define an IFC model view and develop a domain ontology are presented. The left side of Figure 1.3 visualises the combination of the individual modelling processes and the respective implementation focus in each step. This forms the basis for subsequent research and development.

chapter 3 - Related Work This chapter examines the current research, development and implementation of key elements relevant to this thesis. It examines the progress and limitations of steel reuse in construction, the application of robotics in steel fabrication, and advances in data modelling for connected steel fabrication.

chapter 4 - Foundations of the Data Model Chapter 4 analyses the key requirements for the data model. Based on the motivation outlined in Chapter 1, the general requirements are defined as a basis for the design of the data model. The requirements analysis is conducted using various methods, including literature and standards reviews, consultations with domain experts, analysis of industrial data, and assessments of the current state of the art. The research is based on input from a diverse group of stakeholders within the steel fabrication value chain (as explained above). In the following chapters, this model is implemented using the appropriate modelling languages.

chapter 5 - IFC Fabrication View Building on the defined foundations and general requirements, this chapter first analyses existing property sets and ongoing efforts to utilise IFC for the exchange of steel fabrication-related information. Based on this analysis, the existing IFC concepts that can be leveraged are identified. Subsequently, new property sets for data exchange in steel construction are defined to comprehensively meet the specified requirements.

chapter 6 - DSTV Ontology In line with the applied methodology for developing a domain ontology and following a similar approach to Chapter 5, this chapter first analyses existing ontologies within Linked Building Data that can be utilised to describe steel fabrication data. Building on this foundation, new classes, object and data properties, as well as their relationships, are defined.

chapter 7 - Demonstration To test the two developed data models and evaluate whether either could serve as a successor to the existing DSTV-NC format, robotic demonstrators are conducted. For this purpose, two exemplary steel fabrication processes, drilling and plasma cutting, are selected, each executed based on both the IFC file and Linked Building Data. The newly defined models are enriched with the latest developments and instantiated with real-world values.

chapter 8 - Results Following the demonstration, this chapter presents the results of the overall development and implementation of the data models, with particular emphasis on their application to robotic execution and data acquisition.

chapter 9 - Evaluation This chapter focuses on evaluating the developed data models for an interoperable construction chain, with particular emphasis on their ability to enable automated steel fabrication and provide a data-driven foundation for potential future deconstruction applications. It compares the two implemented technical approaches to the widely used DSTV-NC format, evaluating their suitability as successors by assessing their ability to support robotic fabrication and enable process feedback. The analysis is guided by defined evaluation criteria, and the results are discussed with regard to key findings, limitations, and implications for broader industry adoption.

chapter 10 - Communication The final step of the DSR methodology focuses on communicating the research problem and the developed solutions. Chapter 10 discusses strategies for integrating these solutions into industrial practice and standardisation efforts, to ensure their relevance and long-term impact.

chapter 11 - Conclusion The conclusion summarises the key findings of this thesis, linking them to the initial research objectives and questions. It reflects on the development and evaluation of the proposed data models and outlines potential directions for future research.

Chapter 2

Theoretical Framework

This chapter provides the essential background and is structured into seven main sections. Section 2.1, 2.2 and 2.3 introduce the fundamental terminology of data and information, along with the basics of data modelling. Section 2.4 presents an overview of current data and information models in construction, with a particular focus on the Industry Foundation Classes (IFC). In Section 2.5, the core principles of Linked Building Data and ontologies are explored. Section 2.6 presents the methodologies applied in defining an IFC Fabrication View and developing a domain ontology. Finally, Section 2.7 concludes the chapter by presenting the evaluation metrics that will be applied in the later stages of the thesis.

2.1 Data and information

The success of construction projects may be contingent upon a number of factors, including the quality and reliability of the data and information being described, shared, and preserved [28, 29]. Due to the complexity and multiple stakeholders and their specifications, data and information can be associated in a variety of forms.

The term 'data' is understood to refer to ordered sets of symbols and/or signals. These symbols may take the form of any letters of an Arabic alphabet, or other characters. Additionally, they can also be represented by pixels in images or geometric symbols. Information is represented by data, which can be interpreted to gain insight into its meaning [30, 31] (see Figure 2.1).

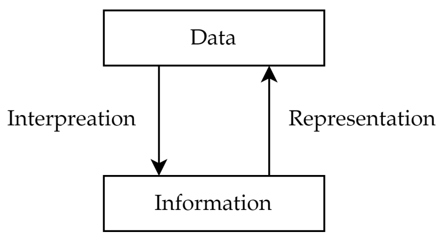


FIGURE 2.1: Relationship between data and information, according to [30]

However, it is also important to note that information is not only derived from data. The concept of information as an interpretation posits that information is the meaning or significance that a receiver (human, system, or organism) ascribes to data based on contextual factors, prior knowledge, and perspective. In this view, information is not merely the raw data itself, but rather emerges through the interpretive process by which the receiver understands, internalises, and derives relevance or actionable insights from that data [32]. This concept emphasises that the interpretation of data is subjective and contingent on the observer's perspective [33]. It also underscores the notion that data, in and of itself, does not possess inherent informational value; rather, it is through interpretation that data acquires its informational significance [34], thereby, underscoring the dynamic relationship between data and information.

An alternative perspective posits that the interpretation of information is influenced by the context or environment in which it is situated. This view holds that the meaning and relevance of data are shaped by the circumstances, perspectives and environments with which they are associated, even in the absence of an explicit communication [35].

The earliest attempts to define information can be traced back to Claude Shannon's work in 1948. In his publication, *A Mathematical Theory of Communication*, Shannon outlines the concept of information as a quantitative, probabilistic entity that serves to reduce uncertainty in communication systems. Shannon's most significant contribution is the concept that information is not inherently tied to meaning or content, but rather to the reduction of uncertainty upon message receipt [36, 37].

2.1.1 Data and information models

Data Models (DM) offer a structured framework for organising and contextualising different data [38]. A data model is the foundation upon which data interpretation is built as Information Model (IM) (see Figure 2.2). It defines and structures the raw data elements clearly, such as attributes and the structures of the system, situation or reality to be modelled. Data Models define managed objects at a lower level of abstraction, whereas the main purpose of an Information Model is to model managed objects at a conceptual level, independent of any specific implementations or protocols used to transport the data [39]. It entails defining the relationships and interpretations essential for data to become actionable information that users (human or machine) can comprehend and utilise effectively.

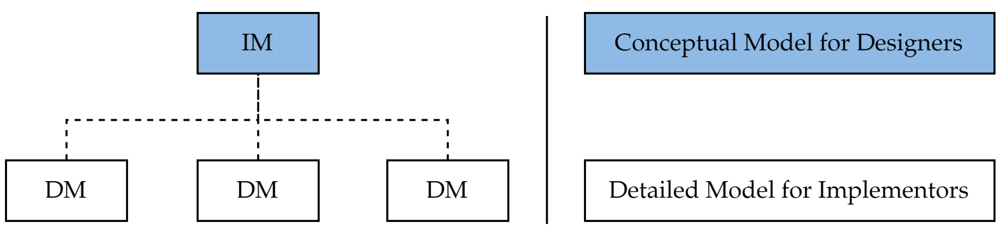


FIGURE 2.2: Relationship between DM and IM, according to [39]

While IM and DM serve different purposes, it is not always feasible to definitively determine which particular details should be conveyed in an IM and which should be included in a DM. It is possible that different literature sources may offer differing interpretations and assignments. In this paper, the term 'Data Model' is employed for the purpose of representing the required elements of steel fabrication processes.

2.2 Basis of data modelling

The process of data modelling can be defined as the dynamic creation, refinement and management of data models that accurately represent the elements and relationships present within a given domain [40].

In the field of digital modelling, it is common to differentiate between various types of models. This category encompasses, for instance, 'reproduction models', which replicate existing reality (e.g., digital terrain models, anatomical models or economic activity models). Additionally, there are 'prototype models', which virtually represent something that does not yet exist in reality, and 'digital models/digital building models', which are typically employed to facilitate computer-aided design (CAD) in engineering and construction. Hybrid models also include application models and process models [41]. The model developed within this thesis is referred to as 'digital model'.

The workflow for data modelling process is structured into two sequential stages: conceptualisation and realisation/implementation [42] (see Figure 2.3). In the initial phase, the desired reality is abstracted in a model, whereby significant items and properties are represented, and relationships between them are established. Subsequently, the conceptual model is applied to a specific use case in the form of actual data, stored in a physical file / database [41].

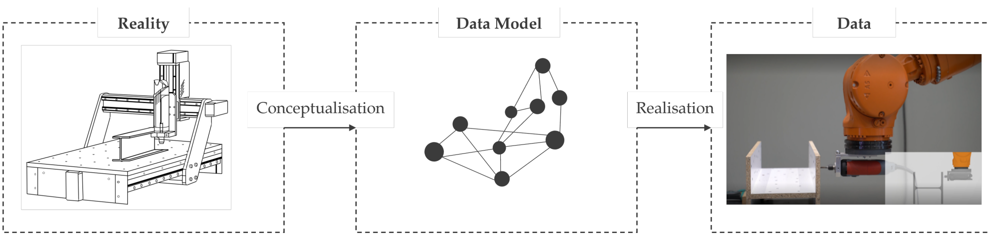


FIGURE 2.3: Abstract data modelling process

The present work is characterised by the fact that the reality to be depicted is not yet fully established in some areas of the steel industry. This applies in particular to the data that has to be modelled for robotic steel production and for deconstruction. The first phase therefore includes not only a conceptualisation of existing reality, but also of future reality.

The first phase of the workflow can be broken down into a more detailed process by applying the modelling process by Lee et al. [3] for an IM to a DM (see Figure 2.4). This five-step approach was originally developed for defining a machine shop information model in manufacturing. However, it can also be applied to other domains. The preliminary stage of developing a model entails the delineation of the model's scope of applicability, including a collection of processes, information and constraints to be abstracted. The **Scope Definition** is followed by a **Requirement Analysis**, which can be done by literature analysis, domain experts interviews or state-of the art assessment. The data requirements are consequently transformed into a **Conceptual Model**. Subsequent to the previously undertaken definition of the structure of the model and the data to be integrated into it, the model is implemented in accordance with the notation and language chosen for this purpose.

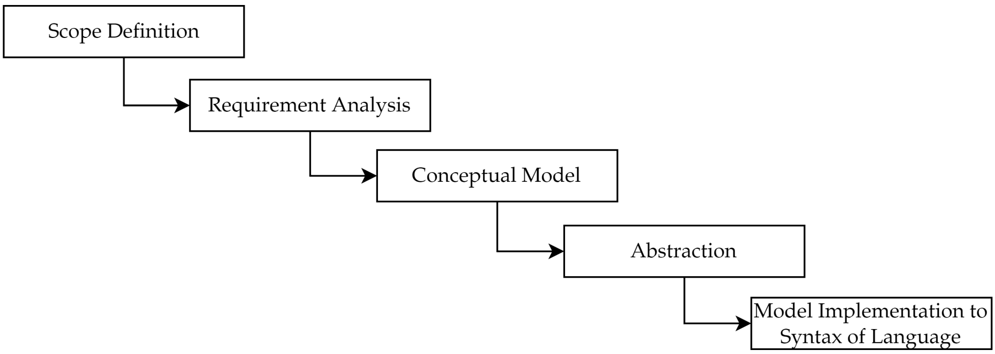


FIGURE 2.4: Data modelling process according to Lee et al. [3]

2.2.1 Modelling methodologies

In general, there are different methodologies for modelling. The prominent ones are Entity-Relationship (ER) and Object-Oriented (OO) [43].

The Entity-Relationship (Diagram) (ERD) is recognised as a fundamental tool in the field of data modelling [44]. The Entity-Relation is the general framework for describing data, and the ERD is the visual representation of the structure. First introduced by Chen 1976 [45], the model emphasis particular on relationships between various components and can be seen as a basis for a unified view of data. It aims to create a simple and intuitive way of modelling real data structures that is understandable for both users and developers.

The model's main concept focuses on Entities, Relationships, and Attributes. An 'entity' represents a type or class of an object. It is mainly used to visualise the structure of data in a database and does not explain individuals directly. Furthermore, there can be 'entity sets', where single entities can be a subset from (e.g., *Oak* can be a sub entity of the entity set *Tree*). In contrast, the term 'relationship' refers to an association or connection between entities. For instance, a 'person-to-person' relationship may be defined as an association between two individuals, whereas a 'company-to-employer' relationship can be understood as an association between a company and its employer. Relationship sets can also appear here as general relationships.

Furthermore, the concept defines 'attributes, value and value set' as properties or characteristics that describe entities and relationships (for example, a person's name or an order date) [45]. This overarching concept is applied in the modelling process later in the thesis.

The object-oriented modelling (OOM) was developed during the 1970s with the objective of facilitating the creation of complex software systems in a straightforward and extensible manner. The fundamental concept of object-oriented modelling is predicated on the notion of perceiving the world as an aggregation of distinguishable, identifiable and individually describable objects. This encompasses both simple and complex objects. Furthermore, objects are described by both static, structural features, known as properties or attributes, and dynamic, behaviour-related features, known as operations or methods [46].

One way to graphically represent OOM is using Unified Modelling Language (UML) [47], which has gained acceptance in building modelling and provides the base structure for the EXPRESS modelling language. EXPRESS is used to implement the IFC data schema for representing building and construction information.

2.3 Data modelling languages

Various modelling languages have already been developed. In general, a distinction is made between two different approaches: graphical form and text-based form. UML and XML are presented as standardised modelling languages, as well as EXPRESS for construction specific applications. The specifications of Linked Data are explained in Section 2.5.2.

2.3.1 Unified Modelling Language

The Unified Modelling Language (UML) is a standardised language (ISO/IEC 19505) that employs texts and symbols to graphically represent various aspects of a system. The language was initially developed by Grady Booch, James Rumbaugh and Ivar Jacobson and has since become an industry standard, employed by various stakeholders involved in the software development area [48].

UML provides support for a variety of diagram types (e.g., class diagrams, activity diagrams, and sequence diagrams), thereby enabling the provision of tailored representations for the various aspects of steel fabrication processes. For instance, class diagrams can be utilised to model material properties, while sequence diagrams can be employed to describe workflows in fabrication.

The language's deficiencies can be attributed to two main factors: its increasing complexity and the substantial size of its diagrams, which frequently leads to an increase in word count. For individuals lacking expertise in software design, the language's formal nature at the architectural level can be daunting and is often perceived as 'too complicated' [49].

2.3.2 Extensible Markup Language (XML)

In contrast to UML the Extensible Markup Language (XML) is a text-based language, standardised by the World Wide Web Consortium (W3C, www.w3.org). XML can be employed for two purposes: firstly, to define a data model (an XML schema file) and secondly, to store the actual data (an XML data file) [41]. Furthermore, the documents are both human- and machine-readable.

Unlike previous languages, XML schema can represent ordered relationships. A primary element is the 'element-sub element' relationship [50]. Furthermore, XML is a highly flexible data exchange language that allows developers to define custom tags that suit specific data structures, which can be used to represent structures within the steel fabrication.

The XML does not provide native data type support, such as numbers, booleans, or arrays; all data is treated as a string. It is only possible to do this in combination with XSD structural and typed validation [51]. Consequently, developers frequently need to incorporate additional logic to facilitate data type conversions, which might be required for steel fabrication. To illustrate this point, consider the question of whether a steel part is a single part or the main part where single parts are assembled to. In such cases, a Boolean answer is sufficient.

2.3.3 EXPRESS

In addition to the data modelling approaches that had been developed in other industries, the construction industry was also engaged in discussions towards the end of the 1980s on how to implement a higher-quality data exchange. The approach adopted was centred around product data modelling, with the objective of describing the product both geometrically and semantically [52]. The Standard for the Exchange of product model data (STEP) was then published as part of ISO 10303 [53]. The international standard aims to provide a mechanism for describing product data throughout the entire life cycle of a product independently of a specific system. Due to the nature of this description, the standard is not only suitable for neutral file exchange, but also as a basis for the implementation and sharing of product databases and archiving. EXPRESS was published as a modelling language as part of the publication of ISO 10303 Part 11 and describes the structure, relationships, constraints and rules of a data model.

Based on the various parts of the ISO standard, specific product and data exchange scenarios for the respective industries were defined in the form of 'application protocols' (AP). One such protocol was AP 225, 'Building elements with explicit representation of component geometry'. However, the participants were unable to quickly reach a consensus on its definition. As a result, in 1994, the 'Industry Alliance for Interoperability (IAI)' was founded under the leadership of Autodesk. The organisation, which was renamed 'buildingSMART' in 2005, aimed to accelerate the development of standards for the construction industry. A first version of a standardised modelling approach for construction-related data was then published in 1997 as Industry Foundation Classes (IFC) [54]. IFC is based on the EXPRESS modelling language and uses it to define the structure and relationships between the various elements of a building and infrastructure.

Main aspects of the EXPRESS language are: classes to describe entities and their structure, attributes to display properties or characteristics of those entities, and relationships to define how entities are connected or interact with each other. Building upon the OOM concepts for inheritance and sub-types are implemented to represent hierarchical structures. It is important to note that EXPRESS is used to define a data model. It is not feasible to utilise EXPRESS to describe particular instances of the data model. However, there are a number of options available for this purpose, including STEP Physical File, XML-instances, JSON string or RDF [54].

2.4 Existing exchange formats

Following the introduction of individual data modelling methods and languages, this section shifts the focus to exchange formats. While modelling languages are used to formally define data structures, relationships, and rules, thus establishing the underlying information logic, exchange formats contain concrete data instances and enable the structured exchange of this information between heterogeneous systems.

2.4.1 DSTV-NC

Within the field of steel fabrication, the DSTV-NC standard (Deutscher Stahlbau-Verband – Numerical Control) defines a standardised interface between design, production planning, and machine control. Developed by bauforumstahl e.V., it facilitates data exchange between computer-aided design and manufacturing (CAD/CAM) systems and numerical control (NC) machinery. The standard has seen widespread international adoption and is primarily used for the fabrication and processing of individual steel components, such as beams and plates [16, 17]. Although a release extending the standard to include assembly-related information has been published, its adoption within the industry remains limited [55]. The format's main function lies in the provision of geometric manufacturing data, while the ability to convey detailed machining or robotic processing instructions is comparatively limited.

Despite these constraints, one of the key advantages of DSTV-NC is its ability to standardise data exchange across heterogeneous software and machine systems, thereby reducing errors and enhancing interoperability along the supply chain. The file format is designed to be simple and robust, allowing reliable communication between planning tools and fabrication machinery. Originally introduced as a plain text ASCII format, an extended XML-based version was later released [17]. The DSTV-NC file is structured hierarchically around a central workpiece object, which contains various elements and attributes—some mandatory, others optional.

The file content is organised into clearly defined sections, each following a specific syntax and semantic meaning. These include:

- **Header Data:** General information such as project name, part number, and creation date.
- **Geometry Data:** Specifications of the part's physical dimensions and shape (e.g., length, width, thickness).

- **Operation Instructions:** Details on fabrication actions such as drilling, cutting, and marking, including coordinates and tool parameters.

The DSTV-NC standard also defines a consistent coordinate system ('NP') and a set of fabrication views (Bottom ('U'), Top ('O'), Front ('V'), Rear ('H')) to specify machining locations and reference geometries (see Figure 2.5). Additionally, features can be aligned to specific reference lines, such as the top edge, bottom edge, or a central symmetry line.

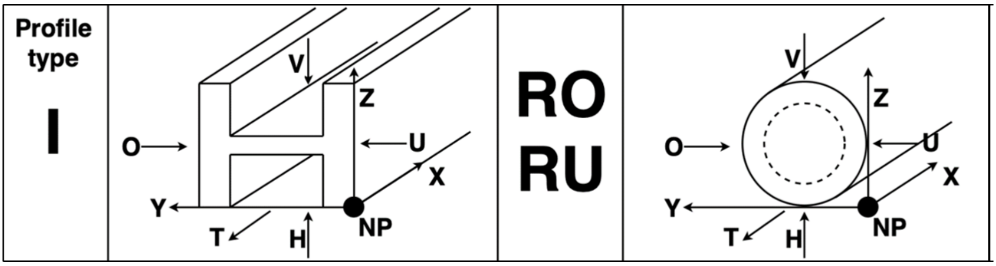


FIGURE 2.5: DSTV reference structure for profiles according to [17]

In this thesis, the DSTV-NC format serves as the foundation for the proposed data modelling approach and is extended to meet the requirements of the defined use cases. The structure of the standard is illustrated and explained using the selected case study beam (see Chapter 4).

2.4.2 Building Information Modelling

Building Information Modelling (BIM) builds upon earlier data exchange approaches by providing a comprehensive, interoperable digital representation of the entire building lifecycle. Prior, the construction industry was largely reliant on two-dimensional (2D) drawings, which were created using computer-aided design (CAD) software [56]. Such drawings represented a single, static view of the construction project. This approach frequently gave rise to difficulties when the intricacies of a complex structure were to be conveyed in a single two-dimensional representation [57]. In order to address this challenge, civil engineers would frequently create three distinct views of a single building element, with the objective of facilitating a more comprehensive understanding of its construction process [58]. However, this approach often resulted in confusion and misinterpretation due to the lack of a consistent digital representation. It also led to delays and the loss of productive days on site [59]. In response to these challenges, BIM was introduced as a transformative technology that has had a significant impact on the construction industry. It represents a digital approach to the design, construction, and operation of buildings and infrastructure, based on the continuous creation and use of integrated data and information models [60]. BIM is based on the principles of data modelling, whereby a central, accessible, and reusable repository of information is created that represents not only the static elements of a building but also dynamic processes, functional relationships, and interdependencies [61]. This approach permits a more comprehensive view of a building's lifecycle, connecting disparate stages (such as design, construction, operation, and even deconstruction) to guarantee a consistent data flow and well-informed decision-making across all stakeholders [62].

BIM is widely accepted within various projects of the construction industry [63]. Despite the existence of national initiatives such as 'BIM Deutschland', a federal government initiative that serves as the central public point of contact for BIM-related information and activities (as stated in 'Nationales Zentrum für die Digitalisierung des Bauwesens (BIM) – BIM Deutschland', [64]), various challenges persist [65].

The prevailing challenges pertaining to BIM can be categorised as follows: the standardisation of information models across a range of tools and software, ensuring data interoperability, and the management of the complexity inherent in modelling interdependent processes. openBIM can be defined as an approach to ensure that the design, realisation and operation of a building is performed in accordance with open standards and workflows. In order to facilitate interoperability among BIM software applications, Industry Foundation Classes have been defined [66].

Although Building Information Modelling was designed to create a comprehensive database for capturing data on a building and its individual components and processes throughout its entire life cycle, in practice it has so far been used mainly in the planning phase [67]. The implementation of the BIM model in pre-fabrication, operation or even the deconstruction phase has so far been limited and is only just being investigated [68–70]. However, the capacity of BIM to collect extensive data about a structure throughout its entire lifespan positions it as a suitable tool for enhancing the principles of circular economy [71]. The subject of this thesis is therefore the investigation of the requirements for the use of the BIM model in steel prefabrication from the point of view of data generation for deconstruction.

2.4.3 Level of Information Need

Despite the clear benefits of BIM in terms of its capacity to digitally represent a building throughout its entire lifecycle, it should be noted that the level of modelling and the requirements for the exchange of information between stakeholders vary across the different phases of construction [72]. Accordingly, [73] provides a comprehensive framework for understanding the Level of Information Need (LOIN) across various life cycle stages of a construction project, in the context of the utilisation of BIM. The depth of information requirements are described by the following concepts: Geometric information, alphanumeric information and documentation. Complementary, in other publications, the maturity of elements in a BIM model, related to requirements of information in different stages of the construction project, are categorised into five different stages (Level of Development (LOD) 100 - LOD 500), representing information on a low level to a high level [74].

2.4.4 Industry Foundation Classes

Building upon the comprehensive digital integration enabled by BIM, the Industry Foundation Classes (IFC) provide an open, vendor-neutral, and standardised data model (ISO 16739 [75] by buildingSMART) that facilitates consistent, interoperable information exchange and integration of digital building models across the architecture, engineering, and construction domains. As described in Section 2.3.3, the IFC class model corresponds to the STEP standard.

Until now, the format was mainly used in the 2x3 version [54], but since January 2024, the new 4x3 version has also been standardised in the ISO [76], so this paper will refer to this version.

A significant benefit of IFC is its utilisation of object- and relation-oriented modelling. This encompasses the identification of objects and the definition of independent relationships [77]. Nevertheless, certain disadvantages have been observed in the context of data exchange between disparate software systems, giving rise to the absence or inaccuracy of designated classes [78] and the lack of specifications for stakeholder requirements subsequent to the design phase, a relevant example being steel fabrication [59]. Moreover, the integration of IFC models can present certain challenges and may impede the effective participation of subsequent stakeholders. This is due to the fact that these stakeholders may only require information relevant to their specific area of expertise, such as steel, rather than a comprehensive range of data relevant to bathrooms or interior design.

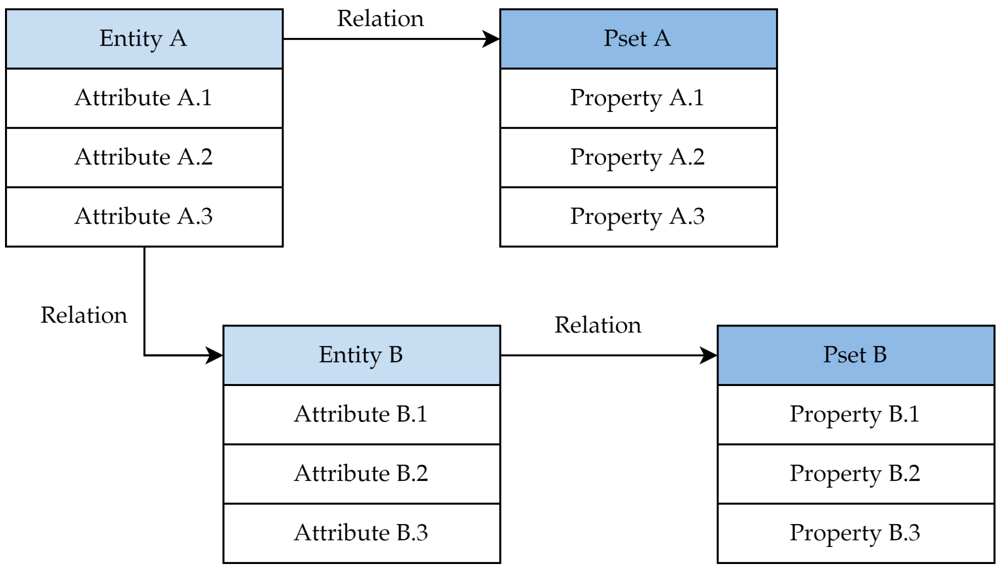


FIGURE 2.6: Basic of IFC key elements

The IFC structural framework is organised hierarchically, starting with general foundational classes such as IfcRoot and branching into more specialised subtypes. The structure is composed of various key elements (see Figure 2.6), with entities representing real-world building elements or conceptual objects such as *IfcBeam*, *IfcColumn*, or *IfcPlate*. Each entity is characterised by a set of predefined attributes that provide a comprehensive description of its fundamental characteristics. These attributes can encompass a unique identifier (*GlobalId*), a designated name (*IfcLabel*), and an additional identifier (*IfcIdentifier*). To illustrate, an *IfcBeam* representing a steel beam in a structural frame would possess an *IfcLabel* that denotes its human-readable name, for example, 'Main Beam'. In addition, it would have an *IfcIdentifier* such as 'B-2025' that facilitates the effective tracking of the beam within the project's documentation. Beyond attributes, IFC introduces property sets (*Psets*), which provide a more flexible means of defining additional information about an entity. These property sets group related properties together, thus allowing for extended descriptions without modifying the core entity structure.

In order to establish meaningful connections between elements, IFC employs relationships that define how different entities interact. For example, a steel beam (*IfcBeam*) might be linked to its spatial hierarchy using *IfcRelContainedInSpatialStructure*, thus placing it within a specific IfcBuildingStorey. Furthermore, in the event that the beam contains holes, it can be associated with an *IfcVoidingElement* via *IfcRelVoidsElement*.

2.4.5 Model View Definition

In order to address the problem of rich and redundant IFC models, the Information Delivery Manual (IDM) (ISO/DIS 29481 [79]) and Model View Definition (MVD) [80] concepts have been introduced [81]. The aim of introducing these concepts is to reduce the scope of the full IFC model to particular model views, to meet the data exchange requirements for particular use cases or workflows in construction and building processes [82]. Figure 2.7 displays the differentiation between the different concepts.

IDM describes the processes and information requirements within a BIM workflow. It defines what information is needed at a specific point in the project and how this information should be exchanged [83].

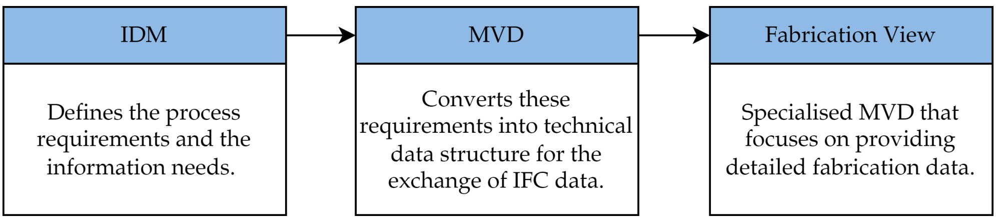


FIGURE 2.7: Differentiation between IDM, MVD, and Fabrication View

The development of an IDM is characterised by the fact that the exchange requirements for the area are defined by technical experts and by selecting specific use cases using the buildingSMART Information Delivery Standard (IDS) [84]. Chapter 4 will explain the use cases of the steel fabrication which are used to define the requirements of the model. Nevertheless, the definition of Model Views is a challenging process. In addition to users defining the exchange requirements, the technically correct modelling according to the IFC schema must also be adhered to. Otherwise, it is not possible to generate semantically precise, reusable and consistent models views which can be integrate with the overall model. The implementation of these requirements is facilitated by the MVD through the utilisation of a technical data structure that enables the exchange of IFC data. The Fabrication View is a specialised MVD that provides stable models to facilitate visual coordination and interoperability for a specific fabrication of a domain. Zhang et al. [82] discuss different methods for mapping exchange requirements to IFC schema to generate the Model Views. Besides the ones that are based on taking the EXPRESS file as a sub-schema of the IFC model (ifcDoc, xPPM), other focus on an IFC independent method, integrating the central model repository [85]. This framework for MDV development and implementation focuses on the use of ontology language as part of Linked Data to overcome the challenge of correct mapping, thereby facilitating interoperability.

2.5 Linked Data and Semantic Web

In addition to established exchange formats that rely on predefined data schemas, approaches based on Linked Data introduce semantic layers that enable dynamic, extensible, and cross-disciplinary information integration.

The concept of 'Linked Data', as it is known today, was initially developed by Tim Berners-Lee and Robert Cailliau at CERN in 1990 [86] and popularised in 2006. It's purpose was to enhance data sharing among scientists by allowing information to be referenced independently of its physical location [87]. This approach, which is rooted in the foundations of the World Wide Web (WWW), ultimately evolved into the Semantic Web and Linked Data.

Linked Data is grounded in four fundamental principles [88]: resources are identified using Uniform Resource Identifiers (URIs); HTTP URIs are used so that these resources can be looked up over the web; when a URI is accessed, it should return useful, structured information; and this information should include links to other URIs to enable discovery of related resources. At present, the WWW principally comprises human-readable documents that are interconnected by hyperlinks. The Semantic Web aims to restructure the information within these documents by utilising Linked Data technological standards, thereby enabling machine-readability [89]. These address issues of fragmented information, incompatible formats, and the need for dynamic information sharing as circumstances change [90].

The concepts of Linked Data and Semantic Web have been already applied in other domains such as geography, biology or healthcare [91]. In the construction industry, where numerous stakeholders operate across disparate systems with continuous data updates, the principles of Linked Building Data (LBD) provide a solution to facilitate cross-platform information sharing [92]. The objective of the Linked Building Data-Community Group (LBDCG) is the development of modular vocabularies that address specific building-related topics [93]. This approach foster the flexibility to model and address specific construction topics [94].

2.5.1 Ontologies

A fundamental aspect of Linked Data is the utilisation of ontologies, which are standardised, formal representations of concepts, their attributes and relationships of a specific domain [6]. This concept was first proposed by Gruber in 1993 [95]. Ontologies facilitate the integration of disparate data and information models by formalising the concepts and relationships that underpin contextual understanding. Ontologies provide the framework as a unified description and can be formed with real instances to provide a knowledge base and real data [96]. Ontologies are fundamentally built from semantic triples, which consist of three components: a subject, a predicate, and an object (see Figure 2.8). In this structure, the subject and object are linked by the predicate, representing a relationship between them. This mirrors the way humans typically use language to convey statements and share information.

FIGURE 2.8: Concept of semantic triplets

Ontologies can also be referred to as vocabularies, with a view to facilitating a shared understanding of the objects, properties and relationships described between different human actors, as well as between humans and machines [97].

In the context of Linked Data, ontologies can be categorised into distinct layers, thereby emphasising their inherent modularity. The potential for interconnecting ontologies across different levels is highlighted [4]. Figure 2.9 displays the different level of ontologies. In the context of this thesis the establishment of a *Domain Ontology* will be undertaken. As steel fabrication constitutes a sub-industry of the overall construction value chain, specific requirements may be expressed in a domain ontology, which can be linked to other levels of existing ontology. To illustrate this, consider a beam that is to be assembled. This is not merely part of the overall design of the building; it is also described in greater detail with regard to its fabrication prior to installation on site. It can thus be concluded that the domain ontology for steel fabrication can be linked to other mid-level or top-level ontologies.

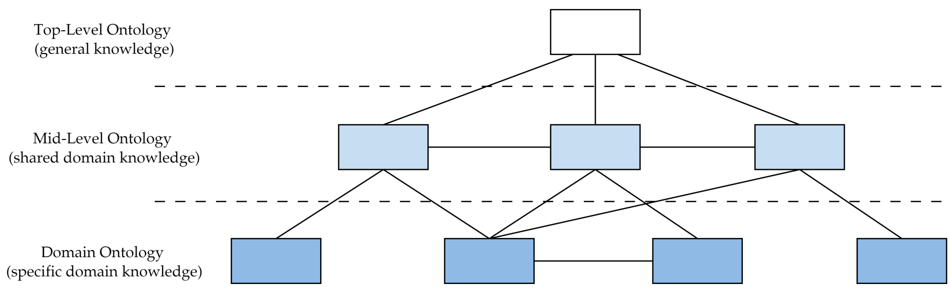


FIGURE 2.9: Levels of ontologies, according to Amirhosseini et al. [4]

Ontologies are of particular value in the context of construction, as they provide a structured and shared vocabulary that improves consistency and interoperability across a range of tools and teams. The construction sector can achieve more coherent data integration by adopting domain-specific ontologies, which ensure that complex processes and project elements are uniformly described and accessible across platforms and stakeholders [98]. The standardised modelling of information not only supports collaboration but also facilitates a structured approach to the management of data throughout the lifecycle of a project, from the initial planning stages through to deconstruction and reuse [99].

The following example provides a graphical representation of the fundamental structure of a simple ontology. Figure 2.10 illustrates the core components of an ontology, which primarily consist of classes and their hierarchical relationships. These relationships are established through subclassing (*Subclass of*), enabling inheritance between classes. Each class is characterised by specific attributes, which can be categorised into object properties and datatype properties. Object properties define relationships between two classes, whereas datatype properties establish connections between a class and a literal. A literal represents a concrete value that can be further classified according to its data type, such as text, date, or numerical value.

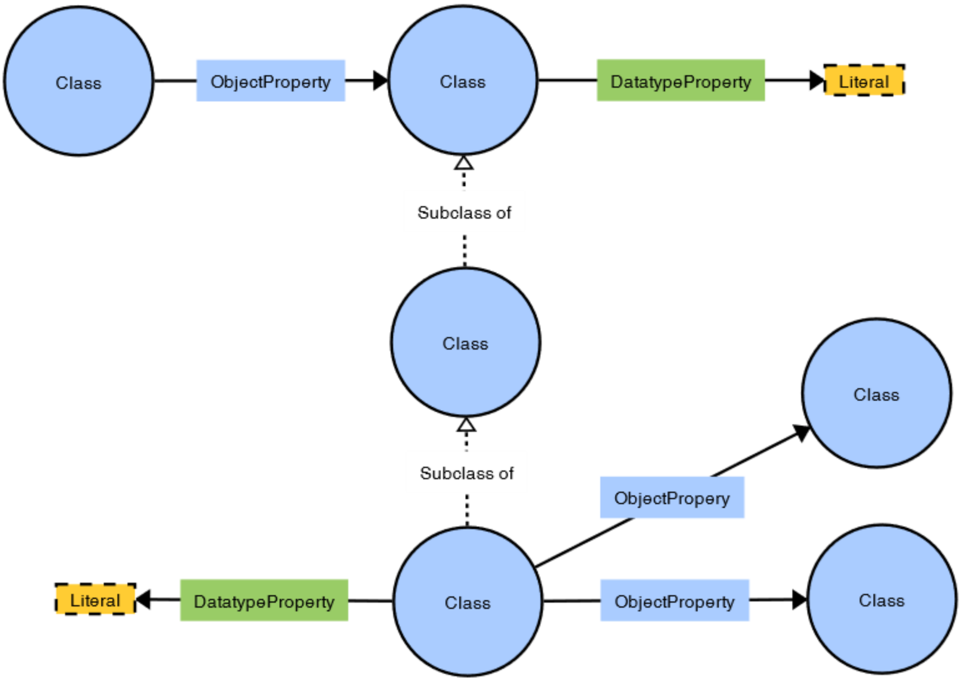


FIGURE 2.10: Basic elements of an ontology including classes, class hierarchies and properties

The basic structure of the ontology defined above can be filled with knowledge about the subject area under consideration (see Figure 2.11). The example chosen here is the concept of winter sports. *Athlete*, *Coach* or *SportsGear* can serve as classes, while *WinterSportsman* is a subclass of *Athlete* and *DownHillRacer* is a subclass of *WinterSportsman*. Classes can also be related through object properties. In addition, data type properties assign specific values to the class, such as *Name* or *SpeedRecord*.

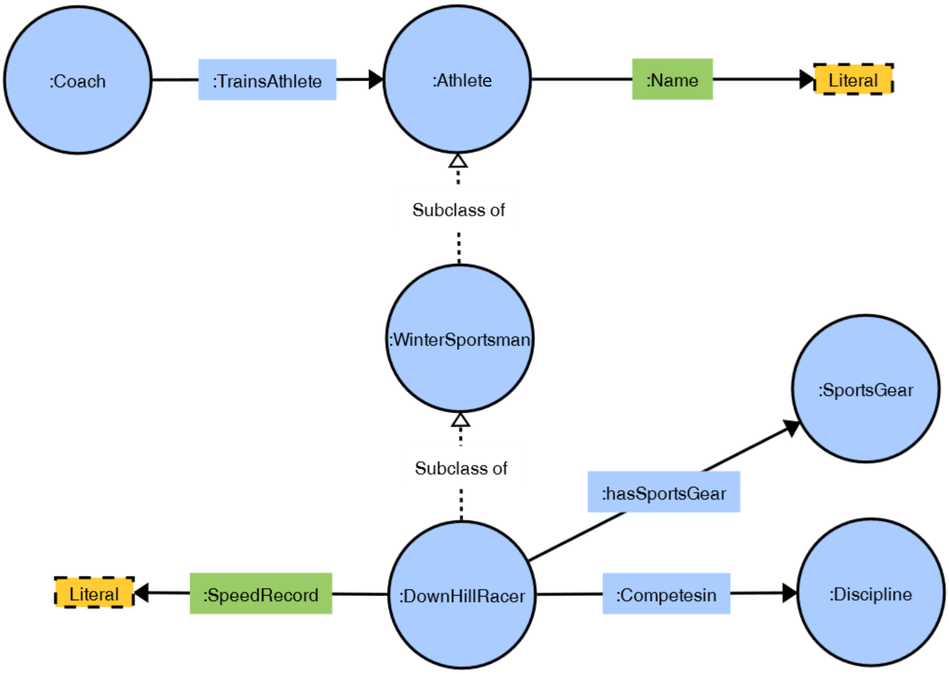


FIGURE 2.11: Example of an ontology, applied to the basic structure in Figure 2.10

Both graphics were created using Visual Notation for OWL Ontologies (VOWL) [100].

2.5.2 OWL and RDF

As with other data models, there are also standardised languages specific to ontologies and published by the W3C. The main description languages are Resource Description Framework (RDF) [101], Resource Description Framework Schema (RDFS) [102], Web Ontology Language (OWL) [103] and Unified Modelling Language (UML).

RDF, a framework for representing information on the web, was developed by the W3C. Its fundamental concepts are predicated on the abstraction of reality in triplets as logical statement (see Figure 2.8) and it contains the basic structure for describing ontologies such as classes or properties [101]. Each concept of the triplet has a Unique Resource Identifier (URI) [104] to label it explicitly [105]. RDFS provides additional definitions, whereas OWL is another extension of the RDF(S) [18]. By providing a differentiation between object and data properties or the introduction of the ’sameAs’ relationship OWL enables a greater and more precise expression the data [89].

The RDF statement can be represented with different syntaxes: RDF / XML (*.RDF*), N-Triples (*.N_T*), Turtle (*.TTL* - [106]) and Notations-3 (*.N3*- [107]). These representations enable the storage and querying of RDF statements in specialised databases known as *triple stores* [61, 97, 108]. Within this thesis, GraphDB is used [109].

2.5.3 IFC and OWL

Despite the two different approaches to the specification and standardisation of IFC for the construction industry and RDF for web data, there are similarities and developments to integrate these techniques [105]. Although the structure of RDF and EXPRESS can be compared to a certain extent, there are limits to expressing all relevant semantics with EXPRESS [110]. Therefore, the ifcOWL ontology was developed to represent building data in terms of the Semantic Web and Linked Building Data [111]. This development built upon the given EXPRESS structure and is standardised by buildingSMART [105, 112]. The development of a converter from IFC EXPRESS data to ifcOWL [105], an alignment from ifcOWL to LBD [113] and for converting IFC instances to RDF graphs [114] facilitates the mapping of corresponding elements and ensures the correct representation in OWL [111, 115].

The developed framework facilitates the integration of construction data with other data types, including sensor data and material properties. The present work is founded upon the extant definitions and concepts inherent within the framework of IFC and ifcOWL, with a view to mapping the requirements from steel fabrication.

2.6 Methodologies for Model View & domain ontology

Based on the generally described methodology in Section 2.2 for a modelling process, there are specific guidelines for the definition of a Model View and for the development of ontologies. The methodology employed for each data representation is explained in turn in the following section. This methodology will be applied in the development stage of the overarching DSR methodology (see Chapter 5 and Chapter 6).

2.6.1 Development of a Model View

The standard procedure for delivering IFC based data exchange have been published by [5, 116–118]. The preliminary step of the development is defining the **Information Delivery Manual (IDM)** (see Figure 2.12). It consists of the sub-steps **Process Maps**, which defines the industry process and **Exchange Requirements**, which defines the information to be exchanged. The requirements are then organised into structured exchange requirement models and subsequently linked to specific IFC entities, attributes, and types, forming the basis of the **Model View Definitions (MVD)**. Once MVDs have been documented and published, it is imperative to provide support for vendor **IFC Implementation** in their software product as well as **Data Validation** within real projects.

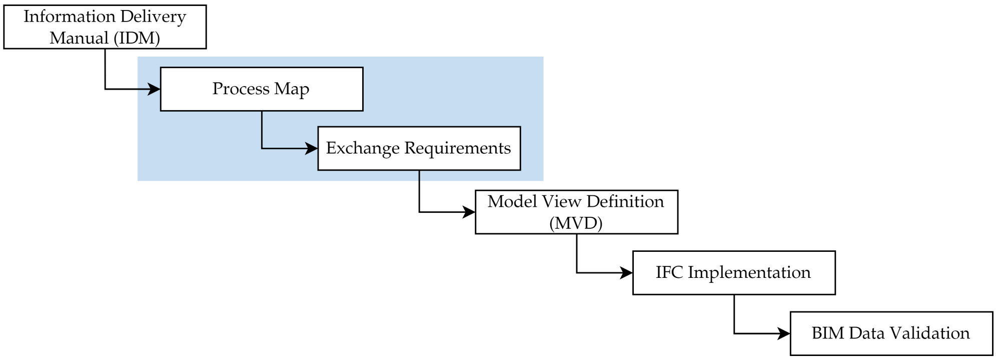


FIGURE 2.12: Steps for reaching deployment of IFC based solutions according to Hietanen et al. [5]

Available tools

Providing useful tools for data modelling ensures a standardised approach and consistent results for widespread implementation [119]. buildingSMART has released several servicees and standards for the different steps of the methodology.

The objective of the ISO 19650 standard is to establish standardised processes for information management throughout the entire life cycle of a building [120]. The most significant principles include the clear definition of information requirements, the structured exchange of information, the unambiguous assignment of responsibilities in information management and the assurance of information quality. The ISO 19650-1 provides a foundation of theoretical knowledge, addressing the concepts of information requirements, information environment, and roles in information management [121]. Additionally, ISO 19650-2 centres on the implementation of these concepts during the planning and construction phases, offering methodologies for the exchange of information requirements and the facilitation of information provision processes [122].

The Use Case Management (UCM) is a platform for systematically defining, managing and sharing BIM use cases [123]. For providing the exchange requirement referenced to IFC, the Information Delivery Specification (IDS) format is provided [84]. buildingSMART Data Dictionary (bSDD) is a collection of interlinked data dictionaries containing definitions of terms describing the built environment [124]. It ensures consistent and accurate definitions of terms, properties for different domains within construction and serves as a library for referencing shared definitions [125]. The primary objective is to distribute the established specifications for steel fabrication in the bSDD, making them universally available for use in modelling steel elements and processes in accordance with established standards.

The present work focuses on development of a specialised MVD for steel fabrication, further referenced as ’Steel Fabrication View’.

Evaluation

The efforts of the MVD and BIM standard are centred on the development of domain-specific concepts and requirements. In order to ensure efficient application, these documents must also undergo validation to ensure interoperability and implementation [126].

mvdXML, released by buildingSMART as an open standard for defining model subsets and validation rule sets [127], enables the validation of an IFC Model View by specifying rules and constraints for required entities, attributes, and relationships. The latest update was released in 2021, but no further releases or specifications for steel fabrication have been published since then [128]. Zhang et al. [126] proposal entails the implementation of an open-source model-view checker, predicated on open standards, for the purpose of validating IFC building models.

In general, the Fabrication View defined here is modelled on the basis of the 4x3 specification to ensure compliance with the IFC standard. In addition, the demonstrators are used to verify that the Fabrication View fully and consistently meets the requirements. A specific mvdXml is not part of this thesis.

2.6.2 Development of an ontology

There exist a variety of established procedures and guidelines with regard to the creation, refinement, and updating of ontologies. A simple approach is described by Lopez et al. [129], consisting of three main phases: 'Specification', 'Conceptualisation' and 'Implementation', being aligned with the general process for data modelling [42]. The methodology delineates the sequence of actions to be undertaken in the execution of each activity, the techniques to be employed, the products to be generated, and the manner in which ontologies are to be evaluated. Staab et al. [130] proceed to present a more detailed methodology for developing an ontology, building upon the previous research. In addition to the technical aspects of the development process, the *Handbook of Ontologies* advises that a feasibility study should be conducted at the outset, with consideration given to organisational and economic factors. This is to be done not only at the initial set-up stage, but also during the evaluation and refinement phase.

The underlying method of developing the ontology within this thesis is *Ontology Development 101: A Guide to Creating Your First Ontology* by Noy and McGuinness [6]. It provides a detailed and developer focused procedure, consisting of 7 steps (see Figure 2.13).

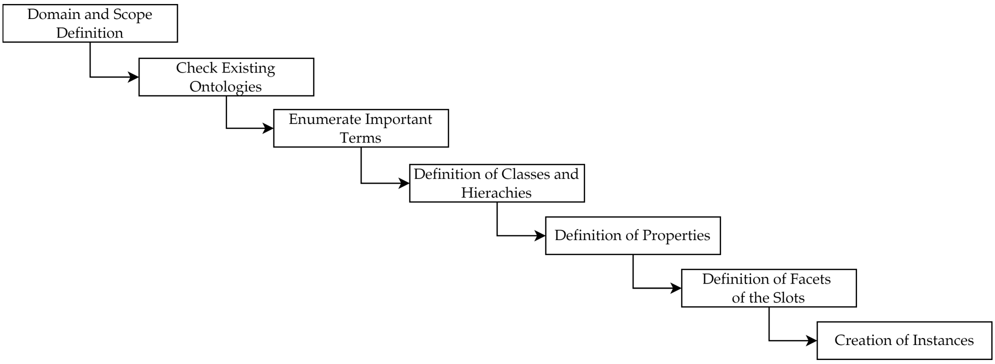


FIGURE 2.13: Steps of ontology development according to Noy et al. [6]

Within the first step, it is suggested to answer key questions to define the domain and scope of the ontology. The questions may change throughout the design-process but provide a framework for the development.

- What is the domain that the ontology should cover?
- For what purpose is the ontology going to be used?
- What kind of questions should the ontology be able to answer?

An important part of the **Scope Definition** are the so-called *Competency Questions*. These questions represent the questions that the ontology should be able to answer and define the requirements from the user's perspective as set in the upfront requirements analysis [89]. Consequently, these inquiries can be employed at the conclusion of the process to assess the development accordingly.

The subsequent stages of the development process, which entail the consideration of **Extant Ontologies**, the identification of **Important Terms** to be represented, and the **Definition of Classes and Properties**, will be elaborated upon in Chapter 6.

Evaluation of ontologies

Evaluation constitutes a pivotal component within the domain of ontology development. A variety of publications have provided definitions of methodologies and criteria that can be utilised for this purpose [131, 132]. The primary motivation is twofold: firstly, to ensure the technical accuracy of the defined ontologies, and secondly, to facilitate their practical application, updating and maintenance [130].

In this thesis, the evaluation criteria published by Gómez-Pérez [133] are applied. These consist of eight criteria:

1. Correctness
2. Completeness
3. Precision
4. Coherence
5. Relevance
6. Acceptability
7. Reusabilty
8. Effectiveness

The assessment of certain criteria is possible through the utilisation of various evaluation tools [131, 134]. However, the determination of criteria such as 'relevance' necessitates the engagement of users in order to formulate a comprehensive statement.

Available tools

In order to facilitate the design, development and extension of ontologies, there is a requirement for tools that allow the user to formalise knowledge [135]. One of the most popular tools is Protégé [136]. The software offers well-structured editors for class hierarchies and functionalities to create object properties, data types and individuals. It is characterised as open source, with its development having taken place at Stanford University. Protégé has been designed to provide support for all relevant serialisations of RDF and OWL.

Visual Notions for OWL Ontologies (VOWL) was originally developed for ontology visualisation [100]. It is a further tool that provides a clear and easily comprehensible method of developing ontologies with visual symbols, representing OWL and RDFS schemas.

The development of an ontology begins with conceptualisation. This is mainly done visually to illustrate the relationships between the different attributes. The concept is then translated into the OWL language using Protegé. To connect these two steps directly, Chávez-Feria et al. [137] have developed a library for diagrams.net. It allows the ontology to be designed using predefined boxes for classes, links for object property relationships or datatype properties. An XML export of the sketched ontology can then be converted directly into a .ttl file.

2.6.3 Combined approach for *Design* and *Development*

This thesis combines the methods presented in this chapter for a general data modelling process with specific approaches for defining an IFC Model View and a domain ontology. The individual steps are integrated into the overview diagram (see Figure 1.3) of the research methodology in Chapter 1 and described in the thesis outline.

2.7 Evaluation metrics

Having now provided an overview of the underlying technologies as well as the methodologies relevant to the development of the data models pursued in this thesis, the final section of the chapter on theoretical foundations presents the evaluation criteria by which these developments will be assessed.

The evaluation is conducted from two different perspectives (see Figure 2.14). First, three technical criteria are assessed: **Completeness**, **Feasibility of Robotic Execution**, and **Scalability and Adaptation**. Second, the evaluation considers application criteria, including **Ease of Definition and Development**, **Practicality in Robotic Testing**, and **Industry Application and Adoption**. For each perspective and each of the three criteria, two guiding questions are defined to specify the evaluation.

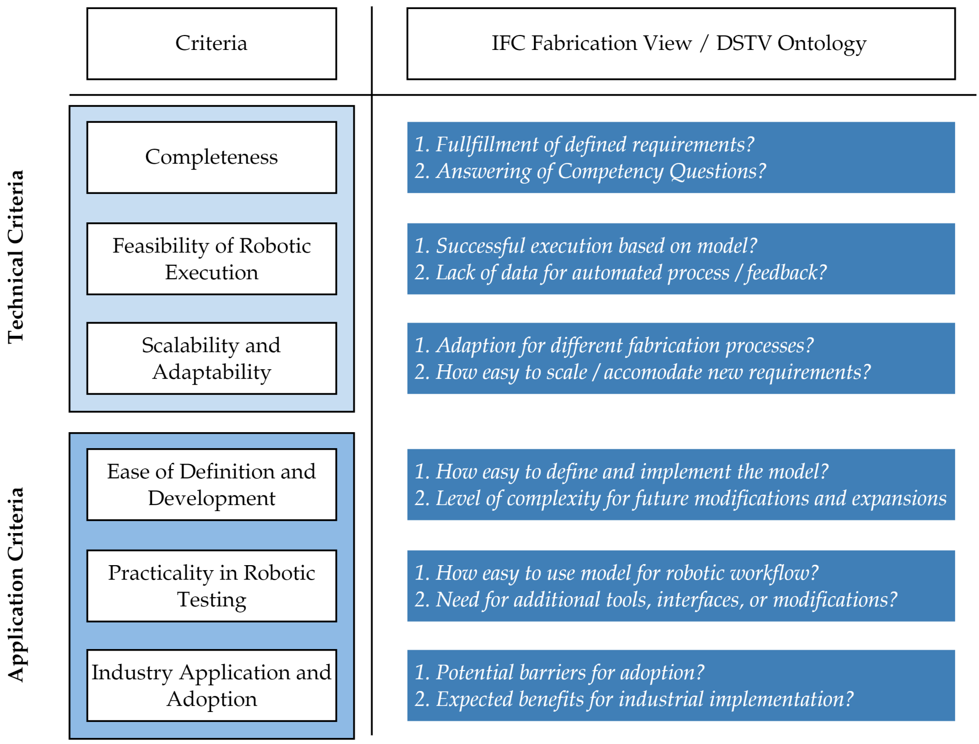


FIGURE 2.14: Evaluation metrics applied in Chapter 9

These two perspectives are critical as they provide a comprehensive evaluation of the data models from both a technical and practical perspective. The technical evaluation ensures that the models are complete, scalable and feasible for robotic execution, addressing the requirements and successful implementation. Meanwhile, the application-oriented evaluation examines how easily the models can be defined, and adopted by industry, ensuring their usability in the real world.

The evaluation is conducted using the Harvey Balls scale, where the degree of fill of each circle visually indicates how well the respective criterion is met (see Figure 2.15).

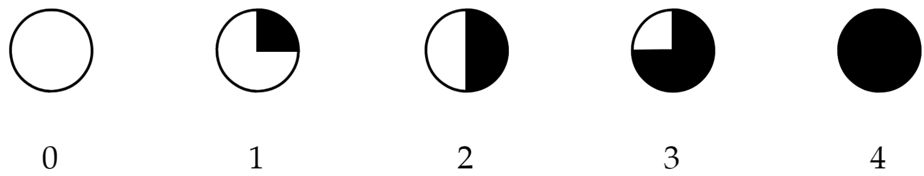


FIGURE 2.15: Harvey ball evaluation scala (see [138])

Chapter 3

Related Work

This chapter outlines the latest research, developments and implementation of the key elements of this thesis. First, Section 3.1 and Section 3.1.1 summarises the progress but also limitations of steel reuse in construction. Section 3.2 highlights the current implementation of robotics in steel fabrication. Section 3.3 provides a comprehensive overview of the contributions made to the fields of data modelling and linked steel fabrication. Section 3.4 concludes this chapter by outlining the shortcomings of existing data modelling in steel fabrication, and explaining how this thesis intends to address these gaps.

3.1 Steel reuse in construction

In the present era, the factors of sustainability, waste generation, CO_2 emissions, energy consumption and land utilisation are the primary drivers for the reuse of steel components [139]. However, this approach is not a novel concept. In the post-war period, steel components were already being reused, albeit as a result of necessity due to the shortage of steel [140].

This historical precedent underlines the practicality of steel reuse, which has since evolved into a strategic approach aligned with sustainability goals. In this context, the waste minimisation hierarchy provides a structured framework for extending the life of materials, components and structures by prioritising retention at different levels (from prevention to disposal, see Figure 3.1), as outlined by the European Parliament [141].

According to Feldmann et al. [140], steel construction reuse can be further subdivided into different approaches. The term 'retrofitting' refers to the reuse of entire buildings and structures, whereas 'reuse' more specifically describes the repurposing of individual components or elements. Building on this concept, the *PROGRESS* research project has developed methodologies and solutions to facilitate the reuse of various components in single-storey steel buildings. Through their research, the authors have outlined multiple reuse scenarios, ranging from the reuse of entire structures to that of discrete elements, offering a structured framework for implementing steel reuse in a systematic and scalable manner [142].

Within this project, several use case projects demonstrate the feasibility of reusing steel elements or even entire structures in practice [142, 143]. The projects vary in the availability of

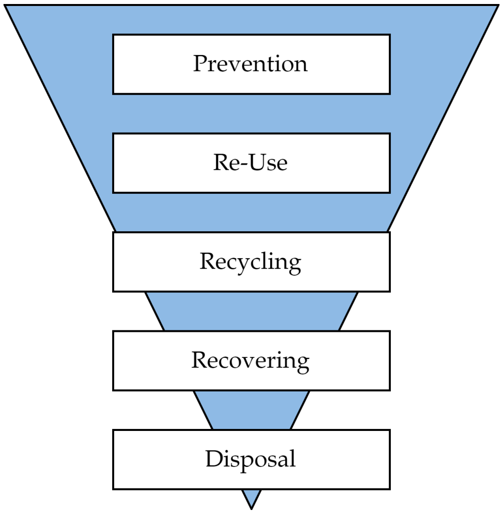


FIGURE 3.1: Waste hierarchy, according to [141]

information. For example, in the NTS building, extensive documentation was accessible, requiring only a few tests to obtain the necessary data on material strength and other properties. In contrast, the Holbein Garden project had limited initial information, necessitating more extensive testing to assess the structural components [14]. Gorgolewski [144] reported on Canadian initiatives in 2008 that focused on the deconstruction and reuse of steel structures.

Although various projects have demonstrated the successful integration of reused steel, current literature is also highlighting the existing barriers. Densley Tingley et al. [15] conducted interviews with different stakeholders working in the steel supply chain in the UK to get the insight knowlegde of the industry. Their methodology distinguishes between two aspects of the research: unprompted barriers, where participants identify obstacles without prior suggestion, and prompted barriers, which are based on a predefined set of options. Cost, material availability and no client demand have been identified as the top three consistent barriers. Lack of information about reused material was also marked by 83% of the prompted barriers.

While the demolition and deconstruction processes are the responsibility of the demolition contractors, other stakeholders are engaged in the subsequent circular processes. Consequently, these stakeholders must be integrated at the outset. It is essential to establish collaborative agreements with stakeholders, such as pre-fabricators, to ensure their support in the collection of valuable data during their processes. This data should be utilised for further purposes, while also providing these stakeholders with the necessary benefits for their participation.

Research and successful practical implementation demonstrate the feasibility of reusing steel. However, they also highlight the necessity for adequate information to serve as a basis for decision-making. This information can be obtained during the steel pre-fabrication process and must be modelled and shared in accordance with established standards and interoperable data formats. This ensures it's usability not only during fabrication and the construction process, but also during operation and deconstruction.

3.1.1 Regulations for reuse

Recent advancements in reuse practices within steel construction have demonstrated both the feasibility and the limitations of integrating reclaimed materials into modern projects. The subsequent chapter will examine the regulatory frameworks that are required to standardise these practices and ensure their safe and widespread implementation.

In alignment with the distinction between reuse and retrofitting, as previously delineated, an examination of the extant regulations reveals a sustained commitment and ongoing efforts to facilitate the long-term durability of steel.

The reuse of steel components was already regulated in 1947 in the standard DIN 1050 Sheet 2 [145]. The standard addressed the reuse of steel components, including steel bars, shaped steel, and sheet metal, salvaged from structures that had been destroyed. Various markers have been defined to categories the different levels of wear. Furthermore, the American norm 'ANSI/AISC 360-16' also deals with regulations for reusing steel [146]. Efforts to define new standards for enabling the reuse of steel are steadily increasing [147]. This document presents a series of evaluation methods that can be employed to assess the suitability of steel components for reuse. These methods are based on an analysis of the available information about the structure, the individual component, and the load history. The information considered includes the steel type, the location and dimensions of the component, test certificates, and any relevant damage history [148]. While the selection of the assessment procedure is based on the availability of different information bases and recommends different procedures accordingly, there is a lack of classification of the various components, such as material, geometry information or load history.

Furthermore, a number of regulations and standards have been established for the retrofitting of existing structures. These include DIN CEN/TS 17440 [149], in which the assessment of existing structures has been incorporated into the established regulatory framework.

Ongoing efforts have been made to establish standards and regulations for the reuse of steel components and structures. However, the development of harmonised regulations and legal frameworks remains crucial to define the limits and opportunities for reuse on a global scale [140]. In addition, a revised view of the value chain is needed [15], as well as more streamlined certification and regulatory frameworks to ensure compliance with existing regulations.

As discussed in section 3.1, adherence to these guidelines is essential in assessing the potential opportunities for steel reuse and ensuring that relevant information is available. This information includes data from fabrication, coating, material composition, testing, loading history and other relevant sources. Some of this information is typically generated during the prefabrication phase. It is therefore essential to define the specific data required and integrate it into a structured data model. This model would allow the collection and storage of critical information and ensure its availability during disassembly. Ensuring accessibility to this data simplifies the verification process and supports more efficient and cost-effective reuse of steel elements by eliminating the need for additional testing and material characterisation.

3.1.2 Design for disassembly

An important part of the reuse of building elements is, among other things, whether the elements were designed to be reused and deconstructed [139]. Hereby, it is important to differentiate between the concept of Design for Reuse (DFReu), the Design for deconstruction (Dfd) and Design for disassembly (Dfa). *DFReu* is the design for reuse, whereas *Dfd* / *Dfa* enables the deconstruction process itself, facilitating material separation but not implicitly reusing the element [150]. *DfReu* focus on the ability of the element to be used in different construction designs (e.g., ceiling, column, beam, etc.) [151, 152]. However, this thesis focuses on the data side of enabling the circular economy, rather than the design side.

3.2 Robotic steel pre-fabrication

The construction industry is increasingly facing a growing shortage of skilled labour, which poses a significant challenge to economic growth. According to a survey by the German Chamber of Industry and Commerce (DIHK), 54% of companies in the construction sector are encountering difficulties in recruiting skilled workers. Beyond the immediate challenge of recruiting the right talent, this shortage has wider implications. It is driving up wages as companies seek to attract and retain skilled workers while also increasing the overall cost of extensive and often lengthy recruitment processes [153].

Robotic systems enable significant efficiency gains by accelerating production processes, reducing material waste, and improving fabrication precision [154]. Robotics and automation have already been successfully implemented in other industries to increase worker safety and take over physically demanding or dangerous tasks [155]. Despite these benefits, the construction industry has lagged behind other sectors in adopting Industry 4.0 technologies [156]. However, the adoption of automation represents a key opportunity for SMEs in steel construction to remain competitive in the long term [157].

This section aims to provide an overview of the current state of the art in robotic fabrication for steel construction.

The field of steel fabrication has undergone significant advancements over the past few decades. Traditionally, steel components were processed manually or with the assistance of conventional CNC (Computer Numerical Control) machines. However, as steel fabrication is a pre-fabrication process, the potential for automation is higher than on site, but still difficult [158]. One of the primary challenges in steel fabrication is the high degree of variability in materials and manufacturing tolerances. Unlike industries where components are produced with high precision and minimal variation, steel fabrication involves significantly higher tolerances [159]. These variations arise from factors such as the size and weight of steel components, wear and tear of rolling tools, and economically unavoidable component deviations [160]. Furthermore, positioning steel products for welding is typically achieved through manual assembly and tacking. This process introduces tolerances that are difficult to calculate and determine within the production workflow [160]. Additional challenges stem from variations in cut edges, differing surface qualities from cutting processes, and discrepancies in steel quality. Consequently, conventional steel fabrication methodologies encounter challenges in terms of inefficiency and inconsistency, thereby constraining the comprehensive integration of robotics. The necessity for flexible process control through efficient and robust deviation detection and processing is evident [161].

To date, robots have mainly been developed for mass production and repetitive tasks in large industries (e.g., automotive). In SMEs, however, quantities are usually smaller and product variations are greater. In addition, these companies usually do not have large IT and development departments to integrate the processes internally Rickert et al. [162]. Consequently, other approaches need to be developed to make robotics integrable for these companies as well.

The EU funded project 'SMErobotics' has looked at how cognitive capabilities based on intuitive human-robot interaction (HRI) can be integrated into robotics to address these challenges [163]. To reduce the programming burden on SMEs, the 'Programming System for Efficient Use of Industrial Robots for Deburring in SME Environment' project introduces an innovative programming system that uses a combination of human input, CAD model data and sensor feedback [164]. The approach emphasises intuitive manual guidance of the robot, automatic retrieval of model data from CAD files and contour detection using triangulation sensors, ensuring efficient and accurate programming for SMEs.

While 'SMErobotics' emphasises cognitive capabilities for intelligent robotics and the 'deburring programming system' focuses on intuitive tools with multiple input options for efficient use, the concept of a skills-based framework extends these ideas by encapsulating advanced robot functions into modular, parameterisable 'skills', as described in the publication 'Robot skills for manufacturing: From concept to industrial deployment' [165]. These skills are designed to address key challenges such as indeterminacy, frequent changes of task and environment, flexibility and the need for programming by non-programmers. By integrating sensor data, human input and constraints, this approach offers a high degree of flexibility and usability.

In addition to general efforts to make robotics accessible to SMEs, there are already numerous projects specifically aimed at integrating robotics into steel fabrication. Zhao et al. [166] developed a robotic cutting system for structural steel elements which represents an advancement in the field. The system's primary objective is to facilitate the fabrication of various shapes of steel, in addition to the conventional standard shapes, such as H-beams and plates. The generation of the robot tool path is contingent on a digital model that is saved in the DSTV file. This model is then simulated with the workpiece to ascertain whether any collisions are apparent. Hippe [167] also investigated the potential of using the existing DSTV-NC file to facilitate robot control for steel fabrication processes. However, it was highlighted that the actual sizes of the steel beam were not represented in the file, resulting in inaccurate fabrication of the feature (e.g., surface marking could not be performed as the beam differed in actual width).

The application of welding processes has been significantly advanced in relation to robotic applications. Wilmsmeyer et al. [160] describe the concept of a CAD-based correction strategy that offers greater accuracy than previous systems. Using a robot-mounted optical 3D sensor, the external shape of the part to be welded is determined and projected onto the CAD data. This process serves to identify the simulated position and shape deviations.

As part of the Internet of Construction (IoC) project [19], the Cloud Remote Control (CRC) has been introduced by Robots in Architecture Research. It is based on KUKA|prc (Parametric Robot Control), a software solution that allows KUKA industrial robots to be programmed and simulated directly from Grasshopper's parametric modelling environment [168]. To enable machine-to-machine communication as part of the Industry 4.0 concept, CRC is applied to steel fabrication processes to remotely control the robot and its task, as well as to connect various IoT devices and still enable human-robot interaction [20].

In addition to the integration of individual robots into steel fabrication, there are already fully automatic production facilities, such as the 'Steel Beam Assembler' from the Zeman company. Various components are joined together by robots in a closed system, whereby a measurement is first taken to check whether the individual parts match the CAD planning model. If the process-specific tolerances are exceeded, the operator decides whether the component needs to be replaced [169].

Although machines measure parts before machining, and sensors and control algorithms ensure effective welding of tolerance-sensitive components the information is not stored or digitally available for post-processing and documentation. The related projects do not address the entire steel fabrication process chain, and do not focus on the overall data and holistic deviation management of raw materials and preceding process steps. For this purpose, a meaningful and informative mapping of the deviations and deviation management are important.

Current literature, e.g., Talebi et al. [170], define tolerances as 'defects associated with dimensional and geometrical variability'. By contrast, the present thesis defines tolerances as the standardised range in which the manufactured element or fabrication feature is allowed to be processed. Consequently, deviations are defined as the discrepancy between the planned value and the actual measured value, which is then compared with the established tolerance for that value (e.g., profile width or hole diameter).

The following is a summary of deviation and tolerance management for automation concepts and industrial robot applications from other domains, where tolerances are discussed rather than deviations. In the domain of series production, the management of tolerances assumes a pivotal role [170, 171]. Tolerances in the manufacturing process are actively controlled by deriving nominal tolerances from design drawings and distributing them across the manufacturing steps [172]. This process enables the determination of an average tolerance for each process step, and if deemed necessary, the adjustment of the tolerance window or the optimisation of the process through the implementation of deviation compensation measures [172]. Tolerance compensation measures for the assembly can consist of active deviation compensation using sensors, control and regulation, passive tolerance compensation using uncontrolled processes, such as the flexibility of joining tools, as well as a narrowing of the tolerances for the production of the individual or raw parts [172, 173].

In the domain of steel construction, however, significantly higher deviations are observed. Consequently, achieving process reliability through the division of nominal tolerances, a practice common in other industrial sectors, is not feasible. Nevertheless, 3D measurement systems, advanced image processing techniques and novel technologies for human-machine interaction, including kinesthetic teaching, gesture tracking and augmented reality, have already been employed in the programming of robots to process real-world scenarios [174–178]. In the field of robot welding, it is also possible to transfer human empirical values to a robot as a meta-semantic process model using visual or tactile sensors [179]. Furthermore, numerous industrial robot manufacturers offer specialised hardware and software packages for gas-shielded arc welding. The adaptivity of these systems encompasses methods for controlling the welding units, measuring the welding wire, localising the components, and sensor-based detection of weld seams [180]. Even new technologies such as machine learning applications have been introduced for seam profile identification in robotic welding [181].

As outlined at the beginning, parts of this work were developed within the framework of the TARGET-X research project. One of the key objectives of this project is to explore the potential of 5G technology for enabling circular and automated deconstruction in the construction industry [22]. The focus lies particularly on robotic screwing and unscrewing operations, both during prefabrication and directly on construction sites. These activities are carried out in collaboration with industrial partners employing cooperative robotic systems. A central research question concerns how screw connections can be autonomously and reliably disassembled by robots on-site. The findings demonstrate that robotic applications in steel construction have now extended beyond controlled production environments and are being implemented directly on construction sites – enabled by 5G as a robust and low-latency communication infrastructure.

The latest research indicates that the integration of robots in steel pre-fabrication is both feasible and highly efficient, provided that the existing material geometry data can be accurately recognised and processed [182]. It should be emphasised that ensuring access to and storage of this data within a linked data model is imperative to facilitate exchange between the machines and processes within pre-fabrication, as well as with upstream and downstream actors and processes.

3.3 Information management

In addition to advancements in robotic applications for steel prefabrication, extensive efforts are underway to develop information and data management systems that go beyond the conventional DSTV format.

The increasing complexity of the construction industry, coupled with the growing demands for sustainability and resource efficiency, necessitates an integrative and digital approach to the planning and implementation of construction projects [183]. In order to adequately address the aforementioned challenges, Building Information Modelling (BIM) has emerged as a pivotal methodology [56]. BIM enables the consolidation of planning, construction and operational data within a unified digital environment, facilitating precise modelling and management of the entire life cycle of a structure [184].

In the context of steel construction, which is characterised by precision and pre-fabrication [185], there are specific requirements pertaining to information management. Standards such as Industry Foundation Classes (IFC) facilitate the interoperability of disparate software solutions. Concurrently, industry-specific formats, such as the DSTV data exchange format, facilitate the expansion of possibilities through the enabling of consistent data transfer between the planning, production and assembly phases [23, 24]. In addition to these established methodologies, novel technologies such as Linked Building Data are becoming increasingly significant. Such solutions permit the flexible linking of disparate data sources, thereby addressing the issue of interoperability [186]. They facilitate the development of semantically enriched, dynamic information systems that can integrate both sustainability aspects and the high demands for precision and efficiency in steel construction.

The development demonstrates a clear transition from isolated data silos to networked, data-driven approaches that not only address the technical requirements of modern construction projects but also align with the objectives of sustainability. The subsequent section provides a comprehensive overview of research on data modelling and representations in steel fabrication.

3.3.1 IFC in steel fabrication - beyond DSTV

As previously stated, the DSTV standard is effective due to its simplicity and extensive utilisation. However, its limitations include its inability to represent additional information and to meet the requirements of a linked steel fabrication in the current era [21]. Consequently, there have been numerous efforts to define and develop data exchange for the steel fabrication that extends beyond the DSTV file.

In 2000, Finland undertook endeavours to establish a shared data exchange and a standardised data modelling framework for steel work, utilising the most recent publications from the ISO 10303 [187]. Within three national development project the ISO 10303-11 for data models and ISO 10303-21 for data exchange were applied for transferring data between disciplines and stakeholders. The project built upon existing steel standards, namely CIMsteel [188]. The practical application of these standards resulted in a more detailed description of steel, as well as the data modelling and exchange that led to the release of the updated CIM 2.0 standardisation (CIS/2) [189]. The results of the three projects were encouraging, but their subsequent implementation in industry was limited in the following years. It had already been determined that education regarding the novel techniques must be incorporated into the projects, and that demonstrations would be required to reduce any impediments.

In 2004, the concept of employing a comprehensive IFC perspective for steel construction was proposed, utilising existing IFC schemata that specified the requirements for this domain [190]. The primary objective was to leverage the IFC file to facilitate the transfer of fabrication information to the machinery. Building on the CIM 2.0 release, in 2006 Lipman [191] highlighted the importance of mapping CIS/2 and IFC product models. A mapping tool has been developed for the coordination between structural steel specifications and other parts of the overall building. The utilisation of CIS/2 and IFC for data representation proved to be effective; however, the conversion process also exposed inherent limitations. Specifically, the inability to accurately map all entities and the occurrence of information loss were identified as significant challenges. To address these limitations, the generation of comprehensive guidelines for a standardised setup of IFC files for the domain of structural steel was emphasised.

In addition to early attempts to add steel construction-specific entities and properties to the general information in the IFC model, the American Institute for Steel Construction (AISC) released an IFC 2x3 Model View (called Exchange Module 11 (EM.11)) in 2012 [192]. This included specific property sets and concepts defined for the requirements of steel construction modelling based on the IFC schema. Despite the fact that Tekla Structure and AutoCAD Advanced Steel provide support for this model view definition, there are no known applications in which IFC models have been directly imported into steel processing machines and used consecutively for data exchange or process feedback. The EM.11 Model View provides a solid foundation for further development and will be analysed in Chapter 5.

Although EM.11 was released in 2012, the DSTV standard is still the one most commonly used in steel fabrication. Rochaa et al. [59] built upon previous efforts of extending the IFC standard for the steel fabrication industry. They highlight the limitations of the DSTV standard in terms of the ability to represent complex geometries and additional information, especially required for enabling robotic fabrication. Additionally, they demonstrate that robotic fabrication demand information about the accurate position as well as the real dimensions of the steel parts. To gain this information they propose a Laser Dot and a 2D Laser Scanner. Rochaa et al. [59] prove that IFC files can be beneficial to represent the required additional information for robotic steel fabrication upon the DSTV file but do not focus on specific entities and properties.

As emphasised by Bertin et al. [150], the construction industry plays a pivotal role in relation to the depletion of resources, energy consumption, and waste generation. The potential for a paradigm shift towards a circular economy, through the reuse of load-bearing structural elements, is significant. Their proposed methodology is a chain-based approach, which is intended to increase traceability and provide information for reassembly. The methodology comprises a series of steps, ranging from the initial design for reuse to the utilisation of BIM as a knowledge base. This is enriched with LCA data and additional information for a material bank. As part of their research they highlight the need to know the physical and mechanical properties of the elements, which can be traced during the design phase. Bertin et al. [150] propose the introduction of a new LOD, designated LOD 700, which contains information pertinent to reuse. This information must be considered at the End of Life (EOL) stage, and should be incorporated into the design phase, with subsequent stages responsible for verification and expansion of the information. They propose a nomenclature for a steel beam with the most important attributes which need to be standardised and available at the EOL. Their table summarises key information which should be exported as IFC but do not cover the specific IFC specifications nor a holistic description of the steel element in terms of (robotic) fabrication.

There is an increasing demand for process and component-related information, specifically for use in automation within steel pre-fabrication. The ad hoc group *NC* of the working committee Information Technology of bauforumstahl e.V. is currently focusing on defining the IFC specifications for steel fabrication on the basis of the DSTV-NC standard [23, 24].

3.3.2 Linked Data for construction and pre-fabrication

The research conducted on the mapping of steel fabrication-related data in IFC demonstrates that the IFC standard offers a robust framework for overcoming the limitations of the DSTV-NC standard. However, Barbau et al. [110] argue that EXPRESS language do not provide additional semantics which can be used for linking additional information and processes to the element. This is particularly pertinent in the context of potential refurbishment or deconstruction, where the availability of information is crucial for the undertaking of this step [186].

OWL enables the representation of real-world concepts while incorporating well-established formats such as DSTV-NC into a Semantic Web framework. By integrating disparate and unstructured data sources, it facilitates a Linked Data environment, addressing challenges in information exchange [193]. Consequently, the utilisation of ontologies for steel fabrication may also be beneficial in tackling the issue of semantic interoperability [194–197]. The following section will review existing ontologies to identify those that can be utilised or extended for steel fabrication. This aligns with the principles of Linked Data and ontology development, which emphasise the reuse of established ontologies to avoid redundancy and ensure semantic consistency.

In 2009, Beetz et al. [111] introduced the first approach for transforming EXPRESS schema into ontologies, laying the foundation for the integration of building information models with the Semantic Web. This approach led to the development of ifcOWL, an ontology based on the IFC schema, which was later standardised by buildingSMART [112, 198]. The standardisation of ifcOWL provided a formalised structure for the representation of building data in a machine-readable format, facilitating interoperability across diverse platforms. Today, mapping and conversion tools are available, enabling the transformation of EXPRESS into OWL and LBD [105, 113, 115, 199].

Subsequent publications concerning ontologies in construction concentrated on the mapping of general conditions at a construction site. For instance, Building Topology Ontology (BOT) defines the core topological concepts of a building, as *bot:Site*, *bot:Building*, and *bot:Element* [93]. In addition to the BOT ontology, other ontologies such as the Product Ontology (BPO) [200], which focuses on product-related concepts and the GEOM Ontology [201], which is used to describe geometry, are noteworthy. The Digital Construction Ontologies (DiCon) comprise a series of individual domain knowledge representations to map the detailed construction workflows. A total of six modules are defined, covering various aspects such as entities, information and variables [193].

In addition to representing common elements and their properties, further research focused on representing construction processes. The Construction Task Ontology (CTO) describes tasks that are inherent in construction projects, such as installation, removal or modification [202], but do not focus on fabrication processes. El-Gohary et al. [203] presented the Infrastructure and Construction PRO-cess Ontology (IC-PRO-Onto), a domain ontology with a focus on infrastructure and construction processes. The concept presents different views of processes and categorises the context in which a process may exist (e.g., core process or knowledge integration process). Processes within steel fabrication may have predecessor or successor processes, but do not have different contexts.

The Internet of Construction (IoC) ontology [19] provides a more general definition of process-related concepts. It is centred around *ioc:process*, which can for example have *ioc:InputElement* and *ioc:OutputElement* for the process, but also *ioc:Resource*, which represents the machine that performs the process [204, 205]. This concept is used to represent the processes within steel fabrication.

Automated and, in particular, robotic fabrication relies on various sensors to continuously monitor the process and provide real-time feedback to the machinery. The Semantic Sensor Network (SSN) ontology offers a structured framework for representing sensors, their observations, the procedures involved, and the properties they measure. It incorporates the SOSA (Sensor, Observation, Sample, and Actuator) ontology, which defines fundamental classes and properties related to sensing and actuation [206, 207]. This ontology can be effectively utilised to describe sensors embedded within the steel fabrication process, such as measurement devices.

As steel fabrication is prior to on-site assembly some research has already been done within the ifcOWL-Design for Manufacturing and Assembly (DfMA) ontology to bridge the gap between traditional construction and pre-fabrication. It aims to translate terminology from the off-site construction domain into a machine-interpretable format by extending the ifcOWL ontology. In order to realise the ontology's overarching objective of serving as a standardised reference model for off-site manufacturing, it is structured to be language-independent and predominantly segmented into the following categories: Process, Resources, Activities and Modality [208].

The objective of this thesis is to define a data model as a successor for the existing DSTV-NC standard in steel fabrication. So far, specified domain ontology for meeting the requirements for the data transfer within this domain have not been addressed. Therefore, the first approach of the DSTV ontology has been developed prior to the formulation of this thesis. Within Chapter 6, the current status is analysed and extended to fully meet the requirements by also incorporating the existing concepts of other ontologies which have been presented in this section. Other manufacturing industries have already shown the importance of modelling the relationship between the product, the manufacturing process and the equipment to streamline and establish a cost-effective manufacturing production system [209]. Consequently, the establishment of a domain ontology has the potential to enhance the overall steel fabrication process, in addition to facilitating data flow.

3.4 Gaps in research

As described in the introduction (see Chapter 1), the widely used DSTV format provides a standardised description of component information for steel fabrication. However, the file does not fulfil the necessary requirements for incorporating additional information to support other processes, facilitate circularity, or enable the integration of digital and automated production equipment, such as robotic systems. A review of current research and development in the area of data modelling for circular construction has shown that the reuse of steel as a solution to environmental challenges requires the collection of sufficient information about the workpiece, which can be obtained during the pre-fabrication process. Efforts have been made to convert the DSTV standard into an IFC structure. This approach enables the extraction of individual data files that do not support interoperability and facilitates the storage of additional information beyond the simple representation of geometries. However, it should be noted that these developments have not been finalised and have not been conducted in the view of a holistic approach to enable robotic fabrication, interoperability and data exchange, as well as requirements for reusing steel. Linked Data and ontologies offer a robust and standardised approach to enabling interoperability between systems [186]. Previous research has focused on the representation of generic construction elements or processes, but has not focused on the specifications and potentials of steel fabrication.
The two approaches presented in this thesis have been shown to have certain advantages, which suggests that they could potentially provide a solution to serve as a successor to the existing DSTV-NC standard, while meeting the requirements of automated fabrication and data storage for reuse. To date, there has not been a detailed fabrication view for steel construction nor a domain ontology covering all the necessary entities, properties or relationships. A parallel development of the two approaches is being evaluated, along with a robotic demonstration and subsequent comparison of the two, to determine whether one or the other representation is more suitable.

Chapter 4

Foundations of the Data Model

The following sections represent the *Design* stage within the overall DRSP methodology. As demonstrated in Chapter 2 in Section 2.6.3, a supplementary methodology is presented within the *Design* and *Development* stages, with the purpose of defining a IFC Fabrication View and a domain ontology for steel fabrication in a standardised manner.

Section 4.1 specifies the focus and scope of the data model in general, followed by Section 4.2 where the exchange requirements are underlined for the different use cases. In conclusion, Section 4.3 presents the identified primary data groups in a conceptual model. This model will serve as the foundation for the development in the subsequent chapters.

4.1 Focus and scope

The preliminary step in determining the domain and scope of data representation is initiated by employing the techniques outlined in the first step of the ontology methodology, as articulated by Noy et al. [6]. Despite the fact that the techniques provided pertain to ontology development, they are applied in the more general way for defining the data model, irrespective of IFC or Linked Building Data modelling. Section 2.6.2 have been stated the fundamental queries that must be addressed in order to define the domain and scope of the ontology. The following questions are identified as being of particular relevance and are addressed in the context of the current development[1]:

Scope Question 1 (SQ1): What is the domain the ontology / data model should cover?

The data model should cover the steel fabrication domain and its related use cases.

Scope Question 2 (SQ2): For what purpose is the ontology / data model going to be used?

The data model is employed to depict the requisite classes, attributes and relations of the steel fabrication process. It facilitates the digital representation of the steel parts geometries and processes, building upon the DSTV-NC standard to encourage robotic application and further data use , e.g. for process optimisation and reuse.

[1] Parts of the SQ have been already published in Kirner et al. [21]

Scope Question 3 (SQ3): What kind of questions should the ontology / data model be able to answer?

The data model is expected to provide a comprehensive description of the steel fabrication process, along with the requisite data. Consequently, it is expected to address queries pertaining to the overall representation of steel components, the fabrication instructions (e.g., data for drilling bore holes or internal countering), and the provision of feedback in the form of additional data (e.g., measurement data or process feedback data).

4.2 Exchange requirements

Subsequent to the data model's scope, the following step is to define the exchange requirements that the model should contain. An exchange requirement is specified as a provision that delineates the information supporting the exchange in non-technical terms [81].

In alignment with the buildingSMART Use Case Management, the requirement descriptions for the given data model are structured around four use cases, which collectively encompass the entire process, from pre-fabrication in steel construction to the deconstruction of an element. As illustrated in Figure 4.1, a visual representation of the individual use cases, the processes occurring in each case, and the information flow to the database is provided. In this figure, it is evident that standardisation of the information exchange and accessibility and interoperability of the database for all stakeholders involved is essential. It is important to note that the objective of this thesis is to define the information derived from the initial three use cases, summarised as steel pre-fabrication, in a manner that aligns with the requirements of the fourth use case, namely reuse and deconstruction.

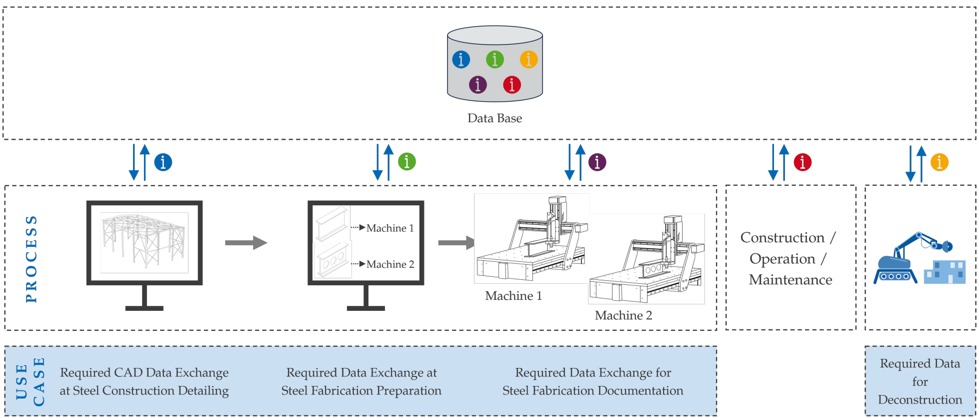


FIGURE 4.1: Process overview and use case description for requirements definition

The initial use case describes the exchange of information that is necessary at the outset of the fabrication process. These information must be provided by the party responsible for the design (CAD). The second use case focuses on the transfer of information from the initial stage of work preparation to the workplace, encompassing the operation of steel processing machines, including robotic fabrication. In conclusion, the third use case addresses the exchange of information sent back from steel processing for the purpose of documentation and subsequent utilisation in construction, maintenance, and explicitly for deconstruction. The requirements for the construction, operation and maintenance phase are not covered within this thesis.

4.2.1 Basic exchange requirements

The subsequent Table 4.1 provides a preliminary delineation of the requirements for the four use cases, in accordance with the ISO 19650-2 standards [122].

TABLE 4.1: Table of basic requirements

Use Case	Data Content	Level of Detail	Timing
Steel Detailing	Geometry, material properties, connection details	High: Includes full detailing for production	Before work preparation
Steel Fabrication on Machines	Machine instructions, dimensions, material types	Medium: Focused on fabrication-relevant details	Before fabrication on machines
Fabrication Documentation	As-built geometry, quality assurance data, process feedback	High: Includes inspection results and modifications	After fabrication
Reuse / Deconstruction	Fabrication information, material properties, connection details, location	Medium: Focused on information for reuse certification and deconstruction	Before deconstruction

The **Data Content** delineates the information that is required. The **Level of Detail** specifies the level of information need (LoIN) that the information must satisfy. The **Timing** defines the time frame in which the information must be provided.

The following Sections 4.2.2, 4.2.3, 4.2.4, 4.2.5, and 4.2.6 provide written requirement presentation of the different use cases. The tabular representation can be found in the Appendices A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8 and A.9.

4.2.2 Steel detailing

In the contemporary professional context, engineers are responsible for the general design of steel elements. This is typically undertaken using specialist software such as Tekla or Advanced Steel. These programs are utilised by engineers upon receipt of the relevant files from architects who have previously employed software such as Autodesk Revit or ArchiCAD. Research has demonstrated that there are discrepancies in the export of IFC classes and properties when comparing different software applications, despite the utilisation of identical modelling techniques [210]. Although the bSDD provides a library of BIM-related classes and properties, there are no specifications for describing steel elements in a unified way.

At the present time, the DSTV-NC standard is used as an interface between design, production planning and machine control. To demonstrate the general structure of an .NC file and the various types of data that must be modelled for the holistic steel fabrication data model, the steel beam from the case study is used as an example (see Figure 4.2). The beam is an IPE 160 with a length of 1005.00 mm. It has four bolt holes on the top front and two cutouts at the web.

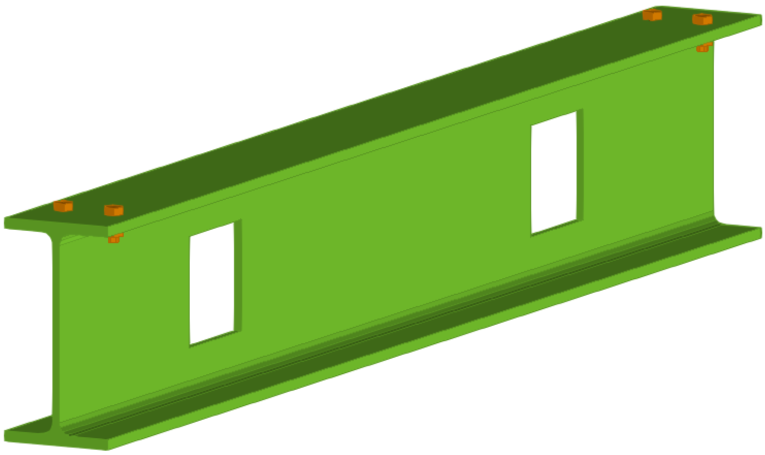


FIGURE 4.2: 3D visualisation of the steel beam for the demonstration

As described in Chapter 2, the file is organised into sections that follow a specific syntax, with each section containing distinct information about the steel component and the operations to be performed. The respective NC-file structure in ASCII is displayed below (see Listing 4.1).

The first lines within the **ST** block refer to the **Header Data**. The first four lines refer to the *order number*, *drawing number*, *part number* and *position number*. This is followed by information on the *material*, the *quantity*, the *profile name* and the *profile section*. The consecutive numbers describe the **Geometry Data**, including *length* and *height* of the profile, as well as the *flange height* and *thickness*, the *web thickness*, *radius*, *weight* and *lateral surface*. Information about web or flange mitre can also be detailed, but is not applied in this case.

The following blocks of the structure refer to different fabrication characteristics and the associated data for execution. In general, there exist various fabrication processes, such as sawing, mitring, powdering, signing or graining. In this case, the outer and inner countering are denoted by **AK** and **IK**, respectively. The out countering signifies the profile shape, while the inner countering denotes the shape for the cutouts. In relation to the described level of fabrication, the letter 'v' / 'o' identifies the side of the profile on which the feature is described. The 's' positioned behind the initial column indicates a symmetrical reference. It is imperative that all contours are closed, i.e. the X and Y coordinates of the first and last points of the contour must correspond.

LISTING 4.1: DSTV-NC file of IPE 160

```
ST
** 1.nc1
  1
  1
  1
  S235JR
  1
  IPE160
  I
  1005.0000
   160.0000
    82.0000
     7.4000
     5.0000
     9.0000
   15.80000
    0.62300

AK
  v      0.0000s    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
      1005.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
      1005.0000   160.0000     0.0000     0.0000     0.0000     0.0000     0.0000
         0.0000   160.0000     0.0000     0.0000     0.0000     0.0000     0.0000
         0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
IK
  v    725.0000s   40.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       725.0000   120.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       805.0000   120.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       805.0000    40.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       725.0000    40.0000     0.0000     0.0000     0.0000     0.0000     0.0000
IK
  v    200.0000s   40.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       200.0000   120.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       280.0000   120.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       280.0000    40.0000     0.0000     0.0000     0.0000     0.0000     0.0000
       200.0000    40.0000     0.0000     0.0000     0.0000     0.0000     0.0000
AK
  o      0.0000s    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
         0.0000    82.0000     0.0000     0.0000     0.0000     0.0000     0.0000
      1005.0000    82.0000     0.0000     0.0000     0.0000     0.0000     0.0000
      1005.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
         0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
BO
  o     50.0000s   21.0000    10.0000
  o     50.0000s   61.0000    10.0000
  o    955.0000s   21.0000    10.0000
  o    955.0000s   61.0000    10.0000
EN
```

The last block **BO** describes the list of holes. The first column describes the X value of the position, referenced from the front of the profile, the second column describes the Y value of the position. The third column indicates the diameter of the holes. In this case, all four holes have a diameter of 10 mm. **EN** marks the end of the .NC file.

In addition to the standardised description of individual parts, a standardised description for assemblies in steel fabrication was published in 2013 based on the WISCON interface from AutoCAM [55]. The document contains information regarding the relative position of the components in relation to one another, as well as optional details pertaining to the weld seams utilised. The actual parts are referenced via external part descriptions, which are typically the NC data. The requirements for steel assembly are firstly described in [211], building upon the work of this thesis.

The DSTV-NC format is constrained in its capacity to depict the life cycle of a steel structure, the bidirectional flow of information, and the capability to integrate new data essential for developing novel fabrication processes. It is therefore evident that the requirements for the steel detailing are centred on a consistent description of the material, geometry, and general project and part data. These were previously described in the DSTV file. Some of the data defined in the requirements does not need to be remodelled, as it is already defined in the overall model. This will be explained in more detail in the development section (e.g., objects or counts, as everything has an individual ID).

4.2.3 Machine fabrication

The DSTV-NC format acts as a link between design and fabrication, providing simple, standardised information for fabricating specific steel parts on the machine. As seen in the ASCII .NC file, fabrication characteristics and associated execution data are described for various processes that need to be represented in the new data model. However, it is evident that the current toolkit is inadequate in the contemporary context of different fabrication processes, e.g., milling's integration into steel construction. In practice, intelligent comment lines, self-defined blocks and bidirectional agreements between manufacturers and customers are employed. This deviation from the function of the universal interface is therefore apparent.

Machine fabrication requirements focus on geometric data for the location and reference of the fabrication process itself. In addition, data for prohibited and recommended processes for a particular feature must be represented. The requirements also include shop floor data, such as the stock material to be used, the scheduled process time and machine, and the fabrication sequence. The following section discusses additional requirements for robotic fabrication.

4.2.4 Robotic fabrication

As previously stated, the steel construction industry is facing a number of challenges, including a shortage of skilled labour, cost and time pressure, and rising quality requirements. The implementation of flexible automation systems represents a potential solution, particularly in the context of labour-intensive assembly operations within the steel construction industry. However, their use is limited due to various challenges. The key challenges of robot-assisted assembly in steel construction are geometric tolerances, material variations, component size and weight, and small batch sizes of complex components [159, 160]. The current DSTV-NC structure does provide a space to save tolerances 'to' but this it not fulfilling the requirements for robotic fabrication. Moreover, the structure does not facilitate the incorporation of actual workpiece data (e.g., precise height, width, etc.). It is essential to determine the risk of the robot or tool hitting the workpiece if the deviations from the actual values are significant. This can only be achieved by having access to the above information.

The following section summarises the findings from collaborations with industry partners on various research projects (see Chapter 1) to understand current practices, challenges and limitations in handling deviations in relation to defined tolerances and recording information within steel fabrication.

Current state of deviation management

In many companies, deviations are not consistently documented or monitored. Changes and modifications to the steel element during production are often guided by the practical experience of skilled workers, who typically share these adjustments informally, rather than through official documentation. This practice highlights a significant gap: while knowledge is retained within the workforce, it lacks formalisation and documentation. Consequently, the absence of structured records on empirical data and deviations hinders standardisation, scalability, knowledge transfer, and consistent quality control. In cases where deviations occur, companies employ mechanical compensation methods to be in line with tolerances. By incorporating overlength compensation and counter-tensioning, companies can effectively mitigate distortion and irregularities in production. Measurement systems installed on various machines help to detect part lengths and confirm dimensional accuracy, but these systems primarily ensure that the correct part is loaded rather than accurately capturing deviation data. Current measurement systems provide a basic level of quality control, but lack the precision required for more advanced deviation management and data capture that is not stored anywhere.

Robotic welding systems

During my thesis research, I collaborated with several companies, including Wurst Stahlbau GmbH, which has implemented robotic welding systems like the Zeeman line (see Figure 4.3).

FIGURE 4.3: Zeeman robotic fabrication system at Wurst Stahlbau GmbH

This system, although automated, has several limitations in terms of deviation management. It performs an initial scan of the components at the start of the process, but lacks real-time adjustment capabilities, meaning that it does not account for deviations that occur during operation. While the robot can automatically detect gaps in welds and adjust the weld height accordingly, this process relies on pre-defined adjustments rather than dynamic tool changes. In addition, the system includes an error detection mechanism that identifies deviations at weld points and alerts the operator if excessive gaps prevent the process from continuing.

Although the robotic system represents progress in automation, it still requires human intervention for setup and monitoring, especially when there are complex tolerance issues. Moreover, each part undergoes a preliminary test to confirm that dimensions and placement match the model, further underscoring the reliance on human oversight.

In summary, robotic fabrication necessitates the integration of precise measurements of the actual workpiece both before and after the fabrication process. The incorporation of digitally defined tolerances is essential for validating fabrication features, while the recording of process parameters facilitates analysis of the relationships between various settings, materials, and fabrication processes. This approach is vital for the effective management of deviations and the digitalisation of fabrication workflows.

Table A.4 and Table A.5 in the Appendix A present examples of the defined tolerances for steel profiles (specifically I/H beams) as well as an excerpt from the process tolerances for individual features. The tolerances specified in other steel construction standards, such as DIN EN 10024, DIN EN 10025, DIN EN 10365, and DIN EN 1993-1-1, are only available in PDF documents or printed books. Consequently, these values must be modelled in a way that allows for machine-readable interpretation.

4.2.5 Fabrication documentation

Fabrication documentation and data feedback offer multiple benefits for data analysis, including quality control, traceability, and process optimisation. The data that can be stored generally falls into one of four categories. The first category includes general information such as the location, time and company involved in the fabrication. The second category includes machine and tool data, with further points specifying process parameters that can be tracked. These include parameters for drilling, welding and marking. Finally, the documentation covers quality control, i.e. the recording of actual fabrication features. This refers, on the one hand, to the mapping of measured values, which has already been described in this chapter, and, on the other hand, to the digital storage and subsequent retrieval of DIN/ISO standards to enable comparison of production with defined tolerances (see last section).

Documenting process parameters, such as the machines and tools used, facilitates process optimisation within production. For example, tool failure can be documented and predictive maintenance models employed to plan replacement, thereby reducing downtime. Comprehensive documentation of processes and machine parameters enables the design of more efficient processes and machines. In terms of energy efficiency, machine and process consumption can be recorded to enable process optimisation and cost allocation to customers. This data could also be valuable for life cycle assessment (LCA) databases. Quality can be used to monitor values such as speed or amperage during welding, providing an effective control instrument.

The Appendix A.6 and A.7 give the current definitions for the fabrication of individual parts, as this is the focus of this thesis; assembly processes have only been added as an example for welding.

4.2.6 Reuse and deconstruction

In recent years, there has been a growing emphasis on sustainability and resource efficiency in the construction industry, which has highlighted the importance of reusing steel elements. In order to establish a holistic and accepted process for the reuse of steel elements, the availability of information is crucial. As identified in the preceding literature review, information on the material itself, how it was fabricated, and its operational and maintenance history is required.

This approach includes several key datasets essential for enabling deconstruction and reuse, depending on data availability. Material properties, such as steel type, mechanical characteristics, and surface treatments, provide fundamental information about the components. Geometric data, including precise dimensions and connection details, ensure accurate reconstruction. Quality assurance records verify compliance with DIN norms and other relevant standards. Digital identification allows for efficient tracking and management of materials. Structural performance data, covering aspects like load history and deformations, help assess the integrity of reused components. Lifecycle information, documenting original usage as well as installation and removal history, supports decision-making in reuse applications. Inspection and testing records, including results from non-destructive testing, corrosion analysis, and material certification, further validate the suitability of components. Lastly, environmental impact and sustainability metrics, such as carbon footprint, life cycle assessment data, and reuse potential, contribute to assessing the ecological benefits of material repurposing.

Within the scope of this thesis, the focus is exclusively on datasets pertaining to pre-fabrication, as they are the primary concern for data modelling. The list of requirements for deconstruction and reuse are listed in the Appendices A.8 and A.9.

4.2.7 Structural requirements

The general requirements for the data structure include the use of established standards (e.g., IFC classes, attributes, relationships, and properties) and the incorporation of Linked Data structures such as ontologies, OWL, and RFD schemas to represent the necessary data. The structure should be extensible to accommodate future requirements, capable of digitally representing processes and machines, and support semantic connections. Additionally, it must integrate IoT data and ensure interoperability with other formats and stakeholders outside the steel fabrication domain.

4.3 Conceptual model

A review of the existing DSTV standard reveals significant limitations. In particular, critical information is missing, which could pose challenges for applications such as robotic fabrication, data reuse, or meeting the growing demand for more comprehensive fabrication details and bi-directional data flow. The extensive requirements for all the identified use cases are summarised in more detail within the Appendices A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8 and A.9.

Following the methodology for developing a data model (see Figure 1.3), the conceptual model is evolved from the preceding requirement analysis. It structures and consolidates the identified needs and elements, derived from the four use cases outlined in Section 4.2, into clearly defined groups. Figure 4.4 illustrates the conceptual structure for the data model. The grey boxes indicate the main categories: 'Process Requirements', 'General Data', 'Fabrication', 'Reuse & Deconstruction', and 'Sub-Processes'. In the past, the central part of the DSTV-NC file was the **Work Piece**, upon which everything was linked. However, this workpiece-centred modelling approach was too rigid to accurately represent the complex workflows and information exchanges in the pre-fabrication phase of steel construction. In contrast, a **process**-centred approach allows for a more dynamic and flexible structure, enabling more precise modelling linking associated information. Therefore, the conceptual model and the subsequent modelling follows this approach.

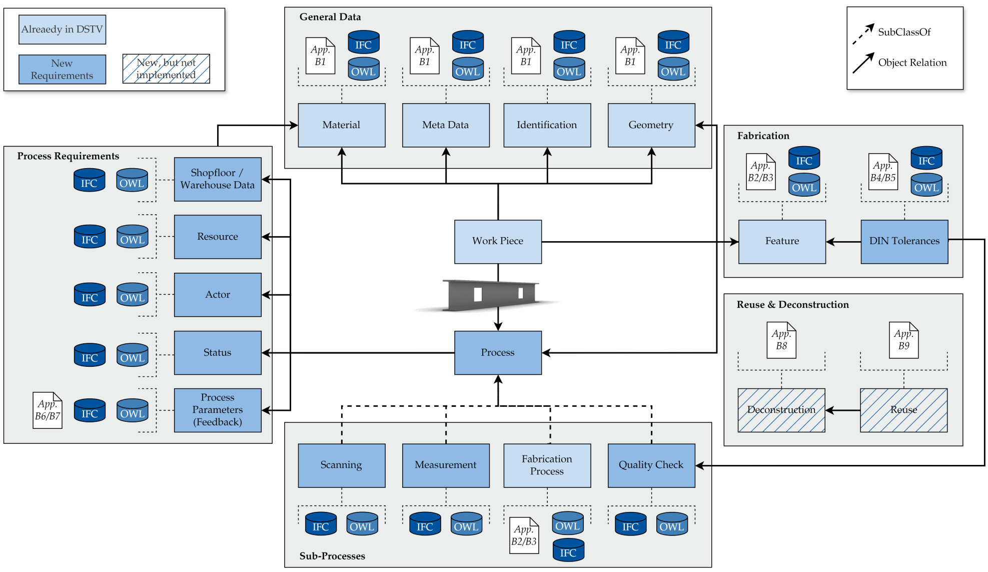


FIGURE 4.4: Representation of the conceptual model, summarising the requirements for the multi-modal implementation strategy for the data representation in steel fabrication

Within each category group, there are rectangular boxes summarising the data required for digital pre-fabrication. Light blue boxes represent data that is already defined in the DSTV-NC file, but which must also be incorporated into the new data model. The darker blue boxes highlight newly identified data requirements, as outlined in this section. Additionally, the hatched boxes show requirements that have been defined but will not be realised within this thesis. The curly brackets on the boxes indicate where the details of the respective criterion can be found in the appendix (see the note sheet). Cubes labelled 'IFC' and 'OWL' indicate that the requirement has been implemented in both data formats used in this thesis. Boxes that are not implemented in this thesis do not contain any representation of these data models.

Building upon the general description of Figure 4.4 the following provides a more detailed description of the conceptual model. The different requirements are linked to each other, either describing subclass relationships (dashed lines) or object relations (solid lines). The central part is the **Process** which is linked to the **Workpiece**, general data, different process specifications and various sub-processes. The general data comprises **Material**-related information, as well as **Meta Data** pertaining to the creator and software of the file, the orderer, and the order number, which serves to specify the workpiece. It also incorporates **Identification** data, including the unique identifier, the project to which it relates, and the part number, as well as **Geometry** and profile data about the workpiece. This information have been so far described in the header part of the DSTV-NC file. The **Fabrication Feature** is linked to the **Workpiece**, but also to the geometry, process and **DIN Tolerances**. The fabrication feature delineates the tasks to be performed on the workpiece. The link to the defined DIN tolerances facilitates a database-based adjustment of feature parameters before and quality check of the workpiece after fabrication, in relation to specific features, e.g., tolerance for a hole.

The **Process** class incorporates sub-process classes, thereby facilitating a holistic model of the various steps involved in the steel fabrication process. Firstly, the component or workpiece is **Scanned** to ensure precise recording of the geometric properties. **Measurements** are then carried out on the basis of the recorded data, which enable the target/actual deviations to be checked precisely. If necessary, adjustments are then made to the production data in order to take into account the specified dimensional reference and the tolerated deviations. In this context, the subsequent **Fabrication Processes**, such as drilling, sawing, cutting or welding, are closely interlinked and carried out precisely, taking into account the previously determined parameters and specifications. The **Deconstruction** process is also characterised as a sub-process, and the information required for **Reuse**, generated during prefabrication, is linked to the workpiece and the associated information accordingly. Data relating to assessment after installation, for example, is not displayed here as it is not linked to pre-fabrication.

Each process is characterised by a set of additional specifications, including the utilisation of a specific **Resource** (e.g., a machine) where the process is conducted, the designated **Actor** (i.e. the operator or the company responsible for the process), the **Status** of the process (e.g., whether it is 'planned', 'ongoing', or 'finished'), and the precise **Process Parameters** (e.g., the tool's speed). These parameters can also be used to provide feedback for the purposes of documentation and process analysis. Additionally, data from the **Shopfloor** and **Warehouse** are associated with the process and the material. These include specific data such as the material used from the stock, the machine scheduled for the process on the shop floor, or the maintenance status.

4.3.1 Competency Questions

Building on the requirements and conceptual model previously defined, this section introduces a set of *Competency Questions* to guide the evaluation of the data model's ability to represent and support the identified information needs (see Table 4.2). The approach of the competency questions are in line with the *101 Ontology Development* of Noy et al. [6].

TABLE 4.2: Table for Competency Questions

No.	Type	Competency Question
1	General Information	What is the ID of the steel part?
2	General Information	What is the software it was detailed in?
3	Material	What is the material?
4	Geometry	What is the overall geometry of the workpiece?
5	Fabrication Feature	What are the features of the workpiece?
6	Fabrication Feature	What is the planned value of the diameter?
7	Process	What are the processes linked to cutout?
8	Process	What are process parameters for drilling?
9	Process	What is the scheduled time for drilling?
10	Process	What machine was used for the plasma cutting process?
11	Measurement Process	What is the real value of the height of the cutout?
12	Adjustment Process	What is the defined tolerance for a cutout?
13	Process Feedback	When did the plasma cutting process take place?
14	Process Feedback	What was the average speed of the robot while plasma cutting?

4.3.2 Response to RQ 1

This chapter analysed the essential data categories for a data model that integrates design, pre-fabrication and automated fabrication processes in the steel construction domain. To achieve this, requirements were identified through a literature review and collaboration with industry partners to ensure that both theoretical and practical perspectives were considered. The analysis identified key categories of data, which were structured into a conceptual model. This model provides a visual representation of how different data sets relate to each other, highlighting the key categories that need to be modelled in response to the research question. This illustration clarifies which requirements will be implemented and applied using the defined example. Implementation in the IFC schema is described in Chapter 5, and implementation in OWL is described in Chapter 6.

By incorporating insights from four industry partners at different stages of the German steel construction value chain, a broad perspective was ensured. However, it is recognised that international requirements and additional stakeholder perspectives may not have been fully captured. Future work should extend this model to include additional data categories relevant to other regulatory environments and manufacturing workflows.

Chapter 5

Development - IFC Fabrication View

The following chapter builds upon the foundations of the data model. Section 5.1 and Section 5.2 examines existing entities and properties within the context of the IFC schema. The aim of this analysis is to identify which concepts can be used to meet the requirements, and which concepts need to be redefined to fully address all identified needs. The subsequent Section 5.3 describes the development of custom property sets for steel fabrication. However, it should be noted that further specifications may exist, and this work aims to provide an initial approach. Section 5.4 will shortly cover the options for checking compliance of IFC models.

In order to facilitate the mapping of the entire life cycle, IFC version 4x3 provides over 800 classes, numerous standardised property sets and a series of relations. It is important to note that an extension of the data model within a released IFC version in terms of classes and relations is not permitted. Conversely, the definition and utilisation of property sets is permitted for users. The following definitions refer to the current IFC version 4x3.

5.1 Exchange Module - EM.11

In 2012, the American Institute of Steel Construction (AISC) has published the EM.11, as a first approach to specify steel related classes and properties [192]. Despite the fact that Tekla Structure and AutoCAD Advanced Steel are capable of supporting the EM.11 MVD with detailed steel classes and properties, the broad application of this model within industry is not yet established. Furthermore, there are currently no known applications where IFC models have been directly imported into steel fabrication machines. However, EM.11 provides the basis for defining the steel fabrication specifications, from which further property sets are defined to map the requirements of the use cases.

The following Table 5.1 provides an overview of the AISC EM.11 property sets and their associated properties. It also classifies them according to their suitability for steel fabrication requirements. This depends on whether the properties are not important for fabrication or are covered by a new concept for meeting additional requirements. The following sections focus on analysing the use of existing IFC entities and defining new properties to fully meet the requirements. The table here displays the first twelve property sets of the EM.11 release.

TABLE 5.1: AISC EM.11 property sets and classification

Number	AISC EM 11 Property Set	AISC Properties	Classification
5.2.4.1	Pset_StatusInformation	Status	suitable
		Change Status	suitable
		Status Approved	suitable
5.2.4.2	Pset_Assembly Identification	Assembly Mark	suitable
		Client Mark	suitable
		PrelimMark	not suitable
		Shipping Mark	suitable
		Barcode	not suitable
5.2.4.3	Pset_PieceIdentification	PieceMark	suitable
		Indication Mark	not suitable
		Main Piece Tag	suitable
		PrelimMark	not suitable
		IDCode	not suitable
		Custom Type	not suitable
5.2.4.4	Pset_DrawingNumber	DrawingNumber	suitable
		Drawing Index	suitable
5.2.4.5	Pset_VersionInformation	VersionNumber	not suitable
5.2.4.6	Pset_SchedulingInformation	PhaseSequenceID	suitable
		SequenceLevel	suitable
		OnSiteBy	suitable
5.2.4.7	Pset_ToleranceForLayout	LayoutTolerance	not suitable
5.2.4.8	Pset_Castellation	CastellationType	not suitable
		CastellationEndPostWidth1	not suitable
		CastellationEndPostWidth2	not suitable
		CastellationSpacing	not suitable
		CastellationHeight	not suitable
		CastellationWidth	not suitable
		CastellationDepth	not suitable
5.2.4.9	Pset_Material	MaterialID	suitable
		MaterialGrade	suitable
		MaterialCertification	not suitable
		MaterialType	not suitable
5.2.4.10	Pset_SurfaceTreatment	SurfaceTreatment	suitable, but will be defined in more detail
5.2.4.11	Pset_BoltHole	FeatureSubType	not suitable, all holes will be defined in a separate Property Set attached to the Voiding Feature
		BoltHoleDiameter	not suitable, all holes will be defined in a separate Property Set attached to the Voiding Feature
		PartialHoleIndicator	not suitable, all holes will be defined in a separate Property Set attached to the Voiding Feature
		PartialHoleDepth	not suitable, all holes will be defined in a separate Property Set attached to the Voiding Feature
5.2.4.12	Pset_Feature	CriticalFace	not suitable, will be defined in general "Fabrication Process" Property Set
		FabricationMethod	not suitable, will be defined in general "Fabrication Process" Property Set
		DesignConstraint	not suitable, will be defined in general "Fabrication Process" Property Set

5.2 Analysis of existing IFC structure

The specifications for the IFC Steel Fabrication View are outlined in this section. Initially, an analysis is conducted to identify which existing IFC 4x3 elements may be utilised or adapted for steel pre-fabrication purposes. Figure 5.1 provides an overview of the IFC specifications developed in this thesis, which are derived from the conceptual model and aligned with the defined requirements. Existing IFC elements are depicted in light blue, whereas those introduced within the scope of this work are shown in dark blue. This approach ensures the fulfilment of all requirements for IFC-based fabrication of steel components and establishes a data-driven basis for future deconstruction processes. The individual specifications are subsequently detailed according to the overarching categories (indicated by grey boxes). For additional information please refer to [23, 24].

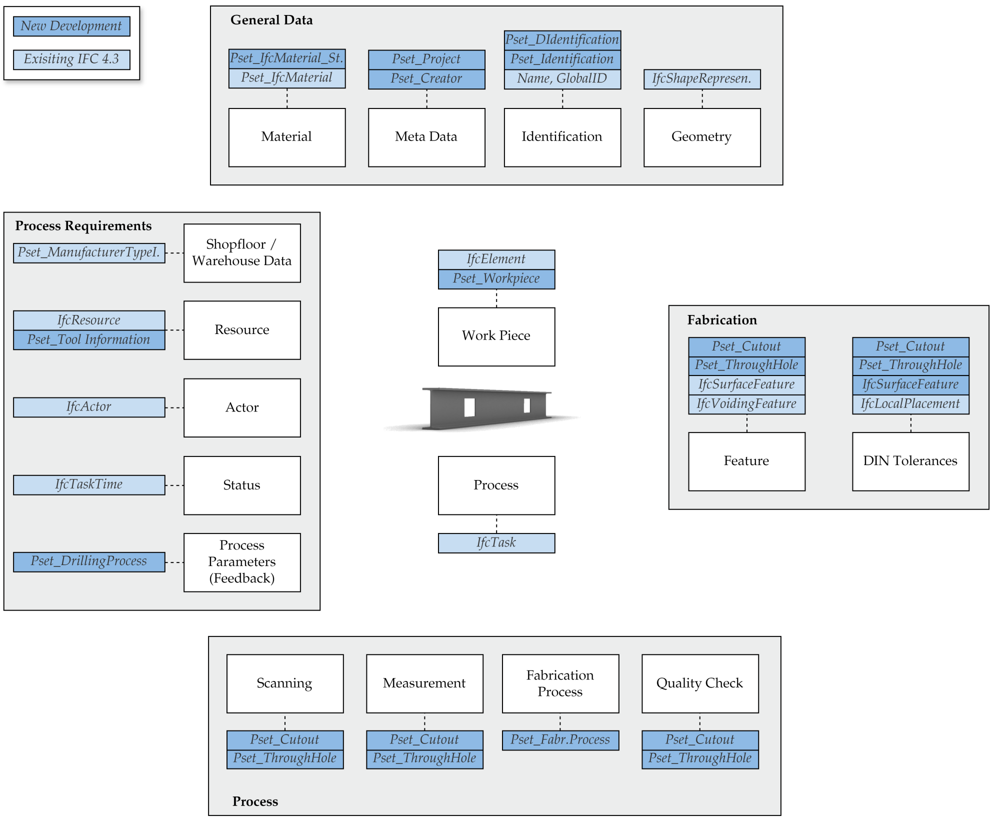


FIGURE 5.1: Summary of IFC specifications for steel pre-fabrication, with existing entities in white and newly defined property sets in green, supporting the replacement of the DSTV-NC standard

5.2.1 General data

In accordance with the IFC 4x3 schema, the general data related to the workpiece, originally contained in the header block of the DSTV-NC file, such as material properties, metadata, identification, and geometry, are already represented using existing IFC entities and attributes (e.g., *IfcElement*). The detailed about the applicable elements are described in the subsequent sections. However, data from the ST block of the DSTV-NC file, which includes information such as the order number, CAD system name, fabrication company or details of the steel detailing process (e.g., the entity responsible for generating the detailed drawings), will be consolidated into a new property set specifically designed to capture the steel detailing process (*Pset_Workpiece*). Not all of this information is relevant to all project partners; some details may be intended for internal use only. Therefore, it is essential to analyse how this data can be managed to ensure that it is not inadvertently shared or integrated into the overall IFC model of the entire structure. Instead, appropriate measures should be taken to keep such information within the company.

Material

The material is defined through *IfcMaterial*, which can be associated with elements via *IfcMaterialProfile* or *IfcMaterialProfileSet*. A specific property set *Pset_MaterialSteel* is defined in addition which is explained in Section 5.3.

Identification

Identification and metadata are typically captured using attributes such as *GlobalId*, *Name*, and *Description*. Additional requirements for the identification of a steel element may arise, for instance, when distinguishing whether an element serves as a main piece to which sub-pieces are welded or when multiple elements are grouped into a single shipping unit. These details are incorporated into specific fabrication property sets. Moreover, the potential reuse of materials in the future necessitates a digital identification method to ensure traceability within the IFC model, allowing retrieval of the relevant information. To address this, a dedicated digital identification property set is defined (see Section 5.3).

Geometry

The focus of the work is on the production of standard profiles and flat sheets. Standard profiles are defined as parametric profiles, with their geometry represented as an extrusion. Sheets are also represented as extrusions, but can take on any contours. The given IFC definitions for profiles are adopted (*IfcShapeRepresentation*); however, only parameterised profile (*IfcParameterisedProfileDef*) definitions, such as *IfcIShapeProfileDef*, are permitted. This ensures a standardised and efficient representation. Other profile definitions have to be modelled using *IfcArbitraryClosedProfileDef*.

Structure

The IFC model imposes no restrictions on the spatial structure, supporting buildings, bridges, roads, railways and facilities with or without subdivisions. For the purposes of fabrication, it is essential to define fabrication units. The *IfcElementAssembly* entity aggregates assembly parts for unique identification. Although IFC mandates a globally unique identifier for *IfcElementAssembly*, it does not align with steel construction conventions. Consequently, a property set that is tailored to steel construction identification is required for assemblies. The property set from the AISC EM.11 *Pset_AssemblyIdentification*, in conjunction with the selected properties (see Table 5.1) may be utilised for this purpose. The primary focus of the present thesis is the presentation and robot-assisted production of individual parts. The fundamental principles and concepts are developed in this research, and can subsequently be expanded for the purpose of assembly.

5.2.2 Process

The DSTV-NC file does not explicitly describe the fabrication process itself. While it contains information about geometric features such as contours and holes, it does not specify the recommended or prohibited fabrication methods for these features (e.g., prohibiting plasma cutting for hole). Additionally, the DSTV-NC file does not include details on which machines execute the process, which tools are used, scheduling information, process status, or feedback. Only the geometric attributes of the features are provided, such as diameters and vertex coordinates defining the processing locations. The specific fabrication data related to the process are assigned to the voiding feature and elaborated in the next section, while general process-relevant data are addressed here.

The IFC 4x3 schema provides the entity *IfcProcess*, which is defined as a single activity or event that is ordered in time. It can have sequence relationships with other processes (*IfcRelSequence*) and can be defined as *IsPredecessorTo* or *IsSuccessorFrom*. An *IfcTask* is a a specialisation of *IfcProcess* and describes a specific task or step in a process. A task can be linked to the output (e.g., *IfcVoidingFeature*) via *IfcRelAssignsToProcess* and can be associated to an IfcResource (e.g. machine) via *IfcRelAssignsToResource*. The *IfcTaskTime* attribute is attached to the *IfcTask* and provides time-related information (e.g., scheduled start, scheduled finish). Furthermore, each task can have specific property sets linked to it for the specific process (e.g., drilling a hole, plasma cutting a cutout). These are explained in Section 5.3.6.

To capture fundamental steel fabrication process-related information, a general property set is introduced (*Pset_FabricationProcess*, see Section 5.3). This property set includes basic parameters such X and Y coordinates and specifications regarding permitted and prohibited fabrication methods. Data about the machine which executes the process can be stored in the given property set *Pset_MachineFabrication*.

5.2.3 Local placements

The *IfcLocalPlacement* application is employed for the purpose of defining the position and orientation of an object in relation to a reference system. The entity facilitates the precise placement of components or groups of components within a spatial context. The general fabrication processes, in particular robotic fabrication in connection with measurements and deviations from the planning data, require a reference point. Within the DSTV-NC file, the *View* has been utilised to specify the side of the profile to be machined, thereby defining a reference point for the machining values. Similarly, in the IFC schema, it is necessary to define the point from which the process is executed or deviations are measured. Knowing only the machining position relative to the element is insufficient. To address this, the *IfcAnnotation* entity is selected to store this additional reference information, which can be used for the process execution, measurement processes and subsequent analyses, including potential adjustments of machining parameters. This entity can be integrated into the IFC hierarchy, includes a local placement, and supports any geometric representation. By definition, the XY plane of the *IfcAnnotation* entity's local placement serves as the reference plane, while the Z-axis defines the measurement direction.

5.2.4 Fabrication feature

In the IFC schema, fabrication features are represented as specific entities that define geometric modifications and processing requirements applied to structural elements. These features, such as cutouts, holes, and machining operations, provide essential information for digital fabrication and fabrication workflows.

Voiding feature

Fabrication processes, such as sawing, drilling, and milling, are typically material-removal operations. Historically, these machining processes were represented as *IfcOpeningElement*. However, with the advent of IFC version 4, the *IfcVoidingFeature* class was introduced, which more accurately represents machining features. The predefined element type enables further specification of the voiding feature, including *CUTOUT*, *HOLE*, *MITER*, and *NOTCH*. The geometry of these void solids is defined as an extrusion of any given contour. Further specification of the processing is carried out by additional property sets. Given that the *IfcVoidingFeature* contains fabrication-related information in its attached property sets, it is now essential that any machining activity, including sawing, is modelled using this class.

Surface feature

Additional features of the DSTV-file also include markings. Point, line or area markings on the component (e.g., centre-punch marks, texts, etc.) are defined as *IfcSurfaceFeature.* The classification of point-shaped markings (*MARK*), line-shaped markings (*LINEMARKING*), and area-shaped markings (*HATCHMARKING*), as well as texts (*TAG*), is facilitated by predefined types. The geometric representation of these surface features is achieved through the use of planar geometries or character strings (*text*). The determination of possible fabrication processes is governed by additional property sets. The surface treatment can also be represented with *IfcSurfaceFeature*. In this instance, the predefined type is set to *TREATMENT*. It is imperative that a separate surface feature is created for each surface treatment (e.g., blasting, galvanising, painting) and added to the fabrication-specific property sets. Since IFC version 4x3, it is necessary to assign surface features to components using a special relation, *IfcRelAdheresToElement*.

Fastener

The *IfcFastener* class is used to describe fasteners in the design or fabrication process. It represents the general concept of fasteners that can be used to join elements together, either by mechanical means (e.g., screws, bolts) or by other methods such as welding or bonding. This class can describe different types of fasteners, including mechanical fasteners or welded joints, and capture their properties such as type, size and material. The *IfcRelConnectsElements* relation defines which elements the fastener connects.

5.2.5 Tolerances

Currently, the DSTV-NC file contains only a 'to' field, which does not store explicitly defined tolerances for fabrication features or the steel profile itself. Instead, it uses predefined values such as *min* and *max*. However, for digital quality assessment against fabricated or measured values, tolerance data must be digitally accessible. Various DIN standards (DIN EN 10034, DIN EN 10279, DIN EN 10056-2 and DIN EN 10219-2) specify the minimum and maximum product tolerances, covering both profile characteristics (e.g., height) and process tolerances (e.g., hole diameter). This crucial information is currently absent from the IFC schema.

There are two ways to add the tolerances. The first option is to add a *ToleranceMin* and a *ToleranceMax* property to the universal fabrication process property set explained in section 5.2.2. This approach ensures that tolerance rules are applied consistently across different feature types and simplifies querying and data extraction in IFC models. The second option is to incorporate the defined values into feature specific property sets. Tolerances are directly linked to feature-specific values such as diameter, depth or width as they are also defined for these specific values. The second option is used and further explained in section 5.3.4.

5.2.6 Process requirements

Specifications

In accordance with the capabilities of *IfcTask* in IFC 4x3, it is possible to assign resources to a process to reflect the equipment, materials, or labour involved in its execution. In this context, *IfcResource*, and more specifically *IfcConstructionEquipmentResource*, is employed to represent the machine utilised in the steel fabrication process. To supplement the generic resource assignment, additional details pertaining to the machine, such as manufacturer, model, and performance characteristics, are specified using predefined property sets. The property set *Pset_ManufacturerTypeInformation* is used for this purpose, providing structured metadata related to the equipment. Furthermore, the location where the steel fabrication is carried out is modelled using *IfcActor*, which allows the identification of the responsible organisation or facility. This entity may be linked to the task either directly or through a more complex role-assignment structure.

Feedback

It is possible to digitally represent process feedback data with some given entities in IFC 4x3 and new defined properties. This facilitates support for fabrication monitoring and quality control. The data also enables documentation for the reuse of the element or internal process optimisations. It can be achieved through a combination of *IfcSensor*, *IfcTaskTime*, and custom property sets attached to *IfcTask*.

IfcSensor can be used to represent measurement devices tracking parameters like cutting speed, temperature, or tool condition during fabrication. These sensors can be linked to the fabricated element using *IfcRelAssignsToProcess*. *IfcTaskTime* can be attached to track the actual start / finish of a process. To store detailed process data, a custom property set can be defined (see Section 5.3.7). This property set can contain specific parameters relevant to the fabrication process (e.g., cutting speed, or power consumption) and are linked to the relevant *IfcTask*.

5.3 Development of custom property sets for steel fabrication

The EM.11 standard and the IFC 4x3 schema offer a comprehensive set of entities, attributes, and property sets to represent machining elements (e.g., holes, cutouts) and fabrication processes (e.g., sawing, drilling, milling) as specified in the DSTV standard. These entities also incorporate aspects related to process scheduling and general workpiece data, partially addressing the defined requirements. However, fully integrating steel fabrication within the IFC framework necessitates the introduction of additional property sets.

Referring back to Figure 5.1 it is highlighted, that the proposed properties cover a wide range of information, from identifying permitted and restricted fabrication processes to specifying parameters for different drilling operations and material input details. They also facilitate documentation by capturing data on the machines and tools employed, along with key information required for robotic automation, such as real-time workpiece measurements and digital representations of processes and sequences. Moreover, they enhance quality control by providing detailed measurements and a digital representation of tolerances, ensuring precise monitoring and verification throughout fabrication. As some property set names appear across multiple categories within the extended conceptual model presented in this chapter, it is important to note that a single Pset may address different requirements by means of distinct properties.

The forthcoming Psets are now the final state. During the preparation of the demonstrator some adjustments were made, so it reflects a cycle within the DSR method from *Demonstration* to *Design* and *Development* again.

5.3.1 Material

As described, the *IfcMaterial* class is selected to represent steel specifications. A property set *Pset_MaterialSteel* is linked to it with additional properties (see Table 5.2).

In the event of an alternative material being selected during the work preparation stage, this must be documented in an additional property set. It is imperative to note that the consistently represents the planned state.

TABLE 5.2: Pset_MaterialSteel

Property Set	**Property**	**Description**
Pset_ MaterialSteel	Material ID	Unique ID of the material from the supplier.
	Material Grade	Material grade, e.g., according to EN 10024-2 to -6.
	Material Quality	The quality of the material, e.g., according to EN 10027-1.
	Supply Condition	Requirements (as planned) placed on the material, e.g., Z-Grades, test procedures, certificates, etc.

5.3.2 Workpiece meta data

To accurately represent the specific data from the ST block of the DSTV-NC file, the following property sets have been defined (see Table 5.3). This property sets enables the storage of steel detailing-specific information. Certain header details from the DSTV file are not included here, as they are categorised within other property sets or already defined in the IFC header or IFC owner history. These are for example, single part identifier (see Section 5.3.3), material specifications (see Section 5.3.1), the quantity, as every element as a unique identifier or the overall object as the steel element is linked to the overall *IfcBuilding*.

TABLE 5.3: Psets for workpiece metadata

Property Set	Property	Description
Pset_ Workpiece	Order Number	The unique order number associated with the workpiece
	Drawing Number	The drawing number related to the workpiece
	Part Number	The specific part number of the workpiece
	Position Number	The position number of the workpiece
	Part Info Text	Informational text related to the workpiece
Pset_Creator	CAD System Name	Name of the CAD system that created the file
	CAD System Version	Version of the CAD system used
	Company Name	Name of the company that generated the file
	User Name	Name of the user who created the file
Pset_Project	Project Order Number	Order number associated with the project
	Orderer	The entity that placed the order
	Object	The object associated with the project

5.3.3 Identification

The definition of steel construction components, such as beams, columns, members and plates, is already established within the IFC framework. However, as with assemblies, there is a discrepancy between the globally unique identifier for these components and the identification system used for the steel structure. Consequently, a property set is required for steel single and assembly identifications (see Table 5.4).

Based on the AISC EM.11 standard, the *Pset_Identification* property set and two selected properties are applied to represent the **Piece Mark** and the **Main Piece Tag**. The latter is a Boolean value indicating whether the steel element is a main piece.

TABLE 5.4: Pset_Identification

Property Set	Property	Description
Pset_ Identification	Piece Mark	A number or string that identifies the single part (usually a beam or plate). Two pieces with identical geometry carry the same single part number, whereas different geometries have different numbers.
	Main Piece Tag	Indicates whether the piece is the main piece (true) or not (false).

Digital transformation, and more specifically the processes of reuse and deconstruction, necessitate the identification of the built-in elements that are linked to the specification of the element in question. The information thus obtained can then be used to facilitate an assessment of the potential for reuse. Therefore, a property set *Pset_DigitalIdentification* is established (see Table 5.5). In Figure 5.1 this Pset is abbreviated with Pset_DIdentification.

TABLE 5.5: Pset_DigitalIdentification

Property Set	**Property**	**Description**
Pset_ Digital Identification	ID Code	Identification code for the digital item. Can be used for referencing or authentication.
	Verification Method	The method or system used to verify the digital identity or signature, such as encryption or checksum.
	Code Type	The type of code used for digital signatures, such as a 128-bit barcode or QR code.

5.3.4 Tolerances

Referring to the previously described approach to representing tolerances (see Section 5.2.5), this section introduces minimum and maximum tolerance properties for each profile dimension value and each fabrication feature property relevant to the execution of the process (e.g., diameter, width). A DIN reference property is also included to store the relevant DIN standard in which the tolerance specifications are defined.

At the beginning of this thesis it was highlighted that the actual deviations of steel profiles from their planned dimensions can be significant. As a result, many machines measure the profiles prior to production in order to adjust the fabrication parameters if necessary. This is particularly important in robotic fabrication, where failure to account for real-world deviations could lead to collisions between the robot and the workpiece. A current challenge is that measurements are often not stored, making it difficult to track deviations and improve fabrication accuracy. To address this, in addition to modelling the defined tolerances for specific fabrication features, a measured value property is incorporated into each relevant property set. This allows actual measurements to be stored and used for future analysis, process optimisation and quality control. By making this data digitally available, it can be compared against tolerance specifications, improving quality assurance and fabrication efficiency.

The implementation of these concepts can be seen in the next Sections 5.3.5 and 5.3.6.

5.3.5 Additional profile properties

The following set of properties (see Table 5.6) is proposed to be attached to the profile definition to store the measured values of the profile prior to fabrication and possibly adjust fabrication parameters or robot paths. The properties presented here serve as an example of those used for profile description, as additional tolerances, such as for web thickness, also exist. For each of these parameters, minimum and maximum values are defined based on the nominal dimension. For simplicity, the table only includes the tolerance for height.

TABLE 5.6: Pset_AdditionalProfileProperties

Property Set	Property	Description
Pset_ Additional Profile Properties	Measured Height	Actual measured height of the profile.
	Measured Width	Actual measured width of the profile.
	Measured Length	Actual measured length of the profile.
	(Height) Tolerance Min	Permitted negative deviation of the actual value from the planned value.
	(Height) Tolerance Max	Permitted positive deviation of the actual value from the planned value.
	DIN Reference	Applicable DIN standard for the profile.

5.3.6 Fabrication process

To enable IFC-based fabrication, it is essential to represent both general information applicable to all features and feature-specific attributes. Therefore, it is proposed to define a universal property set that includes properties relevant to all features. This would encompass attributes such as X/Y coordinates and process constraints, specifying whether a particular process is permitted or prohibited for a given feature.

Table 5.7 presents the universal property set *Pset_FabricationProcess*, which applies across all fabrication features.

TABLE 5.7: Pset_FabricationProcess

Property Set	Property	Description
Pset_ Fabrication Process	Position	Coordinates of the process step, typically represented by vertex x and vertex y.
	Process Required	Indicates the processes necessary to complete the fabrication step (e.g., welding, cutting).
	Process Forbidden	Specifies the processes that should not be used in this step (e.g., no welding allowed).
	Comment	Additional notes or remarks regarding the fabrication process step.

In addition to this general property set, feature-specific property sets are introduced to provide more detailed descriptions of individual features. These primarily include geometrical properties such as diameter, width, or depth.

Holes

Feature specific properties are linked to the relevant IFC feature. Processes where material is removed are defined as *IfcVoidingFeature* with predefined types. Based on the specifications for the different hole types originally in the DSTV file, each hole type is given a property set. As illustrated in Table 5.8, the properties of a through hole are displayed upon incorporating the defined concept of tolerances and measured values. The same concept is defined for a threaded hole, a blind hole and a countersink hole. The respective tables can be found in the Appendix (see Tables B.1, B.2 and B.3).

TABLE 5.8: Pset_ThroughHole

Property Set	**Property**	**Description**
Pset_ ThroughHole	Diameter	Planned diameter of the through hole.
	Measured Diameter	Actual measured diameter of the through hole after fabrication.
	Diameter Tolerance Min	Permitted negative deviation of the actual value from the planned value.
	Diameter Tolerance Max	Permitted positive deviation of the actual value from the planned value
	DIN Reference	DIN standard defining the tolerance for the through hole.

Cutout

An alternative fabrication process is to be defined with specific properties for fabrication, based on the DSTV file in the *Pset-Cutout*, which is displayed in Table 5.9. This also illustrates that placeholders exist not only for the planned width of a cut-out (**Width**), but also for recording the measured width after fabrication (**Measured With**). This enables a direct comparison between the actual value and the allowable deviation from the planned value as defined by the tolerance specification (**Width Tolerance Min** & **Width Tolerance Max**).

TABLE 5.9: Pset_Cutout

Property Set	**Property**	**Description**
Pset_ Cutout	Width	Planned width of the cutout feature.
	Measured Width	Actual measured width of the cutout after fabrication.
	Width Tolerance Min	Permitted negative deviation of the actual value from the planned value
	Width Tolerance Max	Permitted positive deviation of the actual value from the planned value.
	Height	Planned height of the cutout feature.
	Measured Height	Actual measured height of the cutout after fabrication.
	Height Tolerance Min	Permitted negative deviation of the actual value from the planned value.
	Height Tolerance Max	Permitted positive deviation of the actual value from the planned value.
	Depth	Planned depth of the cutout feature.
	DIN Reference	DIN standard for the tolerance for the feature.

Resource specifications

In addition to the resource specifications for the machine, where the existing *Pset_Manufactuer-TypeInformation* can be applied, a dedicated property set for tool specifications has been defined to capture tool-specific attributes (see Table 5.10).

TABLE 5.10: Pset_ToolInformation

Property Set	**Property**	**Description**
Pset_ Tool Information	Tool ID	Unique Identifier of the tool.
	Tool Type	Type of tool (e.g., drill, cutter, etc.).
	Manufacturer	Name of the manufacturer of the tool.
	Serial Number	Serial number assigned for identification.
	Model Number	Model number or product code for the tool.
	Year Of Manufacture	The production year of the tool.
	Purchase Date	Date when the tool was purchased.
	Operating Hours	Number of hours the tool has been in operation.
	Operational Status	Current operational status of the tool (e.g., operational, under maintenance, out of service).

5.3.7 Process parameters and feedback

In addition to the defined geometry for features such as drilled holes, it is possible to model additional process-specific parameters in order to describe the fabrication operation in more detail. The property set can be linked to the *IfcTask* and the machine or tool. With the *IfcTaskTime* the planned and actual time of the process can be tracked.

These process parameters have two different types of values: planned process values, representing the predefined settings prior to execution, and measured process feedback values, capturing the actual data recorded during execution.

The following Table 5.11 shows the parameters required to drill a hole, incorporating the concept of planned and measured feedback parameters. It should be noted that not every process necessarily has a sensor available to measure the specific parameters. Therefore, the attributes in the table are not defined as mandatory. However, in the context of sustainability aspects, process tracking or the integration of real Life Cycle Assessment (LCA) values from the fabrication process, it is proposed to define these attributes as mandatory in the future to ensure that the data is systematically collected and monitored.

As defined in Chapter 4 different processes require different process parameters. The tables in the Appendices A.6 and A.7 show the defined parameters for each process, which can be modelled accordingly in a specific property set for the fabrication process.

TABLE 5.11: Pset_DrillingProcess

Property Set	Property	Description
Pset_ Drilling Process	Spindle Speed Planned	Planned rotational speed of the drill spindle.
	Spindle Speed Measured	Actual measured rotational speed of the spindle.
	Feed Rate Planned	Planned feed rate of the drill bit.
	Feed Rate Measured	Actual measured feed rate of the drill bit.
	Drill Pressure Planned	Planned pressure applied by the drill.
	Drill Pressure Measured	Actual measured pressure applied by the drill.
	Coolant Usage Planned	Planned volume of coolant used.
	Coolant Usage Measured	Actual volume of coolant used during the drilling process.

5.4 Ensuring compliance with IFC modelling standards

To ensure that IFC models are structured correctly and consistently across different applications, mvdXML (Model View Definition XML) and buildingSMART IDS (Information Delivery Specification) provide tools to validate and enforce specific modelling requirements. buildingSMART provides the mvdXML standard, which was created as part of IFC 4 development (mvdXML 1.1 [127]) but is no longer officially developed by buildingSMART. Although updated version of mvdXML [212] are available, no specific Model View Definitions (mvdXMLs) for steel fabrication have been established. The Information Delivery Specification (IDS) standard by buildingSMART was also not considered, as it was not the primary focus of this work. The proposed IFC Fabrication View of this thesis serves as a foundation for defining an mvdXML in future research, enabling the enforcement of steel fabrication process requirements. This approach ensures that IFC models adhere to industry standards and can be consistently applied across different users and software environments.

5.5 Response to RQ 2

This chapter explored the existing entities and properties within the IFC 4x3 schema to determine how a data model should be designed to be functioning as a successor to the DSTV-standard. Exceeding the basic information saved in the DSTV-NC file the IFC representation support the re-use of steel components while providing the necessary data for automated fabrication processes in steel fabrication. The analysis aimed to bridge the gap between general IFC-based collaboration in construction projects and the specific requirements of steel fabrication.

The results show that IFC 4x3 currently lacks dedicated support for steel fabrication processes. However, existing concepts such as *IfcProfileDef*, *IfcTask*, and *IfcVoidingFeature* can be used to describe steel components and integrate fabrication processes into an IFC-based workflow. In accordance with the buildingSMART guidelines, which allow the extension of property sets but prohibit the introduction of new classes or relationships, this chapter introduced new property sets that comprehensively address the identified requirements for steel fabrication. However, this is also a limitation, as the current IFC schema needs to be adapted to the specific needs of steel fabrication, rather than allowing for the definition of entirely new entities or relationships. As a result, the approach taken here relies on existing IFC concepts and structures, which may not fully capture the complexity and specificity of fabrication processes. Future work may explore whether additional standardisation efforts within the IFC or alternative extensions could provide a more accurate representation of steel fabrication workflows.

Although this research does not define all possible fabrication properties and processes in detail, it provides a structured framework for the representation of selected processes into the Industry Foundation Classes (IFC). It supports standardisation activities within bauforum-stahl e.V. and buildingSMART. This allows automated fabrication steps to be executed based on IFC files containing all relevant information on geometry, fabrication features, processes and machines, some of which was previously summarised in DSTV-NC. Future work may extend this approach by including additional fabrication processes and refining the property sets to enhance the practical applicability of the proposed model in steel construction.

Chapter 6

Development - DSTV Ontology

Building upon the foundational principles and requirements of the data model with respect to the use cases and their initial implementation according to the IFC 4x3 schema, as presented in the previous chapter, the following sections demonstrate how the data model can be realised through ontologies as a basis for Linked Data. Following the overview of this thesis's contributions to representing the DSTV-NC standard using Linked Data (see Section 6.1), Section 6.2 begins with an analysis of existing ontologies to identify those that can be reused and applied effectively. It then describes the new classes, object properties and data properties defined in the DSTV domain ontology, demonstrating how these fully meet the requirements.

An initial approach to the ontology for steel pre-fabrication was presented in [21], to which I contributed through the requirements analysis, initial modelling approach, and by supporting the physical testing and evaluation phases. Building upon this foundation, the present work significantly extends the original requirements and implements a refined and comprehensive domain ontology tailored to the updated use cases. The approach follows the proposed methodology for ontology development by Noy et al. [6].

6.1 Ontologies for steel pre-fabrication

One of the key principles of Linked Data is the reuse of existing ontologies. Research into the current state of ontologies for steel fabrication reveals a significant lack of ontology approaches for this domain (see Chapter 3). None of the available solutions fully address the Competence Questions (see Table 4.2), especially in a practical way for robotic applications. However, ontologies already exist for concepts related to the construction process and elements, as well as for metadata relating to these elements. These ontologies are applicable and have been adopted to represent steel pre-fabrication requirements. Using these ontologies is intended to improve interoperability.

Figure 6.1 provides an overview of the defined requirements for the data model, as well as a distinction between existing concepts from established ontologies (green boxes) that can be reused to represent data related to steel pre-fabrication, concepts that were already published in the initial version of the DSTV ontology (light blue boxes), and those newly defined within the scope of this work (dark blue boxes).

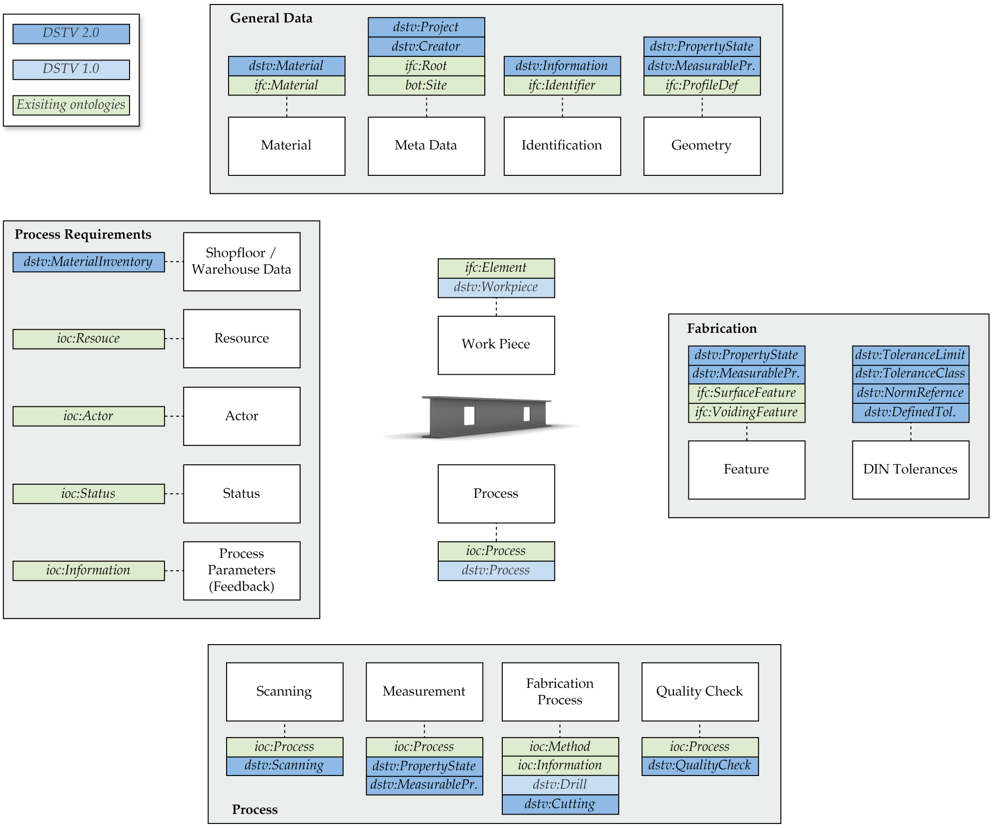


FIGURE 6.1: Summary of ontologies for steel pre-fabrication: existing concepts shown in green, initial implementation in DSTV Ontology 1.0 in light blue, and newly defined concepts in dark blue, representing a semantic successor to the DSTV-NC standard

6.2 Implementation of requirements as DSTV ontology

The following sections provide a detailed description of how the existing concepts are used for pre-fabrication, as well as defining the new classes, attributes and relations.

Section 6.2.1 provides an overview of the linkage between the DSTV ontology and the existing BOT and ifcOWL ontologies. Section 6.2.2 focuses on the representation of general data, including metadata and material properties. A new an important concept for deviation management and quality control is the concept of measurable properties, which is introduced in Section 6.2.3. Section 6.2.4 elaborates on the modelling of fabrication processes, followed by Section 6.2.5, which illustrates the connection to fabrication features. Finally, Section 6.2.6 addresses the modelling of defined tolerances according to standards, ensuring their integration with profiles and processes. The visual representations of the ontology design are sketched with the drawio Plug-in Chowlk [137], which is a tool to transform ontology conceptualisations made with diagrams.net into OWL code.

The following classes, and relationships represent the final state. Adjustments were made during the demonstrator's preparation, reflecting an iterative cycle within the DSR methodology, transitioning from *Demonstration* back to *Design* and *Development*.

6.2.1 Overall link to BOT and ifcOWL

In order to emphasise the correlation with prevailing ontologies and the standardised description of construction elements and processes, Figure 6.2 illustrates the relationship between the DSTV, the ifcOWL and BOT ontology. The BOT ontology is a lightweight ontology developed to represent the spatial and topological relationships of buildings. It provides a high-level, abstract framework that focuses on the relationships between different parts of a building, without going into extensive detail about the properties of building elements [93]. The ifcOWL ontology provides a formal representation of the IFC schema in the Web Ontology Language (OWL) [105]. It enables general building elements and their potential subtypes (e.g., Beam, Column, and various profiles) to be used for describing steel profiles in fabrication contexts. The ifcOWL ontology can be employed in several ways: first, it offers a detailed and interoperable format for the exchange of BIM data; second, it supports semantic querying and reasoning over rich building information; and third, it enables integration with other domains, such as structural analysis or steel fabrication.

The building element is topologically defined within the BOT ontology as *bot:Element* and is declared as equivalent to (*owl:equivalentClass*) the *ifc:Element*, with additional objects and relations linked to it. The *ifc:BuildingElement* is defined as a subclass of *ifc:Element* (*rdfs:subClassOf*) and is declared as equivalent to (*owl:equivalentClass*) the *dstv:Workpiece*. The same applies to *ifc:Beam* and *ifc:Plate*, which are defined as *rdfs:subClassOf ifc:BuildingElement* in the ifcOWL ontology and are declared as equivalent (*owl:equivalentClass*) to *dstv:Beam* and *dstv:Plate*, respectively. These, in turn, are subclasses (*rdfs:subClassOf*) of *dstv:Workpiece*. The general profiles defined in *ifc* can now be further specified in the DSTV ontology. The class *dstv:ProfileDef* is defined as a subclass of *ifc:ProfileDef* (*rdfs:subClassOf*), which allows it to inherit all definitions from *ifc:ProfileDef*. Additionally, *dstv:ProfileDef* can be extended with domain-specific

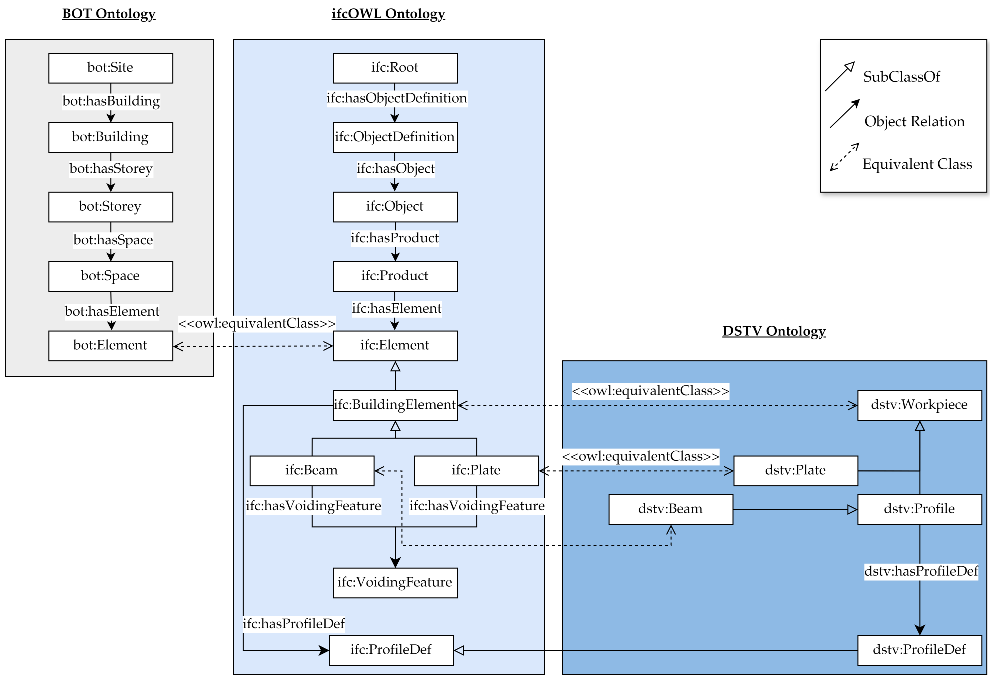


FIGURE 6.2: Representation of the link from DSTV to ifcOWL and BOT ontology

properties. The data from the DSTV-NC file can be represented using classes, object properties (*owl:ObjectProperty*), and data properties (*owl:DatatypeProperty*). As described in Chapter 5, steel fabrication processes involving extraction are defined using the class *ifc:VoidingFeature*. A link to this element can also be established when the different features are described in the DSTV ontology. At this point, it becomes clear that IFC and OWL & RDF(S) are not mutually exclusive but can complement each other. IFC can be represented in ifcOWL, enabling it to be semantically linked with additional domain-specific information.

6.2.2 General data

Based on the header information of the original DSTV-file classes, object and data properties have been defined. Figure 6.3 illustrates the visual representation of the ontology. The left part highlights the link to the ifcOWL ontology. The purple boxes highlight the classes that have already been published in the DSTV ontology version 1.0. The white boxes indicate the use of the ifcOWL concept, while the green boxes represent newly defined classes and properties.

The upper part (grouped in the grey box), linked to the *dstv:Workpiece*, shows the classes *dstv:Project*, *dstv:Creator*, *dstv:Information*, *dstv:Material* and *ifc:Identifcation*. The first box of the row shows the class with an object relation to the *dstv:Workpiece* class. The linked data properties of the class are shown in the boxes below. The *dstv:Material* is (*rfds:subClassOf*) of the

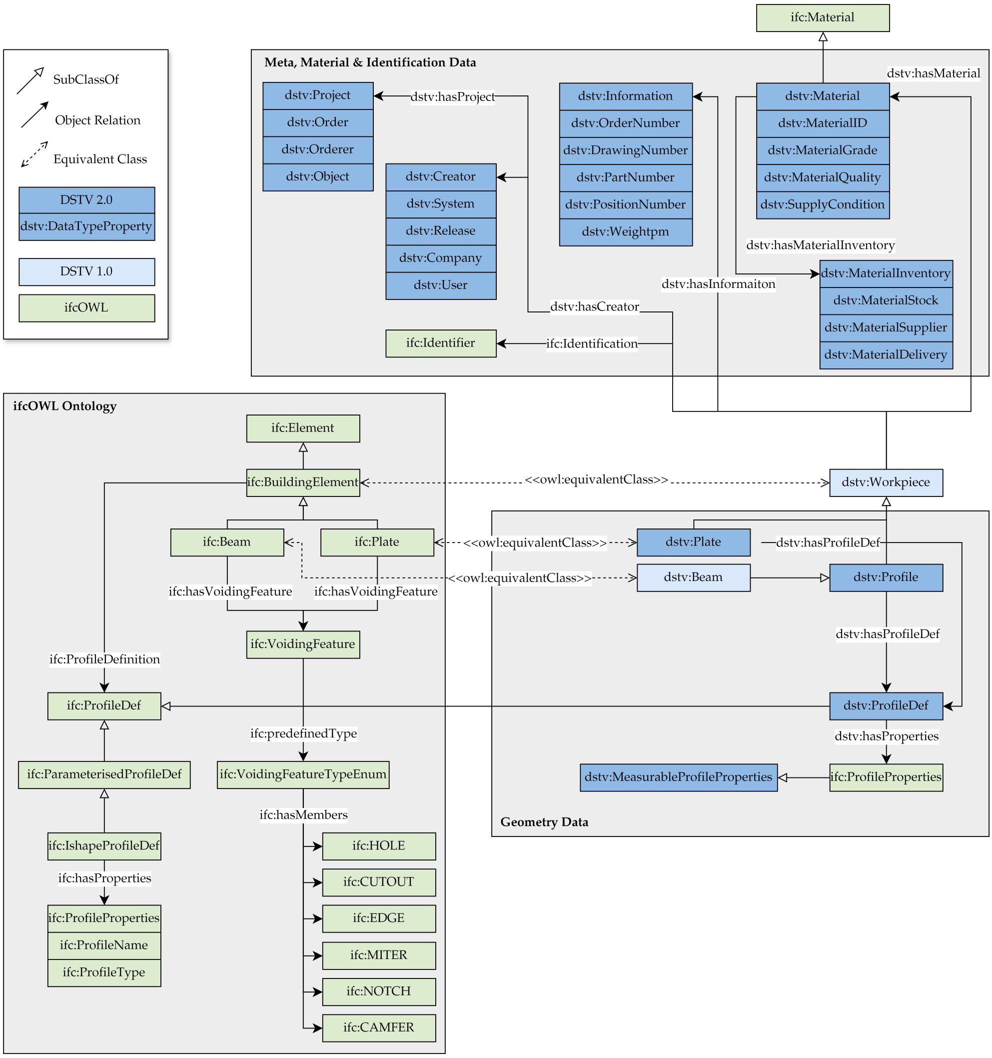


FIGURE 6.3: Representation structure of general data of DSTV ontology

ifc:Material. Defined properties can be inherited, but specific data properties for steel fabrication are added. In addition, the *dstv:Material* has a *dstv:MaterialInventory* where shop floor specific data can be stored. These are the available quantity (*dstv:MaterialStock*) of the material used or planned for the order, the supplier (*dstv:Materialsupplier*) and the delivery date of the material (*dstv:MaterialDelivery*).

As illustrated in the lower part of Figure 6.3, the geometry definitions of the steel parts are delineated as follows. The *dstv:Beam* is a subclass (*rfds:subClassOf*) of the *dstv:Profile*. Both, *dstv:Plate* and *dstv:Profile* have a *dstv:ProfileDef* which is a *rfds:subClassOf* of the *ifc:ProfileDef*. Defining a subclass in a new ontology ensures semantic consistency, as the subclass remains logically aligned with the original ontology. It inherits all general properties from its superclass while adding new domain-specific properties. In this case, all the profile definitions available in ifcOWL can also be used to describe the part within the steel domain (e.g., *ifc:IshapeProfileDef* with specific properties). The reusabilty promotes interoperability and avoids duplication of effort. Furthermore, it provides flexibility by allowing new domain-specific data properties to be introduced without modifying the original ontology.

6.2.3 Concept of measurable properties

As identified during the requirement analysis, robotics and digital applications generally necessitate not only static property values but also dynamic values and the capability to store the actual measured value of a property. Figure 6.4 illustrates the developed concept that allows properties to exist in multiple states. The general class, *dstv:MeasurableProperties*, is subdivided into two subclasses: *dstv:MeasurableProfileProperties* and *dstv:MeasurableProcessProperties*. These subclasses are linked to the respective profile and process properties via *rfds:subClassOf*, ensuring that each property can maintain multiple values while inheriting all definitions from the superclass. The initial value of a property is typically represented as *dstv:hasPlannedProperty*. During the fabrication process, as scanning and measurement occur, this value can be updated with the measured data (*dstv:hasMeasuredProperty*). Additionally, if adjustments are required, an adapted value can be assigned via *dstv:hasAdjustedProperty*. Such adaptations may be necessary due to predefined tolerance values, represented as *dstv:hasDefinedToleranceProperty*. A more detailed explanation of this mechanism is provided in Section 6.2.6.

The concept of *dstv:MeasurableProperties* (abbreviated as *dstv:MeasurablePr.*) is generally applicable and, as shown in the overview in Figure 6.1, is linked to various categories, including 'Geometry', 'Feature', and 'Measurement'.

All four property states are associated with the class *dstv:PropertyState*, which contains two key attributes, using established onotologies: the actual value (*schema:Value* - [213], *qudt:Value* - [214]) and the timestamp indicating when the value was recorded (*prov:GeneratedAtTime* - [215]). This approach facilitates a comprehensive representation of the requirements for specific properties and ensures their proper documentation.

Building on Figure 6.3 the concept establishes that *IfcProfileProperties* has an equivalent class, *dstv:hasMeasurableProfileProperties*. This equivalence ensures that commonly defined properties in IFC, such as length and width, can have a planned value while also supporting additional values. Since these properties belong to the same class, they align with the newly developed property concept. This approach enables greater flexibility by allowing the integration of measured properties. Consequently, the actual dimensions of the workpiece, obtained through the scanning process, can serve as the foundation for robotic fabrication.

6.2.4 Process

A primary objective of the developed DSTV ontology is to represent process-specific requirements. It extends beyond the general data and definitions of steel elements, which are partly

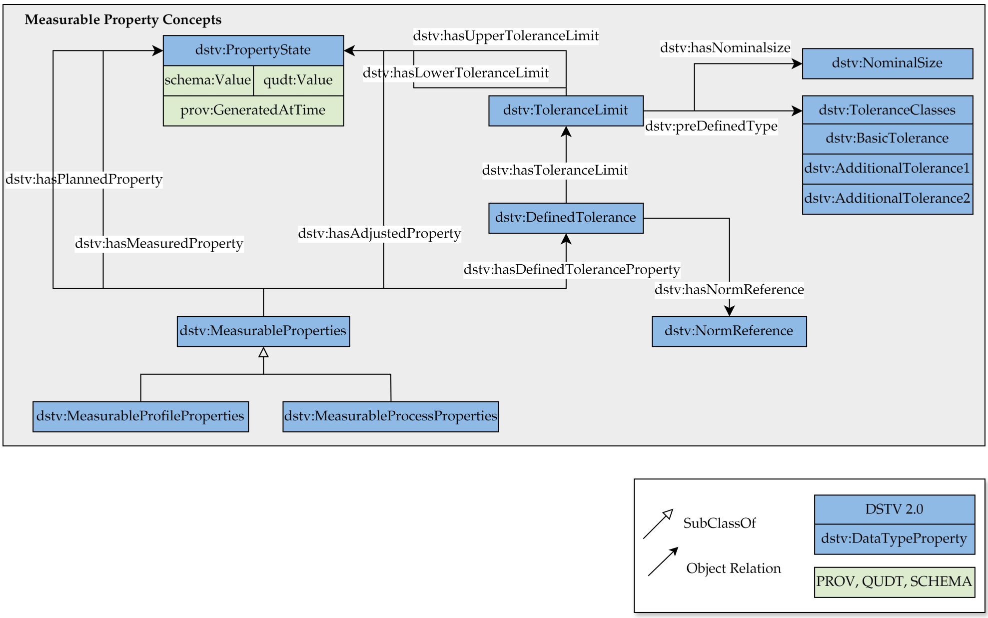


FIGURE 6.4: Representation concept of measurable properties

defined in the ifcOWL ontology. In the context of the selected use cases, the classes and properties assume particular significance for the second and third use cases, i.e. the data that is transmitted to the machine and the data that is fed back from the processes that have been executed.

Figure 6.5 provides a visual representation of the required classes and properties for describing steel fabrication processes. The class *dstv:Process* is a subclass of *ioc:Process*, as defined by the IoC ontology, which provides a comprehensive framework for process description. This ontology encompasses the fundamental classes *ioc:Resource*, *ioc:Actor*, *ioc:Method*, *ioc:RequirementSet*, *ioc:Schedule* and *ioc:Status*. These elements, in conjunction, serve to delineate the resource employed for the execution of the process, the entity responsible for its execution, the method by which it is executed, the defined requirements, the timing of its execution, and its status (*ioc:isReady*, *ioc:isStarted*, *ioc:isPaused*, *ioc:isFinished* or *ioc:isCancelled*). The time is also defined via this status. In the context of using robots as a resource, manufacturing standards provide a unified description of the machine in question (e.g., OPC UA or Asset Administration Shell (AAS)). This description can be employed in the future to apply the terms and definitions to robots and machines employed in steel fabrication. Within the IoC concept, each process possesses both an input element (*ioc:hasInputElement*) and an output element (ioc:hasOutputElement) to which specific information can be linked. A process can be both a parent and a child process, as well as an upstream (predecessor) or downstream (successor) process.

In the context of steel fabrication, there are now specialised processes that can be represented as

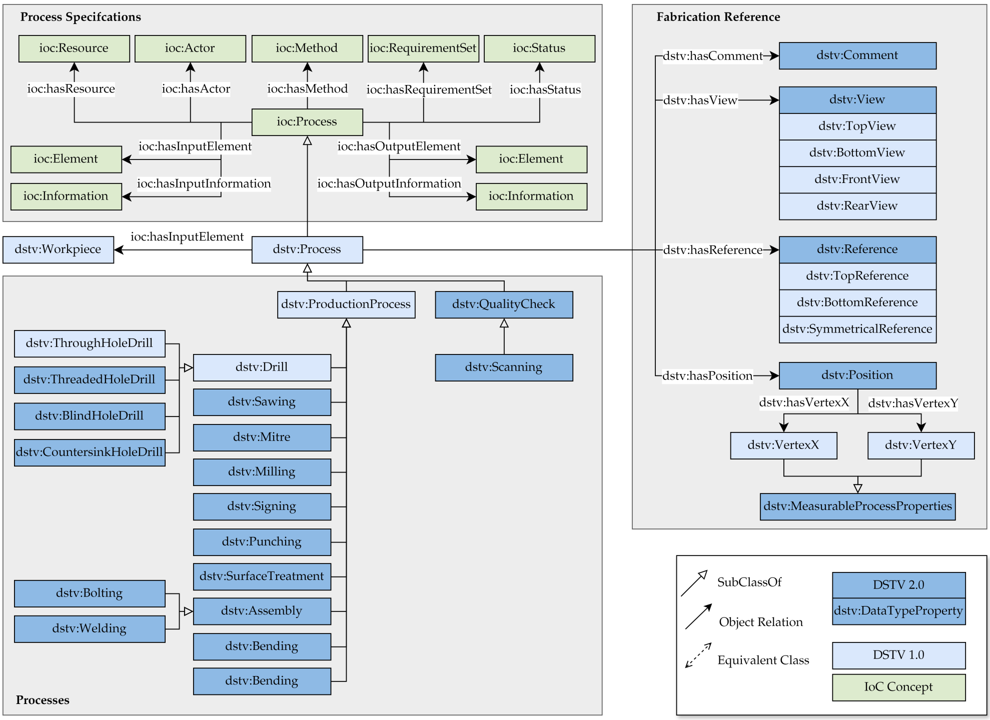


FIGURE 6.5: Representation structure of process data of DSTV ontology

a subclass of *dstv:Process* and, accordingly, also as a subclass of *ioc:Process*. These processes all inherit the defined schema of the IoC ontology, thereby ensuring the consistency and integrity of the data. To every process a resource can be assigned to which can have further information such as name, type, manufacturer, maintenance status. Within the *ioc:RequirementSet*, prerequisites can be defined or fabrication processes for the respective feature (e.g., drilling) can be specified or prohibited with *ioc:Method*. It can also contain instructions and data related to robotic fabrication. With *ioc:hasInputInformation* and the class *ioc:Information* data for the fabrication it self can be defined, such as specific process parameter that should be asserted during the processing on the machine. With *ioc:hasOutputInformation* process feedback can be stored, such as tool speed.

The drilling process, meanwhile, comprises specific subclasses, including *dstv:ThroughHoleDrill*, *dstv:ThreadedHoleDrill*, *dstv:BlindHoleDrill* and *dstv:CountersinkHoleDrill*. In the context of steel construction, assembly can be undertaken using different processes, such as *dstv:Bolting* or *dstv:Welding*. The processes defined here reflect the current requirements. The ontology's strength lies in its adaptability, allowing for the incorporation of future manufacturing processes, thereby ensuring standardisation by stakeholders. The quality control process can be facilitated by a scanning sub-process, in addition to visual inspections.

As illustrated on the right side of Figure 6.5, the fabrication reference delineates the general validity of the classes and properties that are applicable to all processes. With regard to the DSTV file, it is imperative to ascertain the side of a steel element on which the processing is to be conducted (*dstv:TopView*, *dstv:BottomView*, *dstv:FrontView, dstv:RearView*). Moreover, the reference for the machining is defined. The decision as to whether to place the feature on the top edge, bottom edge or symmetrically to the cut is critical when measuring the real part on the machine before machining and comparing the measured values with the planned values. For example, if the holes are to be symmetrical on the surface, the position of the holes must be adjusted. All fabrication features have a *dstv:VertexX* and *dstv:VertexY* value. These classes are a subclass to the *dstv:hasMeasurableProcessProperties* class. Consequently, the vertices can also exhibit four different states.

In Figure 6.1 only the classes *dstv:Drill* and *dstv:Cutting* are visually linked to the requirement 'Fabrication Process', as these are the two relevant processes in the use case of this thesis. However, as demonstrated within this section, the other steel pre-fabrication processes are also defined as classes in the second version of the DSTV ontology.

6.2.5 Fabrication features

The fabrication feature is a central part of the DSTV-NC standards as it describes what is the output after the processing. To represent this in OWL, the basic schema of the IoC ontology is applied. Here, each process has an input element (in this case, the unprocessed workpiece) and an output element (the fabrication feature). Figure 6.6 displays the structure for linking the processes to the output element and also back to the IFC schema.

Figure 6.6 demonstrates the modelling concept for a borehole and an internal contouring. The *dstv:ThroughHoleDrill* is a subclass of the *dstv:Drill* process, with the *dstv:ThroughHole* as an output element on the one hand and process properties on the other. The output element *dstv:Throughole* is linked to the voiding feature defined in ifcOWL via the *owl:equivalentClass*. The throughole can be assigned to the *ifc:Hole*, as predefined types are available. It has previously been defined that the position, i.e. the Vertex X and Vertex Y, is attached to all processes, and thus these are not listed again here. The specific property for this output element is listed here (*dstv:Diameter*, while for other holes, such as the threaded hole, further properties are defined (e.g., thread direction, thread type). The *dstv:ProcessProperties* is a subclass to the general *dstv:MeasurableProcessProperties* class, thus allowing for four distinct states in the properties. This concept is similarly applicable to a cutting process, which is represented by an internal contouring process. The output element of this process is an internal contour, which is linked to the *ifc:Cutout*. This sub process also has specific process properties that are a subclass to the *dstv:MeasurableProcessProperties* class.

The provided examples here form as guideline to know how the processes and fabrication features are linked with the *ioc:OutputElement* and their specific process properties. The same concept can also be applied to other processes besides the production processes. Quality control or scanning can also have output elements, such as the scan or a report but also properties and relations to defined tolerances. Therefore, the general concept of *ioc:Process* is linked to various categories in the overview graphic in Figure 6.1.

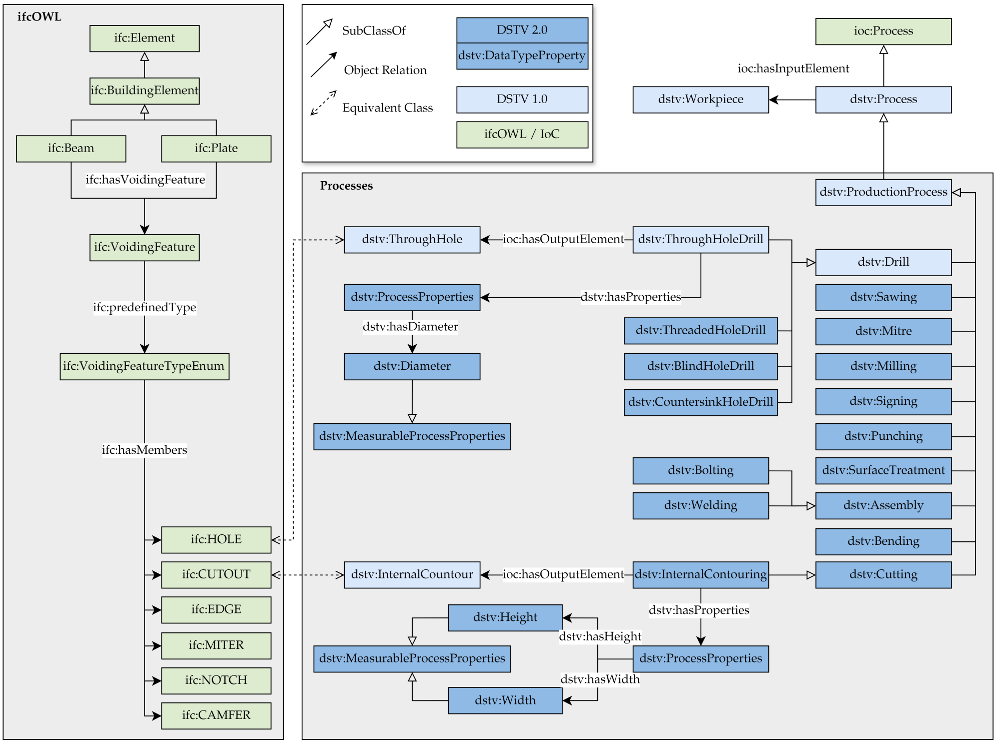


FIGURE 6.6: Representation structure of fabrication feature of DSTV ontology

6.2.6 Tolerances

Tolerances play an important role in steel construction. Adherence to tolerances is important not only for quality control, but also for machining, especially robotic machining, to ensure that the steel components are standardised and have the correct properties, and to ensure that the machines do not collide with the workpiece.

Digitalisation facilitates machine readability of information on defined tolerances for various profiles, contingent on production. This approach offers numerous advantages, including the direct processing of tolerance values by machines and software, thereby enabling faster process execution. Moreover, documented tolerance deviations facilitate identification of material, process and machine dependencies, paving the way for enhanced quality analysis. To be able to extend the current single value of 'to' in the DSTV-NC file, imperative to map the existing tolerance values in the ontology and attach them to the workpiece and the fabrication feature.

In this section, the concept of the measurable property class and its states is referred to (see Figure 6.4). One of the states that a property can have is *dstv:hasToleranceProperty*. As the class has the subclasses *dstv:MeasurableProcessProperties* and *dstv:MeasurableProfilesProperties* the defined production tolerances which are linked to the properties of the profile as well as the defined process tolerance of the process / output element can be represented. To be able to trace the origin of the norms and the defined values, each tolerance property has a *dstv:NormReference*. Within this thesis, different DIN standards have been analysed specifying the constraints of different profiles. DIN EN 10034, DIN EN 10279, DIN EN 10056-2 and DIN EN 10219-2 were considered. At this point, only hot rolled profiles were integrated, but the same structure can also be applied to cold rolled profiles and the specific values. The tolerances are defined in relation to the respective profile and the specific criteria that must be met (e.g., profile height, flange width or perpendicularity). However, the values themselves are defined for different nominal dimensions (dstv:NominalSize). The tolerances itself are defined within a range, resulting in two different property states, (*dstv:hasUpperToleranceLimit*) and *dstv:hasLowerToleranceLimit*.

DIN 1090-2 defines the specified tolerances for dimensions and geometry of fabrication processes and features. This ensures the accuracy of the fit and facilitates the correct assembly of components during installation. There are three different tolerance classes (*dstv:BasicTolerance*, *dstv:AdditionalTolerance1* and *dstv:AdditionalTolerance2*). Basic tolerances are critical to the safety-related dimensions of a component and must be strictly adhered to. They affect the strength, accuracy of fit and functional integrity of the component. Additional tolerances affect secondary characteristics and are less critical for functionality and safety, but important for precision manufacturing and aesthetic requirements.

In general, the tolerances for the respective properties are predefined and remain fixed rather than being continuously modified. They are modelled once as part of a knowledge base. A key advantage of this approach is that the underlying structure remains consistent, even if specific values change or new constraints for profiles or fabrication processes are introduced. These constraints can be attached to the corresponding properties. For quality control purposes, the predefined tolerances can then be compared with both the measured and planned values. Currently, the primary focus is on production tolerances and the tolerances of the output element of a given process. However, in the future, it may also be necessary to define requirements related to the process itself, such as maintaining a specific temperature, welding speed, or other process parameters. These requirements can also be incorporated into the model as needed.

In the overview Figure 6.1 only four new defined classes are linked to the category 'DIN Tolerances' to show the new concept developed within this thesis. However, the full scale, as described in this section, acts as a detailed successor to the simple 'to' field in the DSTV-NC file, covering not only the profile, but also feature-related defined tolerances.

6.3 Final DSTV ontology

This chapter focused on developing the individual components of the DSTV ontology, building on the first published version. It incorporated the requirements for digitised, interconnected and robotic fabrication processes, providing a valuable addition to the limited information currently available in the DSTV-NC standard. To ensure interoperability and prevent redundant developments, existing concepts have been integrated. Figure 6.7 presents a visual representation of the `.ttl` file, which defines the ontology using the correct OWL structure in RDF triplets. The visualisation highlights different elements of the ontology: dark blue nodes represent existing schemas, blue nodes indicate newly introduced classes, and green nodes correspond to data properties.

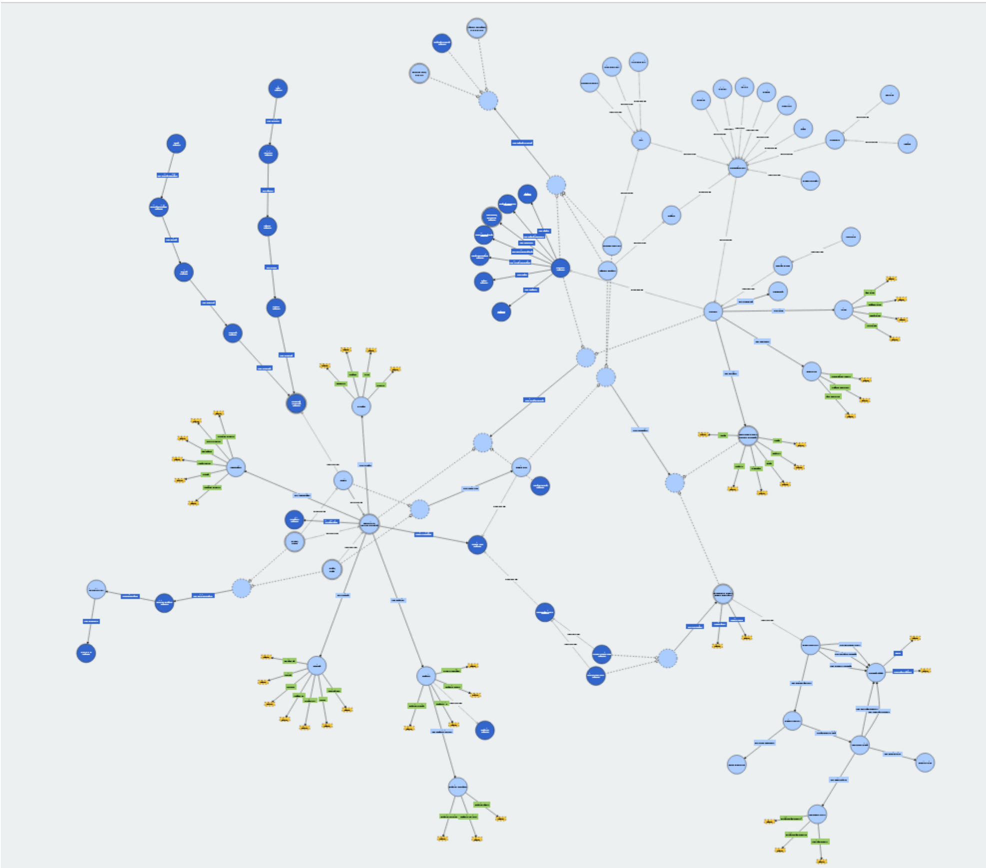


FIGURE 6.7: Ontology representation in VOWL

The ontology was initially modelled using the drawio Plug-in Chowlk [137], then corrected using Protégé [136], and finally checked with the 'OOPS!(Ontology Pitfall Scanner!' [216]. This tool ensured that the ontology was modelled in accordance with the defined rules.

In total, the ontology consists of 70 classes, 30 object and 24 datatype properties and is accessible [25].

6.3.1 Ontology evaluation

With reference to Chapter 2 and the ontology evaluation criteria proposed by Gómez-Pérez [133], the following section evaluates the extended ontology.

The initial criteria *correctness* and *completeness* will be assessed in Chapter 8, as this can only be determined in relation to the application of the ontology to the use case and the subsequent verification of the capacity to respond to the defined competency questions. However, it can be stated that at this point not all existing steel fabrication specifications are modelled. The primary concern relates to the assembly processes, which have already been initiated in the following work [211]. Given the focus on domain-specific ontology development — restricted to concepts pertinent to steel fabrication — data has been modelled with *precision* and *coherence*. The introduction of the concept of measurable properties enables the flexible assignment of different property value states to process and profile properties, without the need to define redundant classes. Although the DSTV ontology has not yet been *reused* in other contexts, the approach of measurable properties, along with digitally available tolerance specifications for quality control, offers potential applicability across other domains. By adopting the concept of *ioc:process* to represent fabrication procedures, the ontology allows for the modelling of a broad range of operations beyond drilling and plasma cutting. Furthermore, evolving process requirements — such as the need to store input and output information — can be accommodated with ease. The criteria of *relevance*, *acceptability*, and *effectiveness* must be evaluated in collaboration with industry stakeholders and within scaled industrial processes.

6.3.2 Response to RQ 2

This thesis presents an ontology-based data model designed to support the reuse of steel components and provide the necessary data for automated fabrication processes in steel construction. The model is structured around a modular and extensible ontology that integrates existing concepts such as ifcOWL and IoC ontology, while introducing new classes, object properties and data type relationships to address fabrication-specific requirements.

The core of the ontology is the *ioc:process* entity, which acts as a central link between the different data categories identified in the conceptual model. To ensure flexibility and reusability, a universal property model is introduced that allows each property to exist in multiple states, including its planned state, actual measured state and acceptable tolerance range. This enables both documentation and digital quality control. The ontology also incorporates hierarchical relationships to represent dependencies between different fabrication steps and machine interactions. By defining standardised object properties, fabrication processes can be linked to materials, tools, equipment and quality control parameters. In addition, the model is designed to support interoperability with existing IFC-based workflows, ensuring that data can be integrated into broader digital construction environments. The proposed ontology thus provides a structured, process-driven approach to linking steel fabrication data, ensuring that it supports both component reuse and automated fabrication processes. In the subsequent chapter, the application and evaluation of the extended DSTV ontology is conducted through the utilisation of two distinct fabrication processes. However, further industry application and validation as well as standardisation efforts are required to ensure its full adoption.

Chapter 7

Demonstration

Following the *Demonstration* step of the overall applied research methodology (see Chapter 1) and the completion of the final step of the combined data model methodology for the *Design* and *Development* phases, a case study was selected to assess the applicability of the Steel Fabrication View and the updated DSTV ontology for robotic fabrication. First, Section 7.1 explains the general robotic fabrication set up, followed by the detailed explanation of the demonstration sequence of both approaches in Section 7.2. Section 7.3 and Section 7.4 outline the specific instantiated data models for both the drilling and plasma cutting processes. The final Section 7.5 provides insight into robotic execution based on both data models. The results of the demonstration will be assessed through two distinct yet complementary approaches, which is presented in the next Chapter 9. Within this chapter, excerpts of the updates, queries and Python code are listed to demonstrate the specific application of the developments. A comprehensive summary of all code is published in Appendices E and F.

7.1 Test set up

Despite the varying data sources underlying the demonstrations, the robotic set up remained consistent (see Figure 7.1). In both scenarios, the KUKA robot KR30-3 F was used to perform the tasks, with different end-effectors attached for each specific operation. For drilling, a specialised drill was mounted on the robot (Bosch GSB 13 RE Professional), while the plasma cutting task used a custom torch connected to Powermax45 XP-Hypertherm system. Additionally, an end-effector was designed for the Keyence LJ-X8400 line scanner. Besides the design of the end-effector, a mobile tool change system was developed at the Chair of Individiualized Production to ensure a flexible robotic set up for different processes.

Before the final operation, the entire system, tool, robot and their relationship to the workpiece were calibrated. Different drill bits were used in the drilling process, so individual calibrations were carried out to ensure that the different lengths of the drill bits do not collide or penetrate too deeply into the material after drilling. For the process execution, the robot's precise movements and operations were simulated and pre-planned in Grasshopper, a visual programming language plug-in for Rhino 3D. Once the collisions have been checked and the path confirmed, the robot and the various tools were controlled using the KUKA|crc (Cloud Remote Control -

CRC) framework ([20, 217]). The various devices were IoT-enabled and connected via a Message Queuing Telemetry Transport (MQTT) broker. All devices communicated over a standard 5 GHz customer-grade Wireless Local Area Network (WLAN). A main control unit received all commands and distributed them to the devices (e.g., robot, plasma torch). This was a two-way communication system, as the devices could also send back state information. Additionally, it was also able to orchestrate the commands, e.g., send the switching plasma torch on command only when the robot has reached a certain position. Detailed information about the execution based on the data models and the data feedback is explained below.

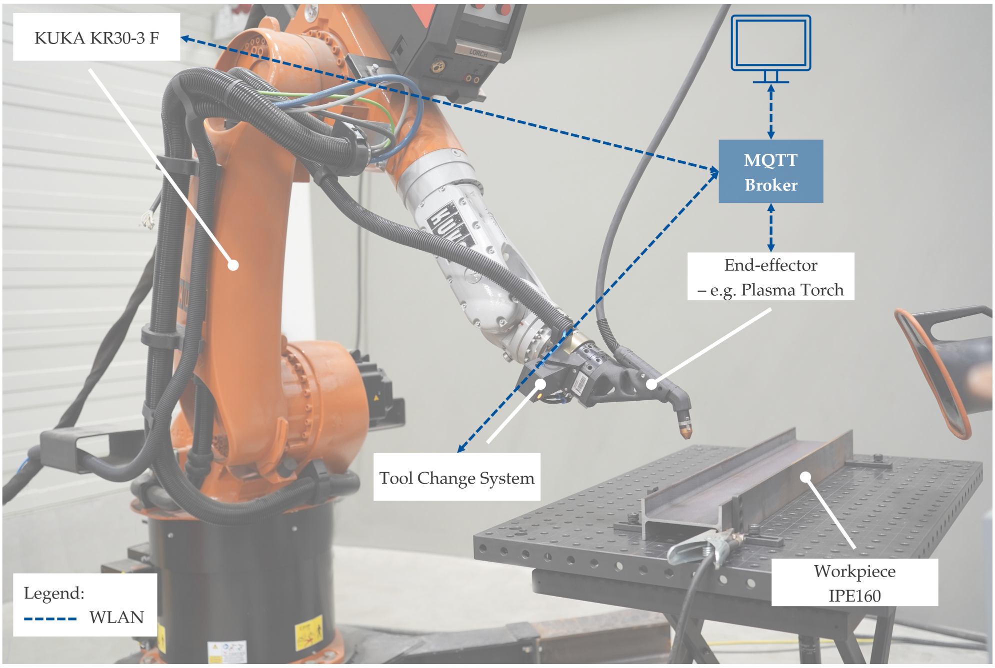


FIGURE 7.1: Overall robotic set up for the steel fabrication processes on the IPE 160

7.2 Demonstration sequence

The following Figure 7.2 illustrates the sequence for the demonstrations. The objective is to prove the feasibility of the data models in addressing the requirements of the four use cases (steel detailing, robotic fabrication and process feedback for optimisation, quality control and deconstruction).

Initially, the chosen sample beam, designated IPE 160, with four boreholes and two cutouts, was modelled with Tekla Structures. Two IFC versions (2x3 and 4) were exported to ascertain differences in steel detailing. However, it was observed that the current exports did not fulfil the new requirements analysed in this thesis. As previously stated, the most recent IFC 4 version was served as the reference. Consequently, this export was retrieved and modified in

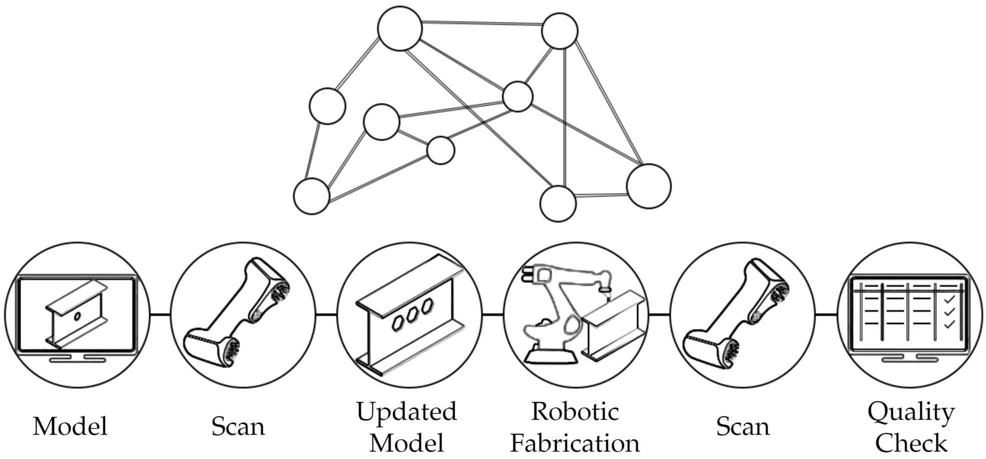


FIGURE 7.2: Demonstration sequence and the concept of the data models

accordance with the properties and entities defined in Chapters 5 and 6. A detailed explanation of the individual procedures and the examples given here for demonstration purposes are provided below. Subsequent to the completion of the first phase, the workpiece was scanned prior to machining. During this process, the profile geometries (e.g., width, height) were measured in real terms and compared with the nominal values and the defined tolerance values. The defined reference of the machining plays a role in this regard. For instance, if a hole is to be placed in the centre of the flange and the reference to the upper edge does not have to be maintained, the x/y coordinates of the hole can be adjusted accordingly. The data is stored in both data files. Once the final values have been determined, the robot's paths were planned and simulated to prevent a collision with the workpiece. Following the final check, the respective processing step was released and executed, after which the processed beam was measured again to ascertain whether the fabrication meets the defined tolerances. The data from the process itself, such as the time, which machine carried out the process or even at what speed, were stored back in the models and can then be used for quality control, documentation requirements or for later use of the data, such as deconstruction. In this context, it is imperative to differentiates between the data that remains internally in fabrication and the data that is fed back into the higher-level model as added value for several project participants.

The following sections demonstrate IFC-based, ontology-driven robotic fabrication.

7.3 IFC based fabrication

In relation to the various categories of requirements outlined in the conceptual model, the majority of the general data contained within the header block of the DSTV-NC file was already incorporated within the header of the IFC file. Consequently, the focus of the demonstration section was directed towards the integration of additional data as property sets, which were not previously included in the file for steel fabrication purposes.

The comparison of the two exports in terms of the two fabrication characteristics was the starting point for the IFC-based robotic fabrication. Various exports highlighted the prevailing lack of standardisation and consistency in data exchange within IFC files. Both the IFC 2x3 and IFC 4 exports defined the cutouts as *IfcOpeningElement*. The holes were also modelled as *IfcOpeningElement* in 2x3 and was is possible to assign an additional property set to describe the cutout and holes (e.g., height and width). Conversely, in the IFC 4 export, holes could not be modelled individually; instead, they were linked to the set of geometries of the mechanical fastener. As a result, the definitions within the Fabrication View, which prescribe the addition of properties to voiding features (e.g., holes), could not be directly associated with the hole of mechanical fastener itself. It emphasised the demand for a standardised description of steel detailing. According to the IFC Steel Fabrication View, any fabrication feature involving material removal should be classified as *IfcVoidingFeature*. To maintain consistency with this definition, the entities were accordingly modified. Details for each feature is explained below in the respective sub section.

For the given demonstrator, generic properties (*Pset_AdditionalProfileProperties*, *Pset_FabricationProcess*) independent on the fabrication as well as feature specific properties and entities (here for the cutout, the holes and also the processes) were added. As demonstrated in this thesis, the limited utilisation of robots in steel pre-fabrication can be attributed to the varying tolerances within which the base material is produced. To address these deviations from target values and, when necessary, adjust the robot's path, a property set was integrated into the profile definition. This property set stores both the measured values and the defined tolerances. The following explains the procedure for manually adding the new property sets and apply existing entities for steel fabrication. The process of adding standardised property sets to IFC elements was implemented using IfcOpenShell and Python to follow a structured and systematic approach to ensure consistency and interoperability within the IFC model (see Listing 7.1).

First, the IFC file was loaded using 'IfcOpenShell .open()', allowing access to its elements and data structure. The target element, such as the *IfcBeam*, was then identified using the 'model.by_type()' function. Next, a new *IfcPropertySet* was created with a unique GUID and a standardised name, such as *Pset_AdditionalProfileProperties*. Individual properties were then defined using *IfcPropertySingleValue* and assigned appropriate data types, such as *IfcReal* for numerical values or *IfcLabel* for textual references. These properties were added to the property set to align with IFC standard. To establish a formal link between the property set and the IFC element, an *IfcRelDefinesByProperties* relationship was created. This ensured that the additional data is correctly associated with the respective element, making it accessible within IFC-compatible software. Finally, the updated IFC file was saved using 'model.write()', preserving all modifications for further use.

For the current demonstrations in this thesis, all property sets and process values were updated within a local IFC file. Ideally, future workflows will incorporate standardised property sets and steel fabrication-specific entities directly within the IFC software, allowing detailers to use them accordingly. Planned process specifications or process feedback data can then be updated into the shared model. For this approach to be effective, the updated IFC model must reside

LISTING 7.1: IfcOpenShell property set creation - additional profile properties

```
model = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.ifc")

beam = model.by_type("IfcBeam")[0]

measured_height = model.create_entity("IfcPropertySingleValue", "MeasuredHeight", None, model.create_entity("IfcPositiveLengthMeasure", 162.1))
measured_width = model.create_entity("IfcPropertySingleValue", "MeasuredWidth", None, model.create_entity("IfcPositiveLengthMeasure", 81.4))
measured_length = model.create_entity("IfcPropertySingleValue", "MeasuredLength", None, model.create_entity("IfcPositiveLengthMeasure", 1002.0))
height_tolerance_min = model.create_entity("IfcPropertySingleValue", "HeightToleranceMin", None, model.create_entity("IfcLengthMeasure", -2.0))
height_tolerance_max = model.create_entity("IfcPropertySingleValue", "HeightToleranceMax", None, model.create_entity("IfcLengthMeasure", 3.0))
width_tolerance_min = model.create_entity("IfcPropertySingleValue", "WidthToleranceMin", None, model.create_entity("IfcLengthMeasure", -1.0))
width_tolerance_max = model.create_entity("IfcPropertySingleValue", "WidthToleranceMax", None, model.create_entity("IfcLengthMeasure", 4.0))
din_reference = model.create_entity("IfcPropertySingleValue", "DINReference", None, model.create_entity("IfcLabel", "DIN EN 10034"))

rel_defines_beam = model.create_entity("IfcRelDefinesByProperties", ifcopenshell.guid.new())
rel_defines_beam.Name = "Defines additional profile properties"
rel_defines_beam.RelatedObjects = [beam]
rel_defines_beam.RelatingPropertyDefinition = model.create_entity("IfcPropertySet", ifcopenshell.guid.new())
rel_defines_beam.RelatingPropertyDefinition.Name = "AdditionalProfileProperties"
rel_defines_beam.RelatingPropertyDefinition.HasProperties = [
    measured_height,
    measured_width,
    measured_length,
    height_tolerance_min,
    height_tolerance_max,
    width_tolerance_min,
    width_tolerance_max,
    din_reference
]

model.write(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.ifc")
```

in a Common Data Environment (CDE), ensuring accessibility for all relevant stakeholders. While this research does not involve real-time file sharing, platforms like Trimble Connect offer application programming interfaces (APIs) that, when combined with IfcOpenShell, enable automated data uploads for broader accessibility.

7.3.1 Initial scan and measurement

As described in the demonstrator workflow the beam is scanned before processing. This was undertaken by the Keyence line scanner attached to the robot.

With the property set *Pset_AdditionalProfileProperties* the measured values of the beam could be stored before machining and the planned value could be checked according to the tolerance which is defined for it. The newly added property set could be visualised with open viewers, such as the *KITModelViewer* [218]. Figure 7.3 shows a screenshot of the new property set *Pset_-AdditionalProfileProperties*, corresponding properties and the added values. The numbers show that the IPE 160 beam with a planned height of 160 mm and a planned width of 80 mm was produced in line with the defined tolerances.

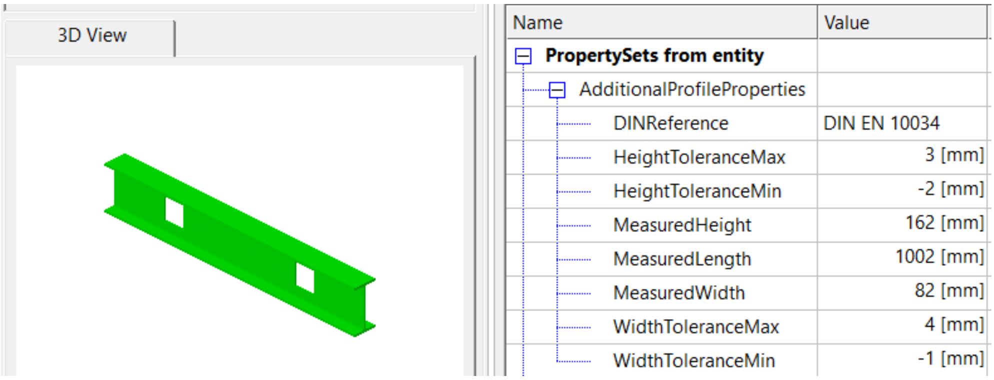


FIGURE 7.3: Screenshot of added Pset_AdditionalProfileProperties

The following subsections provide a detailed explanation of the modifications according to the Steel Fabrication View and the specific properties for the process.

7.3.2 IFC based drilling

In order to run the IFC based robotic drilling process, the following property sets and existing IFC schema entities were manually added to the IFC file, following the same procedure as described above. These components represented exemplary elements of the defined Fabrication View, which addressed the four identified use cases:

- Add *IfcVoidingFeature* where additional property sets could be linked to
- *Pset_ThroughHole*, as defined in Table 5.8 with the planned diameter, as well as the measured diameter and the defined tolerances from the DIN norm
- Use of entity *IfcTask* to describe the drilling process and the related status
- Use of entity *IfcTaskTime* to store the scheduled start and finish time of the process but also document the actual times
- *Pset_DrillingProcess*, as described in Section 5.3.7, although here only the robot speed and planed spindle speed could be stored, as the tool used for the application does not have a sensor to track the spindle speed, and no coolant flow etc is used.

Based on the IFC 4 export, the boreholes were part of the set of mechanical fastener geometries, where two pairs each had a mapped representation. The initial entity could not be changed nor could an additional property set be linked to the geometry of the hole (*IfcCircleProfileDef*). To be in line with the definitions of the Fabrication View, voiding features with the predefined type 'HOLE' were added and linked to the mechanical fastener. Afterwards, the new property*Pset_ThroughHole* set could be attached to the entity, providing details for fabrication but also data for quality control (tolerance values). Listing 7.2 provides the key commands within IfcOpenShell to add the property set. The measured value is empty before the process has been executed.

LISTING 7.2: IfcOpenShell property set creation

```
Pset = model.create_entity("IfcPropertySet", ifcopenshell.guid.new())
Pset.Name = "Pset_ThroughHole"
Pset.HasProperties = [
    model.create_entity("IfcPropertySingleValue", "Diameter", None, model.
     create_entity("IfcPositiveLengthMeasure", 10.0)),
    model.create_entity("IfcPropertySingleValue", "MeasuredDiameter", None, model.
     create_entity("IfcPositiveLengthMeasure", '')),
    model.create_entity("IfcPropertySingleValue", "DiameterToleranceMin", None,
     model.create_entity("IfcLengthMeasure", -0.5)),
    model.create_entity("IfcPropertySingleValue", "DiameterToleranceMax", None,
     model.create_entity("IfcLengthMeasure", +0.5)),
    model.create_entity("IfcPropertySingleValue", "NormReference", None, model.
     create_entity("IfcLabel", "DIN EN 1090-2"))
]

rel_defines = model.create_entity("IfcRelDefinesByProperties", ifcopenshell.guid.
     new())
rel_defines.RelatedObjects = [voiding_feature]
rel_defines.RelatingPropertyDefinition = Pset
```

Chapter 4 highlighted the demand of describing process relevant data within the IFC to be able to execute the process on the basis and to supersede the DSTV-NC file. Within Chapter 5, it was introduced that the existing entity *IfcTask* can be used to specify the fabrication process itself. To achieve this, the *IfcTask* was linked with *IfcRelAssignsToProcess* to the voiding features.

With *IfcTaskTime* the scheduled and actual execution times could be stored. Additionally, the *Pset_ManufacturerTypeInformation* was added and utilised to provide information about the resource responsible for executing the process. In order to define the process parameters, but also to store data for documentation and feedback, the *Pset_DrillingProcess* has been integrated. Both property sets were linked to the entity *IfcTask*. Details on the execution the the parameters for the drilling process are listed below in Table 7.1.

7.3.3 IFC based plasma cutting

A secondary fabrication feature and process was selected for the purpose of evaluating the IFC-based process. With regard to the cutouts in the web, the decision was taken to utilise a plasma-cutting process. Similar to the holes, the basic IFC 4 export of the sample beam was taken and added with plasma cutting specific requirements. The cutouts were modelled as *IfcOpeneingElement* and were changed to *IfcVoidingFeature* with the predefined type of 'CUTOUT'.

The *Pset_Cutout* was attached to it, specifying the parameters of the cutout and implementing the tolerance values for digital availability and quality control. Within the *Pset_Fabrication-Process* it was documented that the required process for this feature was plasma cutting. The measured values were not yet added here, as this represents the planned status. A timestamp has been added to track when the values have been updated. The following Listing 7.3 represents the added property sets as STEP format.

LISTING 7.3: Added Pset in STEP format

```
#290=IFCVOIDINGFEATURE('3W202mdAL4dwQkU01hLgFg',$,'Cutout 1','BL10',$,#178,#189,$,.
    CUTOUT.);
#291=IFCRELVOIDSELEMENT($,$,$,$,#155,#290);
#292=IFCPROPERTYSET($,$,'Pset_Cutout',$
    ,(#293,#294,#295,#296,#297,#298,#299,#300,#301,#302));
#293=IFCPROPERTYSINGLEVALUE('Width',$,IFCPOSITIVELENGTHMEASURE(80.),$);
#294=IFCPROPERTYSINGLEVALUE('MeasuredWidth',$,IFCPOSITIVELENGTHMEASURE(0.),$);
#295=IFCPROPERTYSINGLEVALUE('WidthToleranceMin',$,IFCLENGTHMEASURE(- 0.),$);
#296=IFCPROPERTYSINGLEVALUE('WidthToleranceMax',$,IFCLENGTHMEASURE(3.),$);
#297=IFCPROPERTYSINGLEVALUE('Height',$,IFCPOSITIVELENGTHMEASURE(80.),$);
#298=IFCPROPERTYSINGLEVALUE('MeasuredHeight',$,IFCPOSITIVELENGTHMEASURE(0.),$);
#299=IFCPROPERTYSINGLEVALUE('HeightToleranceMin',$,IFCLENGTHMEASURE(- 0.),$);
#300=IFCPROPERTYSINGLEVALUE('HeightToleranceMax',$,IFCLENGTHMEASURE(3.),$);
#301=IFCPROPERTYSINGLEVALUE('DINReference',$,IFCLABEL('DIN EN 1090-2'),$);
#302=IFCPROPERTYSINGLEVALUE('Timestamp',$,IFCTIMESTAMP(),$);
#303=IFCPROPERTYSET($,$,'Pset_FabricationProcess',$,(#304,#305,#306,#307));
#304=IFCPROPERTYSINGLEVALUE('ProcessRequired',$,IFCLABEL('plasma cutting'),$);
#305=IFCPROPERTYSINGLEVALUE('ProcessForbidden',$,IFCLABEL(''),$);
#306=IFCPROPERTYSINGLEVALUE('Comment',$,IFCLABEL(''),$);
#307=IFCPROPERTYSINGLEVALUE('Timestamp',$,IFCTIMESTAMP(),$);
#308=IFCRELDEFINESBYPROPERTIES($,$,$,$,(#290),#292);
```

As above, the entities *IfcTask* and *IfcTaskTime* have been added also for the plasma cutting process. Before the process has started, the predefined type of the attribute 'Status' is 'NOT-STARTED' and only the 'ScheduleStart' and 'ScheduleFinish' had values.
Last, the *Pset_CuttingProcess* was linked to the *IfcTask* to store the process specifications for planning and documentation. In this instance, the demonstrator and the designated configuration were such that only the robot's speed can be measured. This was due to the fact that not more sensors had been implemented.

This section provided a detailed account of the implementation of the developed elements of the Fabrication View for the designated demonstrator The final modified IFC file can be found in the Appendix D. Details pertaining to the execution of the fabrication process will be highlighted in Section 7.5.

7.4 Ontology driven fabrication

Referring back to the objective of replacing the DSTV-NC format with a multi-modal approach, the following section explains how ontology-driven fabrication was applied and evaluated, and which data were instanced with existing and newly defined ontologies.

The base for the ontology driven fabrication was also the IFC 4 export from Tekla Structures. The file was taken with the cutout and the holes changed to *ifcVoidingFeature*. This is also present in ifcOWL. Using the IFC to LBD Converter [113], the .ifc file was converted into a .ttl file, allowing the information to be stored as triples. The file was then uploaded to GraphDB, a semantic graph database [109], that enabled querying of the stored information using SPARQL queries. Unlike the locally stored IFC file, the GraphDB database could be shared, allowing participants in the demonstration process to retrieve and upload data. As described above the first step of the demonstration was the scan of the beam to check the real dimensions. This scan was performed with the Keyence line scanner, attached to the robot who executed the movements. Subsequent, the collected points were stored and visualised within Cloud Compare. Here, the key features to be stored and compared to the planned values were measured.

To upload this data to the triplestore and establish a link to the beam profile definition, an update query was created using the DSTV ontology. Currently, the profile definition of the beam, more specifically, the *IfcIShapeProfileDef*, stored its values as direct object definitions. This structure allowed only a single value to be assigned, for example, to the overall width or depth of the beam. However, this approach introduced limitations, which are addressed by the concept of measurable properties. These properties can exist in four distinct states: planned, measured, adjusted, and defined tolerance. To overcome the constraints of a single-value representation, the query extended the data model by adding multiple states, each associated with a value and a timestamp. This ensured a more comprehensive and traceable representation of the beam's dimensions over time.

Listing 7.4 presents the query that updated the measured width of the beam while preserving the original value as the planned value, along with its corresponding timestamp. The width of the beam was measured at three points: the front, midpoint and end. The median is documented below. This update query can be implemented in a Python script to automatically update the data, if an automated analysis of the scan has been defined.

LISTING 7.4: SPARQL update for inserting *IfcIShapeProfileDef* values

```
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX schema: <http://schema.org>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX qudt-unit-1-1: http://qudt.org/1.1/vocab/unit#

INSERT {
    inst:IfcIShapeProfileDef_137 a ifc:IfcIShapeProfileDef.
    inst:IfcIShapeProfileDef_137 ifc:hasProperties_IfcProfileDef inst:Width.
    inst:Width a ifc:IfcProfileProperties.
    inst:Width dstv:hasPlannedValue inst:Width_IPE160_P.
    inst:Width dstv:hasMeasuredValue inst:Width_IPE160_M.

    inst:Width_IPE160_P a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Width_IPE160_P qudt-1-1:numericValue 82.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Width_IPE160_P prov:generatedAtTime ?timestamp.

    inst:Width_IPE160_M a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Width_IPE160_M qudt-1-1:numericValue 81.4"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Width_IPE160_M prov:generatedAtTime ?timestamp.

}
WHERE {
    BIND(NOW() AS ?timestamp)
}
```

In addition to the measured values, and in order to have the tolerances of the DIN standards digitally available, the set values for the profile dimensions were also uploaded and linked to the respective class. The Listing 7.5 shows the update query for the upper and lower limits as well as the DIN reference for the beam width. The same data has been updated for the depth.

LISTING 7.5: SPARQL update for beam tolerances

```
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX schema: <http://schema.org>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX qudt-unit-1-1: http://qudt.org/1.1/vocab/unit#

INSERT {
    inst:Width a ifc:IfcProfileProperties.

    inst:Width dstv:hasDefinedTolerancePropery inst:DefinedToleranceWidth.
    inst:DefinedTolerance dstv:hasNormReference inst:NormWidth.
    inst:DefinedToleranceWidth dstv:hasToleranceLimit inst:ToleranceLimitWidth.
    inst:ToleranceLimitWidth dstv:hasUpperToleranceLimit inst:Tolerance_IPE160_Width_U.
    inst:ToleranceLimitWidth dstv:hasLowerToleranceLimit inst:Tolerance_IPE160_Width_L.

    inst:Tolerance_IPE160_Width_U a dstv:PropertyState, qudt-1-1:QuantityValue.
    inst:Tolerance_IPE160_Width_U qudt-1-1:numericValue 4.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Tolerance_IPE160_Width_U prov:generatedAtTime ?timestamp.

    inst:Tolerance_IPE160_Width_L a dstv:PropertyState, qudt-1-1:QuantityValue.
    inst:Tolerance_IPE160_Width_L qudt-1-1:numericValue -1.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Tolerance_IPE160_Width_L prov:generatedAtTime ?timestamp.

    inst:NormWidth schema:Value "DIN EN 10034".
}
WHERE {
    BIND(NOW() AS ?timestamp)
}
```

7.4.1 Ontology driven drilling

Subsequent to the definition of the workpiece general data using the ontology structures and its storage in triplets, the process-related data for the drilling and plasma cutting process have been modelled. Figure 7.4 presents a sketch of the relevant classes and relations for executing the drilling process. As previously demonstrated, the existing ifcOWL definition was employed to represent the element assembly, as the hole was connected to a mechanical fastener. Moreover, the IoC ontology was utilised to describe the processes, and the DSTV ontology was employed for the specific steel structure features. The green bordered boxes show the required classes for potential robotic deconstruction. The centre of gravity is a critical factor in determining the optimal location for the robot or machine to hold the element in a balanced position. It is already established in IFC 4, but has not been mapped with IFC 4's ifcOWL.

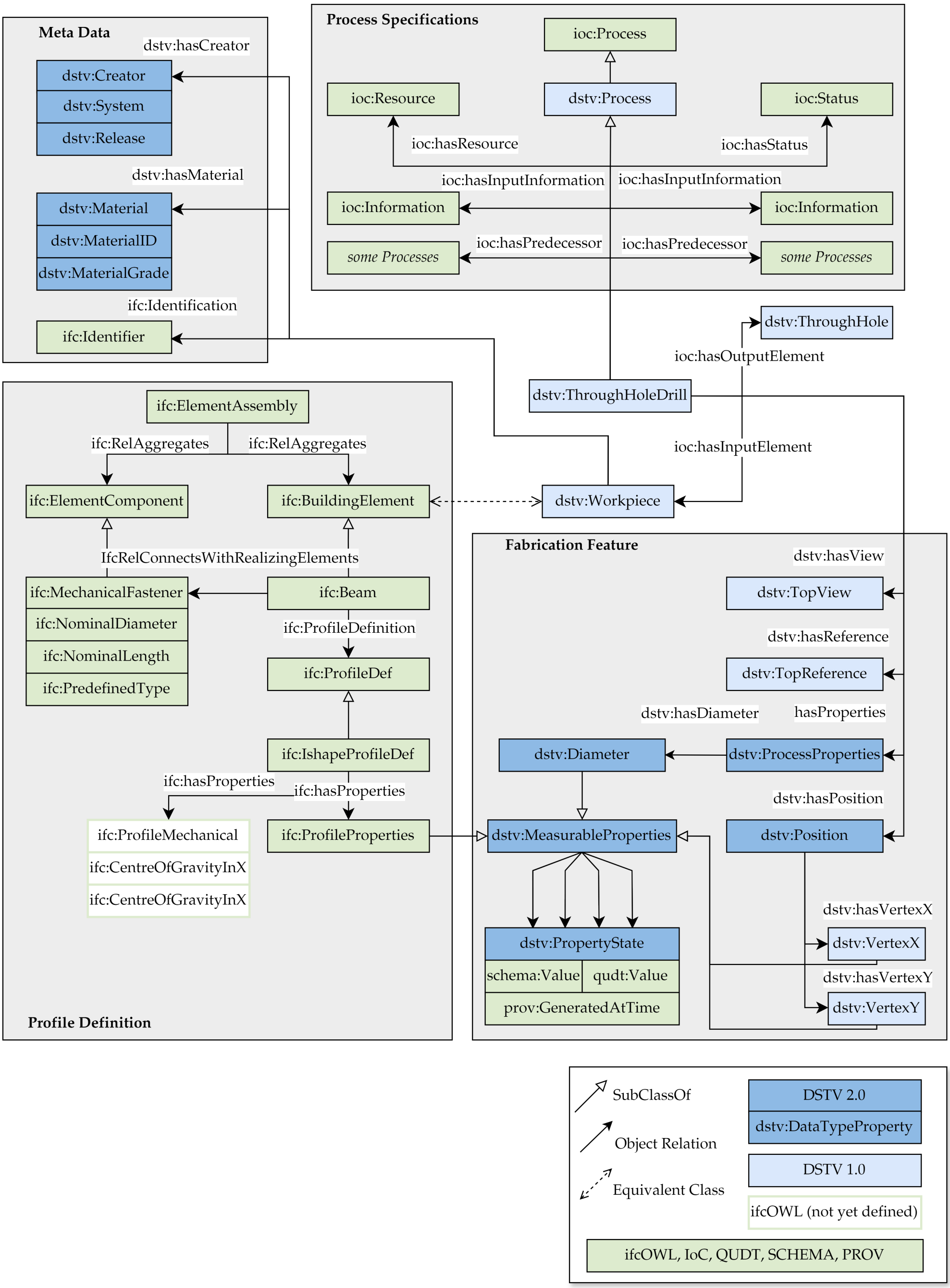


FIGURE 7.4: Representation of relevant classes for drilling process

The core triplets defined for the drilling process pertained to both the process itself and the associated data for the respective feature. The class *dstv:ThroughHoleDrill* constituted a sub-process of *dstv:Process*, which in turn was a sub-process of *ioc:Process*. An input element, the *dstv:Workpiece*, has been specified for the process, as well as an output element, which was the *dstv:ThroughHole*, corresponding to the *ifc:Hole* in the IFC schema. The drilling process has been further subdivided into discrete sub-processes: the robotic system approached the workpiece, initiated the drilling operation, and subsequently retracted. The toolpaths generated in Grasshopper were stored using the *ioc:hasInputInformation* predicate. Further details, including the various process states and their corresponding timestamps, are provided in Section 7.5. The average velocity of the robotic system was recorded as process feedback using the *ioc:hasOutInformation* relation and *ioc:Information* class. The data relating to the fabrication feature primarily concerned the position and diameter of the hole during the drilling operation. As shown in Figure 7.4, the position of the drill hole was classified as a subclass of the measurable property class. Consequently, it could exist in four different states: planned, measured, adjusted, and defined tolerance. The first scan of the original beam, performed at the beginning of the process, was analysed and the actual values were recorded. This revealed deviations in width compared to the planned dimensions (see Listing 7.4), but within the tolerance (see Listing 7.5). In steel fabrication, features always reference a fixed edge. In this case, the drill holes were positioned based on their distance from the upper edge, meaning no adjusted values for vertex X and vertex Y needed to be stored. The drill diameter was inserted with the planned and tolerance values (with upper and lower limits) before fabrication and linked to the void feature. Subsequent to the process, the measured values were incorporated.

7.4.2 Ontology driven plasma cutting

The procedure for the ontology driven plasma cutting was aligned with the drilling procedure and the same concepts were implemented. Listing 7.6 shows the triplets that have been added in representing the application of the IoC and DSTV ontologies to the given use case of plasma cutting a cutout. First, the instance of the workpiece *inst:IfcBeam_155* was created as the input element, and the instance *inst:IfcVoidingFeature_290*, which was the cutout, was the output element. To store real data, the the plasma cutting process itself (*inst:ProcessInstance_-290*) must be linked with the input/output elements. The fabrication feature cutout, originally a *dstv:InternalContouring*, was defined as the instance *inst:_010_Plasma_290*. This feature was detailed by properties including, but not limited to, width and height. At present, the listing only added the planned values of the cutout with the value and the unit. However, it should be noted that a general measured value has already been defined (see triple: *inst:WidthCutout_290 dstv:hasMeasuredValue inst:WidthCutout_290_M*), but not yet added with real numbers. As these process properties were a subclass of the measurable properties, the instance *inst:WidthCutout_290* and height also have defined tolerances that have been added.

For the robotic plasma cutting process, the robot commands and robot speed were also stored as input and output information, as well as the status of the processes (approach, plasma cut, retract) with the corresponding time stamp. Details are given in the next section.

LISTING 7.6: SPARQL update for cutout process data

```
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX ioc: <http://w3id.org/ioc#>
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <http://baufest.org/inst#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX qudt-unit-1-1: <http://qudt.org/1.1/vocab/unit#>


INSERT {
    inst:InputElementInstance_292 a ioc:Element, inst:IfcBeam_155.
    inst:OutputElementInstance_292 a ioc:Element, inst:IfcVoidingFeature_292.

    inst:_011_Plasma_292 ioc:hasInputElement inst:_InputElementInstance_292;
  inst:_InputElementInstance_292 ioc:CurrentProcessDataState;
  ioc:hasInputElementValue inst:InputElementInstance_292.

    inst:ProcessInstance_292 ioc:hasOutputElement inst:_OutputElementInstance_292;
  inst:_OutputElementInstance_292 ioc:CurrentProcessDataState;
  ioc:hasOutputElementValue_292 inst:OutputElementInstance_292.

  inst:_011_Plasma_292 a dstv:internalContouring.

    inst:_011_Plasma_292 dstv:hasProperties inst:ProcessProperties_292.
    inst:ProcessProperties_292 dstv:hasWidth inst:WidthCutout_292.
    inst:ProcessProperties_292 dstv:hasHeight inst:HeightCutout_292.

    inst:WidthCutout_292 dstv:hasPlannedValue inst:WidthCutout_292_P.
    inst:WidthCutout_292 dstv:hasMeasuredValue inst:WidthCutout_292_M.

    inst:WidthCutout_292_P a dstv:PropertyState, qudt-1-1:QuantityValue.
  inst:WidthCutout_292_P qudt-1-1:numericValue "80.0"^^xsd:double;
  qudt-1-1:unit qudt-unit-1-1:MilliM.
  inst:WidthCutout_292_P prov:generatedAtTime ?timestamp.

    inst:HeightCutout_292 dstv:hasPlannedValue inst:HeightCutout_292_P.
    inst:HeightCutout_292 dstv:hasMeasuredValue inst:HeightCutout_292_M.

    inst:HeightCutout_292_P a dstv:PropertyState, qudt-1-1:QuantityValue.
  inst:HeightCutout_292_P qudt-1-1:numericValue "80.0"^^xsd:double;
  qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:HeightCutout_292_P prov:generatedAtTime ?timestamp.
}

Where{
Bind(NOW() As ?timestamp)
    }
```

7.5 Robotic execution

Subsequent to the modelling of both data formats, an empirical investigation was conducted to assess the feasibility of robotic fabrication with these formats. As outlined at the beginning, two of the four holes and one cutout were fabricated based on the data provided in the IFC file, and the feedback related to these holes is written back into the same IFC file. The other two holes and the second cutout were represented using Linked Data, with the corresponding data stored in the triplestore according to defined ontologies.

To initiate the planning of the robot's movements, key data for the features were retrieved from the triplestore and from the IFC file. With this information available, the planning process in Grasshopper was finalised. The robot path generated for each respective feature was first simulated and evaluated in conjunction with the visualisation of the 3D model to identify potential collisions (see Figure 7.5).

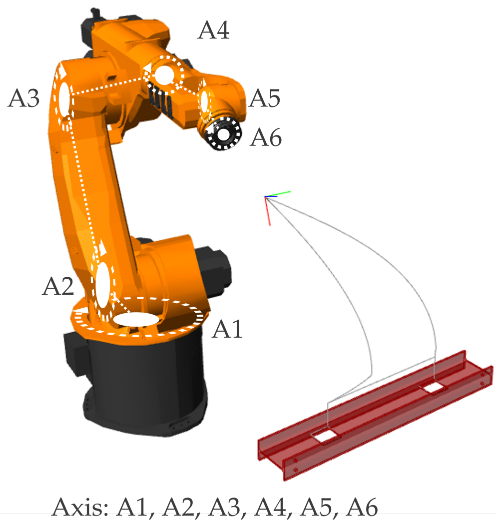


FIGURE 7.5: Simulation of robot path in relation to workpiece

There is an official plug-in for Grasshopper that allows an IFC file to be read directly into the robot design environment. This plug-in, provided by Geometry Gym, facilitated the extraction of structured building information by mapping IFC entities into native Grasshopper geometry and data types, which could then be analysed, filtered and translated into actionable parameters for robot path planning, such as tool paths, spatial constraints and feature recognition (e.g., holes, edges, surfaces).

In general, the plasma cutting and the drilling process itself was divided into three sub-processes, the approach of the tool to the workpiece, the fabrication and the retraction at the end. Each of these processes had its own command stream. For the ontology driven concept, a Python script was used to add the processes to the triplestore as *ioc:Process* with the robot commands as *ioc:InputInformation*. Additional information such as the robot speed was also added as *ioc:InputInformation*. Each process had a *ioc:Status*. Initially the possible states were set to false. A Python script was written to run the process. It started by checking if any process had a *ioc:isReady* state. If none is set to this state, no process was started. Once the process was set to *ioc:isReady*, a confirmation was requested to start the fabrication. At the same time, the status for this process was set to *ioc:isStarted* and this status change was saved with a timestamp to the triplestore with an update query. The stored robot commands for this process were then published to the robot via MQTT. While the process was running, the status was updated to *ioc:isRunning*. Additionally, the robot status was monitored, both the active/passive state and the list of open commands. If the robot was not moving and the last command had been executed, it was known that the process was finished and the status was set to *ioc:isFinished* and saved again with the timestamp in the triplestore. This state could be queried later to know the duration of the process. As the last step of this defined loop was finished, the next process defined as the successor process (here cutting after approach) was set to *ioc:isReady* and the sequence started from the beginning.

Within the IFC model, only the overarching process command stream, for example, the plasma cutting process, could be stored, and this was added as a property to the corresponding voiding feature. The general code for executing the process, including the transmission of pre-planned robot commands to the robot, remained consistent across all operations. However, intermediate status changes were not continuously updated within the IFC file; only the actual start and finish times were recorded.

After describing the general robotic execution, the following outlines the specifications for the drilling and plasma cutting processes.

7.5.1 Drilling

The configuration of the drill, drill bits and robotic arm did not permit direct drilling of the final diameter of 10 mm for the hole. Consequently, the holes were drilled at four distinct diameters, each with a different rotational speed and feed rate. Table 7.1 provides an overview of the final process parameters. The rotation speed per diameter was associated with the established values outlined in norms (e.g., DIN 6584). However, it was not possible to measure the actual values accurately in this instance, as the drill only operated in steps and no sensor had been installed. Consequently, the values were estimated based on the steps values provided by the supplier.

TABLE 7.1: Fabrication parameters for robotic drilling

Diameter [mm]	Rotation Speed [r/min]	Feed [mm/s]
4	ca. 1800	0,5
6	ca. 1200	0,3
8	ca. 1100	0,1
10	ca. 800	0,05

Each drilling operation (16 in total, comprising 4 holes at 4 different diameters) was further subdivided into sub-processes: approach to the workpiece, drilling, and retraction. These sub processes were not explicitly modelled within either data model; only the four final hole geometries were represented. The scheduled start time was recorded using the *ifcTaskTime* entity (see Listing 7.7), while the planned value in LBD was stored within the *ioc:Status* (*ioc:isPlanned*) linked to the *ioc:process*. The various sub-processes were modelled as child processes. For the purpose of demonstration efficiency, all four holes of the same diameter were fabricated consecutively prior to initiating a tool change. The overall procedure of the robot control was explained above.

LISTING 7.7: IfcOpenShell for adding scheduled start time

```
import ifcopenshell
ifc = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\2025
     _Sample_Beam_updated.ifc")

task_time = next((entity for entity in ifc.by_type("IfcTaskTime")), None)

if task_time is None:
    print("IfcTaskTime not found.")
else:
    task_time.ScheduleStart = "2025-04-01T17:00:00"

    ifc.write(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\2025
     _Sample_Beam_updated.ifc")
```

Figure 7.6 shows the robotic execution of the drilling (left side) and the plasma cutting process (right side).

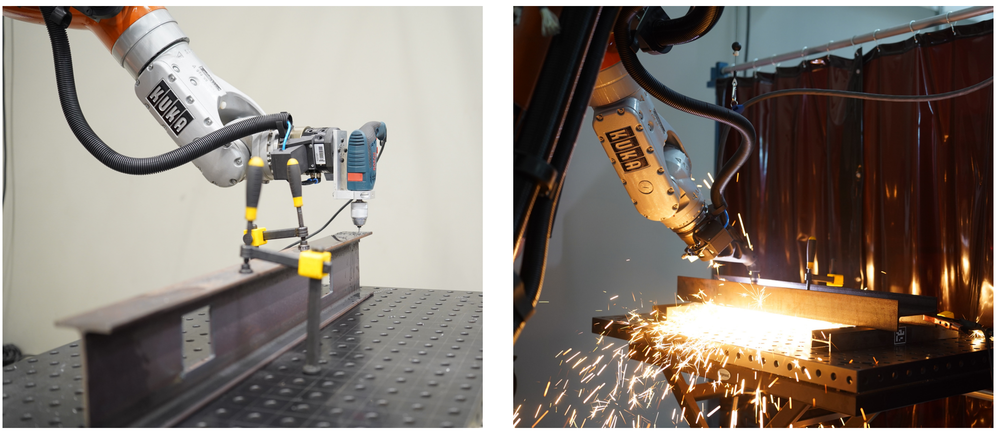


FIGURE 7.6: Demonstration of robotic drilling (left) and robotic plasma cutting (right)

7.5.2 Plasma cutting

In order to optimise the alignment of the material thickness with the plasma cutting tool, preliminary tests were conducted using various robot speeds. These speeds ranged between 5 and 10 [mm/s]. The final speeds selected for the demonstration are presented in Table 7.2. This ensured that the material could be fully cut through without causing excessive material accumulation or fusion at the edges.

TABLE 7.2: Fabrication parameters for robotic plasma cutting

Cutout ID	Feed [mm/s]
290	5
292	6

In total, two cut-outs were made from the web of the IP 160 beam, each with a height and width of 80 mm. The robotic fabrication procedure, as well as the two underlying data models and their respective methods for data storage, were identical to those used in the drilling process. As a result, it was likewise possible to track timings and status changes, and to record relevant process parameters.

In conclusion, both data formats successfully served as input for the robotic fabrication of holes and cutouts. Figure 7.7 shows the final fabricated beam.

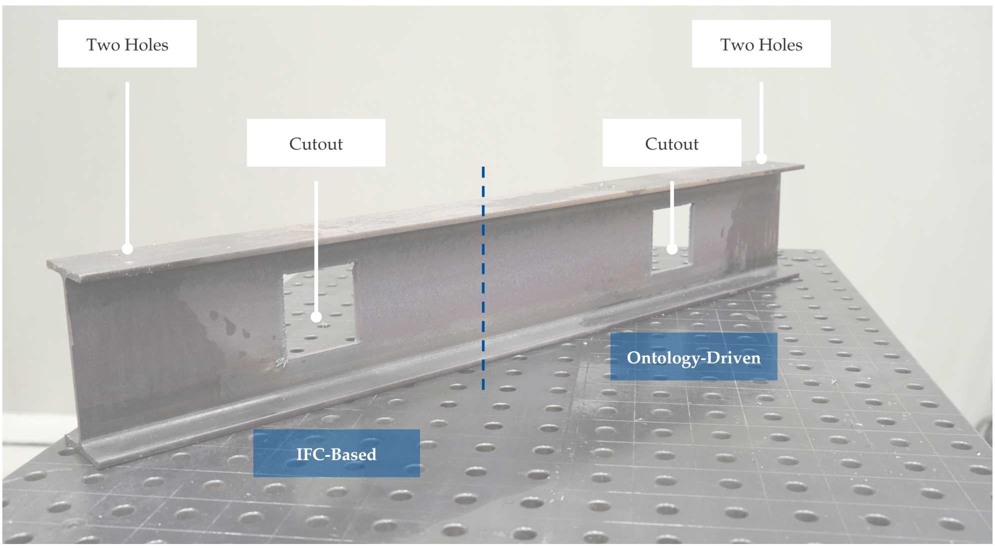


FIGURE 7.7: Fabricated beam

7.5.3 Measurement and process feedback

As outlined in this thesis, the actual values of the fabrication features and the process parameters are important for quality control and documentation. Therefore, another scan was performed (see the reference to the workflow overview in Figure 7.2). Figure 7.8 displays the overlay of the 3D model and the line scan, which was done in Cloud Compare.

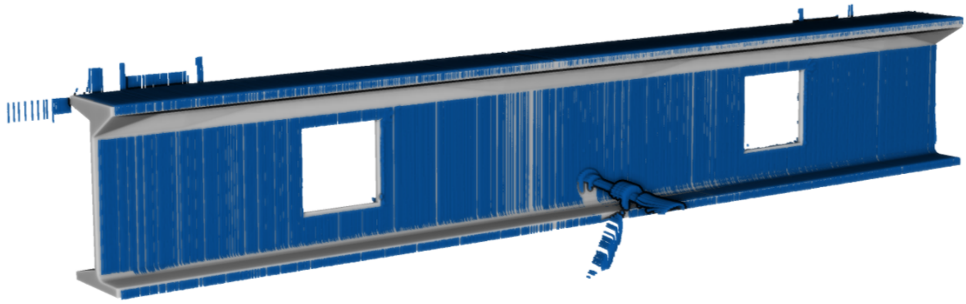


FIGURE 7.8: Overlay of 3D scan and the 3D model after the fabrication

The focus of the second scan laid on the cutouts' height, width and the holes' diameter. The used software allowed for measuring of the fabricated features (see Figure 7.9).

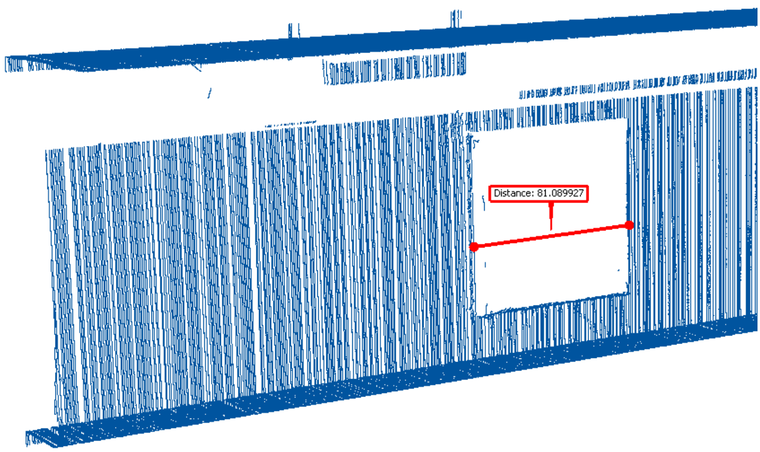


FIGURE 7.9: Analysis of beam scan for real fabrication feature dimensions

The measured values were uploaded to the respective property set in the IFC file (see exemplary IfcOpenShell Python code for measured diameter of one hole in Listing 7.8) but also as triplets to the triplestore and linked to the existing voiding feature (see exemplary SPARQL update for cutout dimensions in Listing 7.9). This data could be used for post processing and checking the fabrication against the defined tolerances.

LISTING 7.8: IfcOpenShell for adding measured values of drill diameter

```python
import ifcopenshell
import time
ifc_file = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam.ifc")

def update_measured_diameter(ifc_file, new_diameter):
    current_timestamp = int(time.time())

    for pset in ifc_file.by_type("IfcPropertySet"):
        if pset.Name == "Pset_ThroughHole":
            for prop in pset.HasProperties:
                if prop.Name == "MeasuredDiameter":
                    prop.NominalValue = ifc_file.create_entity("IfcPositiveLengthMeasure", new_diameter)
                elif prop.Name == "Timestamp":
                    prop.NominalValue = ifc_file.create_entity("IfcTimeStamp", current_timestamp)

update_measured_diameter(ifc_file, new_diameter=10.0)

updated_file_path = r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\2025_Sample_Beam_updated_voidingfeature.ifc"
ifc_file.write(updated_file_path)
```

LISTING 7.9: SPARQL update for measured cutout dimensions

```sparql
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX inst: <http://baufest.org/inst#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX qudt-unit-1-1: <http://qudt.org/1.1/vocab/unit#>

    inst:WidthCutout_292_M a dstv:PropertyState, qudt-1-1:QuantityValue.
    inst:WidthCutout_292_M qudt-1-1:numericValue "81.1"^^xsd:double; qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:WidthCutout_292_M prov:generatedAtTime ?timestamp.

    inst:HeightCutout_292_M a dstv:PropertyState, qudt-1-1:QuantityValue.
    inst:HeightCutout_292_M qudt-1-1:numericValue "80.3"^^xsd:double; qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:HeightCutout_292_M prov:generatedAtTime ?timestamp.
}

Where{
Bind(NOW() As ?timestamp)
```

In addition to the dimensions as process feedback some process parameters were also saved for documentation. The robot speed was constantly monitored and the average value could be saved within the triplestore as *ioc:hasOutputInformation* and in the property set in the IFC file.

In summary, the robotic demonstration was successfully completed, meeting all four predefined requirements. The steel detailing was accurately implemented, both the drilling and plasma cutting processes were executed as planned, relevant properties were integrated into the models, and process feedback was systematically captured and documented. These outcomes were achieved using both of the developed data models. Overall, the demonstration established a robust foundation for the subsequent evaluation and discussion of the results.

Chapter 8

Results

This chapter presents the results of the demonstrations carried out for this thesis. As stated in the introduction, the objective is to develop a data model for the automated fabrication of steel elements, while also providing a foundation for future deconstruction applications. The current widely used standard for prefabrication is the DSTV-NC format, which lacks the capability to store additional information required for robotic fabrication and process feedback. To address this limitation, two different technical approaches were implemented, each targeting the defined requirements for a holistic and interoperable data flow.

Section 8.1 provides a brief recap of the current DSTV-NC structure and highlights the achievements of each data model. The results of the development and application of the data models to the demonstrator process, in accordance with the Competency Questions, are presented in Section 8.2.

8.1 Data models

Both developed data models were designed to meet the defined requirements of the four use cases. They aimed to provide a comprehensive description of steel detailing processes, support machine-based fabrication, including robotic fabrication, and incorporate the necessary machine instructions to enable these processes. They also facilitated process feedback for optimisation, quality control and future deconstruction. The brief summary of the DSTV-NC standard as well as the key features of the IFC Fabrication View and the DSTV ontology are described below.

8.1.1 DSTV-NC standard

The DSTV-NC standard plays a crucial role in enabling steel fabrication by providing a simple and structured data format. However, it operates as an isolated file, disconnected from the overall building model. As a result, it functions as a data silo, limiting interoperability and preventing integration with broader digital workflows. The present file exclusively encompasses fundamental fabrication processes and characteristics. It is incapable of incorporating novel methodologies or additional data.

The development of data models has been undertaken to address the identified limitation, with the aim of establishing a more connected approach. The objective of this was to enable better interoperability, automation in robotic fabrication, and process feedback. It is important to note that not all fabrication-specific data needs to be embedded within the building model, and conversely, not all building-related information is required for machine operations. However, a more integrated and linked data exchange is necessary.

8.1.2 IFC Fabrication View

The IFC Fabrication View for steel fabrication was developed with consideration for existing initiatives, e.g, the EM.11 standard of the American Institute of Steel Construction (AISC), which focuses on defining specific property sets for fabrication. Existing entities, such as *IfcVoidingFeature* and *IfcTask*, have been identified as appropriate for describing steel fabrication processes. Additional properties to provide a more detailed description of the elements, resources and activities have been defined. The first version of the Steel Fabrication View has been updated to the bSDD (see Figure 8.1). It is important to emphasise that the property sets presented in this thesis do not cover the entire demand or all fabrication possibilities. Instead, the primary objective was to demonstrate that specific property sets can be newly defined and assigned to existing entities. The digital information contained in this model, particularly the definition of tolerances and the ability to store real measurement data, enabled the integration of robotic fabrication processes. The development was based on the IFC 4x3 schema. The demonstration shown in Chapter 7 highlights the successful integration and display of new property sets in the existing file.

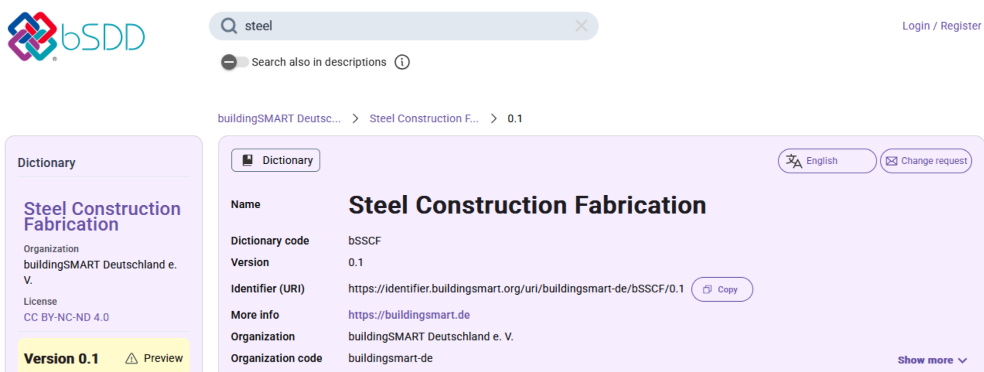


FIGURE 8.1: First version of steel fabrication specifications in the bSDD

8.1.3 DSTV ontology

Building upon the first version of the DSTV ontology, this thesis has focused on developing an extension to the domain ontology that aligns with specific requirements. The design process was characterised by a high degree of flexibility, which enabled the introduction of new classes and relationships. However, to mitigate redundancy and ensure the integrity of existing definitions, it was essential to leverage the reuse and integration of established ontologies. Evidence has demonstrated that the existing ifcOWL schema can be utilised within the broader framework of general definitions, for instance, for profiles and their integration into the overarching model. Concurrently, specific object and data properties, as well as their relationships, were newly defined. Figure 8.2 presents the pubslih availale DSTV ontology [25]. As demonstrated in Chapter 7, the instantiated ontologies can be stored in the triplestore and queried for robotic fabrication. Furthermore, the integration of an update query facilitated the consistent playback of new information from the fabrication process for the purpose of documentation and further utilisation.

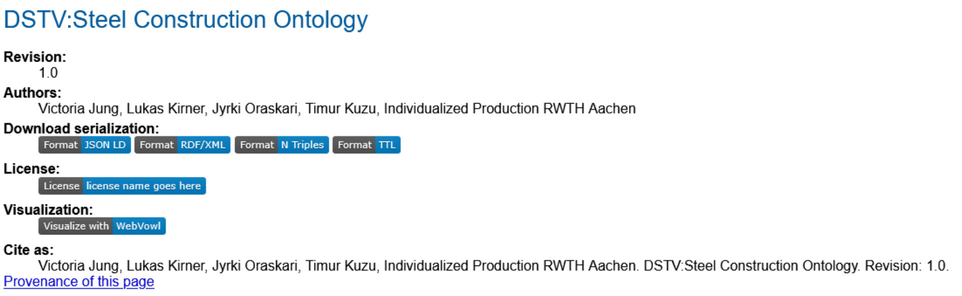


FIGURE 8.2: Revised version of DSTV ontology [25]

8.2 Demonstration

The following section presents the key findings from the demonstration of robotic fabrication based on the IFC and Linked Building Data frameworks. It focuses on the response to the set Competency Questions (CQ). Table 8.1 provides a summary of where to find the information within the structures.

The Industry Foundation Classes schema already provides standardised entities for representing general project information. Consequently, the first five Competency Questions could be addressed using the existing IFC file. The selected sample beam was classified as an element assembly; therefore, the identifier of the steel component corresponded to the GUID attribute of the *IfcElementAssembly* entity. This designation, including both the entity name and its associated value, remained consistent even when mapped through ifcOWL. If specific material information would be available, such as the storage or supplier data proposed in this thesis, it may be stored within an additional property set (e.g., *Pset_MaterialSteel*) or encoded using the DSTV class *dstv:Material*. General data could be retrieved from *IfcMaterial*. The geometry of the beam was defined using the *IfcProfileDef* entity, with specific subtypes representing various profile shapes; in this case, for instance, *IfcIShapeProfileDef* was used. This structure was also represented within the ifcOWL ontology. As defined in the IFC Fabrication View, all features were material is removed from the steel workpiece needed to be defined as *IfcVoidingFeature*. Therefore, the holes and the cutout in this example were also defined as voiding feature with

TABLE 8.1: Competency Questions and corresponding data sources

No.	Type	Competency Question	Representation (Ontology / IFC)
1	General Information	What is the ID of the steel part?	*GUID of the IfcElement*
2	General Information	What is the software it was detailed in?	*IfcApplication*
3	Material	What is the material?	*IfcMaterial*
4	Geometry	What is the overall geometry of the workpiece?	*IfcProfileDef*
5	Fabrication Feature	What are the features of the workpiece?	*IfcVoidingFeature*
6	Fabrication Feature	What is the planned value of the diameter?	*dstv:hasPlannedValue* / *Pset_ThroughHole*
7	Process	What are the processes linked to cutout?	*ioc:Process* / *IfcTask*
8	Process	What are process parameters for drilling?	*ioc:InputInformation* / *Pset_DrillingProcess*
9	Process	What is the scheduled time for drilling?	*ioc:Status* / *IfcTaskTime*
10	Process	What machine was used for the plasma cutting process?	*ioc:Resource* / *IfcResource*
11	Measurement Process	What is the real value of the height of the cutout	*dstv:hasMeasuredValue* / *Pset_Cutout*
12	Adjustment Process	What is the defined tolerance for a cutout?	*dstv:hasUpperToleranceLimit* / *Pset_Cutout*
13	Process Feedback	When did the plasma cutting process take place?	*ioc:Status* / *IfcTaskTime*
14	Process Feedback	What was the average speed of the robot while plasma cutting?	*ioc:hasOutputInformation* / *Pset_PlasmaCutting*

given pre-defined type of *HOLE* and *CUTOUT*. The data could be queried using IfcOpenShell, located within a STEP file, or simply viewed using any IFC-compatible viewer. Additionally, SPARQL queries could be defined to retrieve the data from the triple store. For the following Competency Questions (6-14), sample IfcOpenShell and SPARQL queries are presented.

8.2.1 IFC data extraction

The following subsection presents the outcomes of the IfcOpenShell-based queries applied to the selected IFC file. These queries were specifically designed to extract the necessary information required to address Conceptual Questions (CQ) 6 through 9. In addition to the programmatic results, visual representations of the IFC model are also provided in the form of screenshots. These visualisations, generated using an IFC viewer, serve to complement the query results by illustrating the spatial and structural context of the extracted elements. Together, the code output and the visual evidence offer a comprehensive understanding of the relevant data retrieved from the IFC model.

The Python function presented in Listing 8.1 employed the IfcOpenShell module to extract the planned diameter of a hole feature, specifically the *IfcVoidingFeature* named *Hole_01*, from the IFC file. The function first identified the relevant object within the model. It then traversed the associated property sets to locate the *Pset_ThroughHole* property set. Within this set, it searched for the property named *Diameter*, whose value represented the planned diameter of the hole. This method enabled targeted access to parameterised geometric information embedded within the IFC structure.

The result of the query was : 'Planned Diameter for Hole_01: 10.0'

LISTING 8.1: IfcOpenShell to get the planned diameter of Hole_01

```
import ifcopenshell

def get_planned_diameter(ifc_file_path, voiding_feature_name="Hole_01"):
    ifc_file = ifcopenshell.open(ifc_file_path)
    hole = next((v for v in ifc_file.by_type("IfcVoidingFeature") if v.Name ==
     voiding_feature_name), None)

    if not hole:
        print(f"No voiding feature named '{voiding_feature_name}' found.")
        return None

    for rel in hole.IsDefinedBy:
        if rel.is_a("IfcRelDefinesByProperties"):
            pset = rel.RelatingPropertyDefinition
            if pset.is_a("IfcPropertySet") and pset.Name == "Pset_ThroughHole":
                for prop in pset.HasProperties:
                    if prop.Name == "Diameter":
                        print(f"Planned Diameter for {voiding_feature_name}: {prop.
     NominalValue.wrappedValue}")
                        return prop.NominalValue.wrappedValue

    print(f"Diameter property not found for {voiding_feature_name}.")
    return None
get_planned_diameter(
    r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_modified.ifc"
)
```

Similar to the previous example, the script shown in Listing 8.2 employed the IfcOpenShell library to extract information from the IFC model, this time with a focus on identifying the planned processes associated with a specific cut-out. After locating the relevant *IfcVoidingFeature*, the script searched for any *IfcRelAssignsToProcess* relationships that linked this feature to one or more planned processes. This approach facilitated the retrieval of process-related data directly tied to the selected geometric element.

The result of this query was: 'Processes found for Cutout_01: Plasma Cutting Task (Type: IfcTask, Status: NOTSTARTED)'

This shows that the Cutout_01 feature was linked to a planned process called *Plasma Cutting Task*, which had the status *NOTSTARTED* at the time of the query.

LISTING 8.2: IfcOpenShell to get processes linked to the Cutout_01

```
import ifcopenshell

ifc_file = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\
    IFC4_Sample Beam_updated.ifc")

target_name = "Cutout_01"

voiding_feature = next((v for v in ifc_file.by_type("IfcVoidingFeature") if v.Name
    == target_name), None)

if not voiding_feature:
    print(f"Kein IfcVoidingFeature mit Name '{target_name}' gefunden.")
else:
    processes = []
    for rel in ifc_file.by_type("IfcRelAssignsToProcess"):
        if voiding_feature in rel.RelatedObjects:
            processes.append(rel.RelatingProcess)
    if processes:
        print(f"Gefundene Prozesse f r {target_name}:")
        for p in processes:
            print(f"- {p.Name} (Typ: {p.is_a()}, Status: {getattr(p, 'Status', '-')
    })")
    else:
        print(f"Keine Prozesse mit {target_name} verkn pft.")
```

Figure 8.3 showed screenshots from the KIT Model Viewer, illustrating where the data relevant to answering Conceptual Question 8 could be found. In this case, the property set provided information concerning the process parameters of the drilling operation.

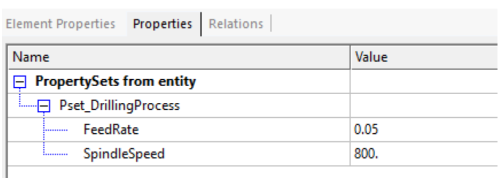
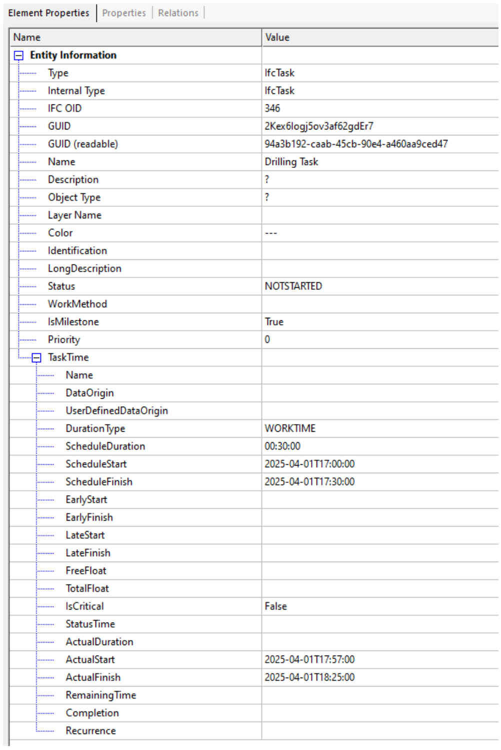


FIGURE 8.3: Visual representation of the defined properties of *Pset_DrillingProcess*

In line with the IFC schema, the *IfcTask* entity, together with the associated *IfcTaskTime*, stored information on both the scheduled start and end times, as well as the actual execution times of the process (see Figure 8.4).


Element Properties | Properties | Relations

Name	Value
Entity Information	
Type	IfcTask
Internal Type	IfcTask
IFC OID	346
GUID	2Kex6logj5ov3af62gdEr7
GUID (readable)	94a3b192-caab-45cb-90e4-a460aa9ced47
Name	Drilling Task
Description	?
Object Type	?
Layer Name	
Color	---
Identification	
LongDescription	
Status	NOTSTARTED
WorkMethod	
IsMilestone	True
Priority	0
TaskTime	
Name	
DataOrigin	
UserDefinedDataOrigin	
DurationType	WORKTIME
ScheduleDuration	00:30:00
ScheduleStart	2025-04-01T17:00:00
ScheduleFinish	2025-04-01T17:30:00
EarlyStart	
EarlyFinish	
LateStart	
LateFinish	
FreeFloat	
TotalFloat	
IsCritical	False
StatusTime	
ActualDuration	
ActualStart	2025-04-01T17:57:00
ActualFinish	2025-04-01T18:25:00
RemainingTime	
Completion	
Recurrence	

FIGURE 8.4: Visual representation of *IfcTask*, *IfcTaskTime*

In summary, this approach facilitated both the automated querying of data for processing purposes and the visual inspection of information embedded in newly defined property sets. Additionally, it supported the effective utilisation of existing IFC entities relevant to steel fabrication.

8.2.2 SPARQL query results

Following the presentation of the results and data retrieval process for the initial competence questions using the IFC model, Competence Questions 10 to 14 are addressed below through the application of SPARQL queries. It is important to note that the remaining questions could likewise be answered using data from the triplestore. The selection of CQ 10–14 serves merely as a representative example to demonstrate the effectiveness and applicability of the underlying data model.

To identify fabrication processes and resources related to plasma cutting (see CQ 10), a SPARQL query was executed that filtered all instances of *ioc:Process* whose URIs contained the substring 'plasma' (see Listing 8.3). As the processes were not explicitly labelled, the filter was applied directly to the string representation of the process URIs using the *CONTAINS* and *LCASE* functions. Additionally, the query retrieved the corresponding resources linked via the *ioc:resource* property, enabling a more complete view of the associated fabrication context.

LISTING 8.3: SPARQL query to retrieve resource of plasma cutting process

```
PREFIX ioc: <http://w3id.org/ioc#>

SELECT ?process ?resource
WHERE {
  ?process a ioc:Process .
  OPTIONAL { ?process ioc:resource ?resource . }
  FILTER(CONTAINS(LCASE(STR(?process)), "plasma"))
}
```

The results of the query are presented in Table 8.2. A total of four processes were identified for the plasma cutting operation. As outlined in Chapter 7, the overall process was subdivided into three sub-processes, all of which were executed using the same KUKA robot and consequently shared the same resources.

TABLE 8.2: Plasma cutting processes and their associated resources

Process (URI)	**Resource**
http://baufest.org/dstv-test02BeamFabrication_010_Plasma_292	KR30-3_F
http://baufest.org/dstv-test02BeamFabrication_010_Plasma_292_Approach	KR30-3_F
http://baufest.org/dstv-test02BeamFabrication_010_Plasma_292_Cut	KR30-3_F
http://baufest.org/dstv-test02BeamFabrication_010_Plasma_292_Retract	KR30-3_F

As demonstrated in the previous chapter, Table 7.9 presents the update of the measured values for the cutout within the triplestore. The following query, which addresses Competence Questions 11 and 12 (see Listing 8.4), retrieved not only the measured values but also the planned values and the defined tolerances. This enabled verification of whether the fabrication adhered to the specified limits. The use of OPTIONAL clauses ensured that the query continues to return planned values even in cases where measured values have not yet been recorded in the triplestore.

LISTING 8.4: SPARQL query to measurable property value for the cutout

```
PREFIX inst: <http://baufest.org/inst#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT Distinct ?plannedValue ?measuredValue ?upperTolerance ?lowerTolerance
WHERE {
  inst:HeightCutout_292 dstv:hasPlannedValue ?plannedState .
  ?plannedState qudt-1-1:numericValue ?plannedValue .

  OPTIONAL {
    inst:HeightCutout_292 dstv:hasMeasuredValue ?measuredState .
    ?measuredState qudt-1-1:numericValue ?measuredValue .
  }
    ?toleranceLimit dstv:hasUpperToleranceLimit ?upperToleranceState .
    ?upperToleranceState qudt-1-1:numericValue ?upperTolerance .

    ?toleranceLimit dstv:hasLowerToleranceLimit ?lowerToleranceState .
    ?lowerToleranceState qudt-1-1:numericValue ?lowerTolerance .
}
```

The results in Table 8.3 demonstrate that the cutout dimensions were in range within the upper and lower limit of the tolerance for the cutout.

TABLE 8.3: Measurable property values of the cutout in mm

Planned Value	Measured Value	Upper Tolerance	Lower Tolerance
80.0	80.3	3.0	0.0

The following Listing 8.5 represents the SPARQL query that was utilised in order to obtain an answer to CQ 13. It is imperative to note that the finished time of a process was not only relevant for the internal processing, but also for the successors of steel fabrication. The utilisation of the *ioc:Status* concept (e.g., ready or finished) resulted in the generation of a timestamp in the event of a status change. Table 8.2 summarises all the processes associated with plasma cutting. Therefore, it was possible to query the timestamp of the cutting process once the final sub-step of the three-step process ('Retract') has been completed.

LISTING 8.5: SPARQL query for timestamp of finished plasma cutting process

```
PREFIX ioc: <http://w3id.org/ioc#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX inst: <http://baufest.org/dstv-test02#>

SELECT ?timestamp
WHERE {
  inst:_Status_f089aee8-ed29-50d4-ac6e-54c77a3b37ff
    prov:generatedAtTime ?timestamp ;
    ioc:hasStatusValue inst:Status_f089aee8-ed29-50d4-ac6e-54c77a3b37ff .

  inst:Status_f089aee8-ed29-50d4-ac6e-54c77a3b37ff ioc:isFinished ?v .
  FILTER (LCASE(STR(?v)) = "true")
}
```

The Table 8.4 shows the timestamp associated with *ioc:hasStatus* in the 'Retract' process. The time value represented the point in time when the boolean for the *ioc:isFinished* value became true.

TABLE 8.4: Timestamp of finished plasma cutting process

Status Instance (URI)	**Timestamp**
http://baufest.org/dstv-test02_Status_f089aee8-ed29-50d4-ac6e-54c77a3b37ff	2025-03-27T18:36:20.082336+01:00

The final Competency Question 14 also focused on process feedback and documentation. It asked for the average robot speed during the plasma cutting process. The table 7.1 summarises the values set for the two cuts. In this case, ID 292 was relevant as it was chosen as the one to be executed with LBD. The planned value was set to 6 mm/s. To retrieve the actual value, the following Listing 8.6 shows the query. As described above, the 'Cut' process was known and could be used to directly query the 'Cut' process and the average robot speed, which was stored as *ioc:Information* for this process.

LISTING 8.6: SPARQL query average robot speed

```
PREFIX qudt: <http://qudt.org/schema/qudt/>
PREFIX ioc: <http://w3id.org/ioc#>
PREFIX inst: <http://baufest.org/dstv-test02#>

SELECT ?speedValue ?unit
WHERE {
  inst:BeamFabrication_010_Plasma_292_Cut_speedOut a qudt:QuantityValue,
     ioc:Information ;
      qudt:numericValue ?speedValue ;
      qudt:unit ?unit .
}
```

The result of the query is displayed in Table 8.5. The value differed from the planned value (which was defined as 6 mm/s see Table 7.2), as the robot slowed down at the corners of the rectangular cutout.

TABLE 8.5: Average speed value of the plasma cutting process in mm/s

Speed Value	**Unit**
5.57	mm/s

All defined Competency Questions were successfully addressed, covering the full range of previously identified requirement categories. The queries were executed using both the IFC-based model and the Linked Building Data approach with ontology representations, demonstrating the practical applicability of both data structures. In the examples provided, each question was answered using one of the two models — either IFC or LBD — selected according to suitability and clarity for the given context. However, it was confirmed that all Competency Questions could, in principle, be answered through both approaches, thereby underlining the interoperability and coverage of the developed data model. The developed models enabled end-to-end retrieval of all data required for automated robotic fabrication, including semantically enriched, queryable information for reuse and validation. This validated the feasibility of the proposed multi-modal strategy and supported a fully integrated digital workflow in steel construction.

Furthermore, it was demonstrated that the developed data models encompassed all fabrication-relevant data contained in the legacy DSTV-NC file format and significantly exceeded its informational scope. By incorporating semantic relationships, higher-level fabrication logic, and connections to lifecycle data, the approach offered a more flexible, transparent, and future-proof solution—supporting both reuse-oriented planning and fully automated robotic execution.

Chapter 9

Evaluation

The following sections provide an in-depth analysis of the results based on the defined evaluation criteria, which have been presented in Section 2.7. Following a detailed evaluation of each data model, the two are compared and their potential as successors to the DSTV-NC standard is discussed (see Section 9.1). In addition, Section 9.2 will highlight the limitations and challenges encountered during the research and critically review the approach and developments within this thesis. An important aspect of successfully integrating the developed model into the industry will be addressed within this chapter, as one of the criteria for evaluation, although a more detailed discussion is provided in Chapter 10.

9.1 Evaluation of results

The presented results in Chapter 8 as well as the development process of both data models in accordance with the two perspectives and set criteria (see Section 2.7) are going to be evaluated in the following section. The final focus lays on the discussion weather one or the other is suitable to be a successor to the existing DSTV-NC standard in steel fabrication.

The evaluation follows a structured approach that integrates analytical and comparative methods. Within the analytical framework, an empirical evaluation is incorporated by demonstrating the applicability of the models in real-world scenarios, specifically in robotic execution. Finally, a comparative evaluation will be performed to benchmark the models against existing DSTV-NC standard.

Figure 9.1 employs Harvey Balls to visually represent the degree to which each model fulfils the defined criteria, which are then discussed in greater detail in the following subsections.

9.1.1 IFC Fabrication View

The evaluation begins with an assessment of the technical criteria relevant to the proposed IFC Fabrication View, with particular emphasis on the criterion of completeness. This involves determining whether the specified requirements have been met and whether the corresponding Competency Questions (CQs) can be effectively answered. Based on the current implementation, it is evident that the set requirements and CQs have largely been addressed. However, the view does not yet encompass the full scope of steel fabrication requirements. Certain

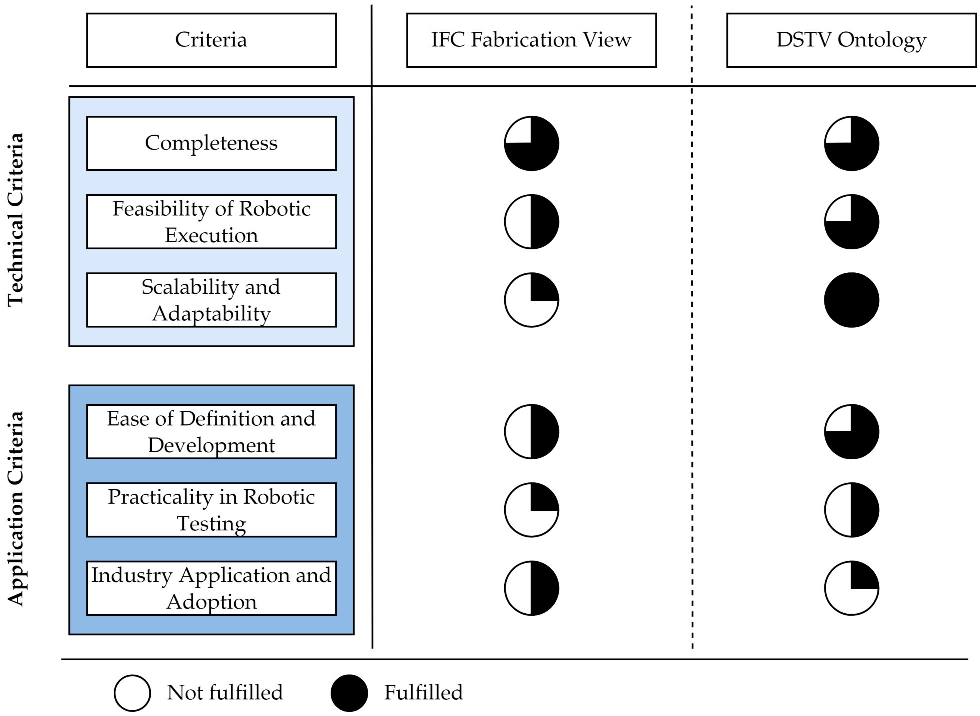


FIGURE 9.1: Visual evaluation of the two data models using Harvey Balls

fabrication-specific elements and details have not been modeled, which would be necessary to fully satisfy this criterion in a comprehensive and practical context. Therefore, while the implementation demonstrates strong alignment with the defined objectives, it does not yet achieve complete fulfillment of the criterion.

Through the utilisation of both existing IFC entities and the definition and integration of novel property sets, the identified requirements could be comprehensively fulfilled. The robotic process was successfully executed, facilitated by the integration of the IFC model, via Geometry Gym, directly into the robotic planning environment Grasshopper. However, since path planning and verification still required manual intervention, a fully automated process was not feasible. Process-related data were stored using a Python script and subsequently written back into the IFC file using IfcOpenShell. It is important to reiterate that, in the context of this work, no integration with CDEs was established. As such, the updated IFC data could not be shared with other stakeholders in a streamlined manner. Future developments should aim to enable such interoperability, thereby mitigating the risk of data loss due to locally stored files. Modifications to the IFC file currently necessitate the generation of a new file, which poses a risk of error propagation due to redundant or inconsistent updates to the native IFC data.

Within the scope of this research, two distinct fabrication features and associated processes were investigated. For each of these cases, specific property sets were defined to provide detailed descriptions of both the geometric features and the corresponding process information. In principle, this approach is extendable to other steel-specific fabrication processes, such as milling or welding. However, the data model remains constrained by existing IFC entities, with flexibility currently limited to the definition of new property sets. Provided sufficient familiarity with the IFC schema and structure, it is feasible to add relevant data for steel fabrication. The work carried out in this thesis, in collaboration with the ad hoc group *NC* of the Information Technology working committee of bauforumstahl e.V. and bS, contributed to the first publication of steel property sets within the buildingSMART Data Dictionary (bSDD). However, it indicates a clear need for further development and for formalisation to ensure practical applicability within industry workflows.

With the application criteria the emphasis shifts from the technical feasibility of the model to its applicability in practice. The development of standardised definitions inherently requires collaboration with industry partners, which can be a time-consuming and resource-intensive process. It necessitates coordinated efforts among technical experts, project coordinators, and industry representatives to drive both the definition and implementation phases. As steel prefabrication has not historically been a core focus of buildingSMART developments, the IFC Fabrication View presents an opportunity to better address the specific data exchange requirements of this sector. However, it remains challenging to accommodate all industry-specific demands in a generalised schema, as clients and contractors often formulate individualised requirements. While individual manufacturers may tailor their workflows to these bespoke needs, such practices rarely contribute to broadly applicable standards.

In terms of practical application during robot testing, the approach proved to be somewhat laborious, as update queries were required to incorporate process feedback into the model. It also became apparent that certain proposed implementations, such as using the property set *Pset_DrillingProcess* to store robot command sequences, were not ideal for this purpose. Representing a full set of control commands within the IFC model was both technically challenging and conceptually misaligned with the intended scope of the schema. Instead, it was found that storing basic geometric data about the feature, along with its overall spatial position, provided more practical and relevant information for integration into the IFC environment.

While IFC adoption has become more prevalent within the construction sector, knowledge of its benefits, as well as its limitations, is still unevenly distributed. In the context of German steel pre-fabrication, which is largely dominated by small and medium-sized enterprises, resources to adopt new digital technologies are often limited. The ability to adapt existing machinery to process IFC data is similarly constrained. Therefore, a multi-level approach is required: governmental support to enforce data collection and retention obligations, updates by machinery manufacturers to accommodate new data models, and demonstrable advantages in terms of both financial return and workforce efficiency.

The visual assessment of the Fabrication View against the defined criteria, along with the detailed evaluation, demonstrates that several aspects are already well (50%) or very well (75%) fulfilled. However, it also becomes evident that open questions remain, particularly regarding scalability, practical applicability in robotic processes, and whether the IFC format is indeed the appropriate choice for data used exclusively within fabrication environments. This section has already described approaches for achieving full compliance with the criteria in the future.

9.1.2 DSTV ontology

The evaluation of the extended DSTV ontology begins with an assessment of its technical criteria. In alignment with the earlier evaluation of the ontology itself, the model can be considered complete and correct to the extent that all defined Competency Questions were answerable and the robotic fabrication processes could be successfully executed within the scope of this thesis. However, as with the completeness assessment of the Fabrication View, certain aspects remain beyond the current implementation. In particular, the modelling of assembly processes has not yet been fully addressed, though initial work on this has already been undertaken in subsequent research efforts[211]. In terms of feasibility for robotic execution, the ontology proved capable of retrieving relevant information according to the defined schema, as well as storing relevant data. Future work will focus on a skills-based programming approach, enabling the use of a flexible data format and fabrication setup that does not require complicated programming or data modelling for each process and setup.

In transition to the application criteria the ontology was straightforward to define and implement, particularly by following the methodology described in *Ontology Development 101: A Guide to Creating Your First Ontology* and with the aid of visual tools such as Chowlk [137]. The proposed DSTV ontology has been published [25] and revisions can be made as necessary. The ease with which ontologies can be scaled and adapted is a key factor in achieving 100% fulfilment of the technical criteria of 'scalability and adaptability', as well as 75% for 'ease of definition and development' from users' perspectives (see visual representation in Figure 9.1). Since the chosen triplestore (GraphDB) can be hosted locally but also shared with stakeholders, collaboration was facilitated, enabling different users to both add and retrieve data, whether from the planning phase or during robotic execution, without the need for additional software tools.

Although Linked Building Data is not yet widely adopted within the construction industry, its integration offers considerable benefits. The data exchange format is both human- and machine-readable, and supports a flexible, extensible design. However, successful data retrieval using existing ontologies (e.g., ifcOWL) is dependent on prior modelling choices within the originating software. It is therefore essential to emphasise the use of standardised property sets and relationships, in order to ensure consistent linkage of additional classes to the DSTV ontology.

In summary, the evaluation shows that the developed ontology satisfies the core evaluation criteria to a high degree. This is particularly evident in terms of its completeness within the defined scope and its ease of scalability to future processes and requirements. The ontology supports the flexible and precise modelling of fabrication-specific data, enabling the successful execution of robotic processes. Although formal reuse and broader validation in industrial settings are yet to be realised, the modelling approach demonstrates significant potential for future applications. Furthermore, its compatibility with existing tools, ability to facilitate decentralised collaboration and extensibility for evolving fabrication requirements highlight the ontology's practical value within a multi-modal, automation-oriented context.

9.1.3 Successor to DSTV-NC

The DSTV-NC format has long been the de facto standard in steel pre-fabrication, providing a simple and reliable way to deliver machine-readable instructions directly to CNC machines. However, its simplicity is also its limitation: DSTV-NC focuses primarily on geometric and structural information, with very limited capacity for representing additional data about the fabrication process, stakeholders, materials, or feedback loops. Figure 9.2 represents the strength and shortcomings of the DSTV-NC format (orange lines). In contrast, both IFC (light blue lines) and Linked Data (dark blue lines) provide far richer data models. They are capable of describing not only the structure but also the context, processes, participants, and lifecycle of components in a construction project. This includes the potential to model and store process-related data, such as which machines are used in drilling, who supplied a specific steel profile, or at what time a part was produced. Furthermore, they support storing feedback from the fabrication process, enabling improved traceability, documentation, and reuse of knowledge in future projects. However, with this extended capability comes increased complexity. Both humans and machines face challenges in interpreting such detailed and multifaceted models. While it is theoretically possible to query any data point using SPARQL or codes, doing so requires knowing exactly what to look for and how it is modelled. If the data structure is not standardised or commonly agreed upon, even sophisticated query tools offer limited help. This increases the barrier for meaningful data extraction and reuse.

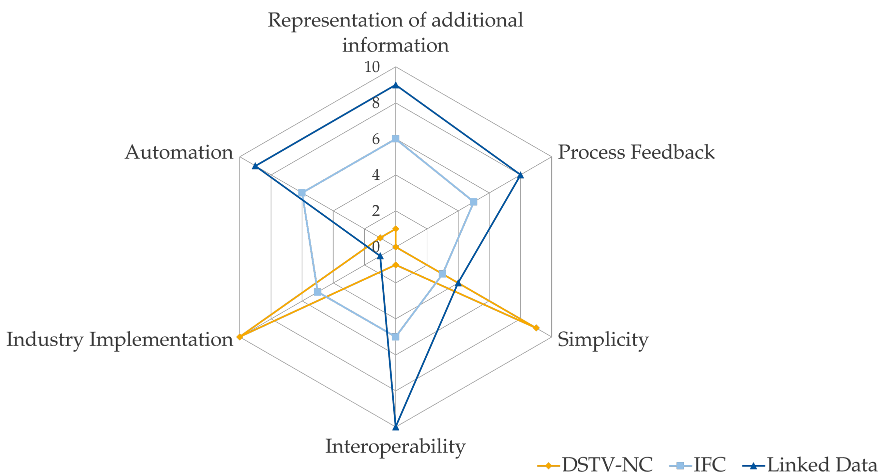


FIGURE 9.2: Radar diagram of strengths and limitations of DSTV-NC, IFC and Linked Data

IFC is a well-established open standard that has seen wide adoption across the architecture, engineering, and construction industry. It offers a structured, schema-based representation of buildings and infrastructure. It is already used as a backbone in many digital planning and coordination environments. However, the format is relatively rigid. Extending it to include custom or trade-specific data, such as supplier details or machine configurations, requires workarounds or dedicated property sets, which are not always standardised or interoperable. This can limit flexibility in highly specialised domains like steel pre-fabrication.

On the other side, ontologies and Linked Data represent a more flexible and semantic approach to modelling information. It allows users to define and link data from different sources, domains, and disciplines, which makes it particularly suited for capturing fabrication-level details, contextual information, or interdependencies. For instance, with LBD it is straightforward to model that a specific steel beam was cut using a certain type of machine, delivered by a specific supplier, and later passed a quality inspection. These relationships are easy to represent and update, and they can be queried using standard semantic web technologies.

In summary, both IFC as well as Linked Data and the use of ontologies are technically capable of replacing DSTV-NC by offering a more comprehensive and integrated data model for steel pre-fabrication. Yet, each comes with its own strengths and trade-offs. The present advantages of the simplicity of the old format and its wide industry adoption are counterbalanced by the fact that it the new models are capable of representing more information.

A combined approach is therefore recommended:

- Use IFC to represent the core building and fabrication information that is relevant to all stakeholders, and maintain this model within a CDE. This ensures interoperability, stability, and accessibility across disciplines and software platforms.
- Use ontologies (DSTV) to model trade-specific and flexible data, such as the machine configurations, processes, schedules and stakeholders. These details are often too specific or dynamic to be efficiently modelled in IFC alone.
- Feed back only the essential process outcomes, like time of production, status of quality checks, or measured dimensions, into the IFC model to ensure that the lifecycle history is preserved and available for future use and reference.

This layered strategy allows for a rich yet manageable digital representation of the steel fabrication process, ensuring both standardisation and adaptability. It enables collaboration across disciplines while supporting innovation and detailed process integration where required, ultimately providing a robust path beyond the limitations of DSTV-NC.

9.2 Limitations and challenges

Whilst the present PhD thesis proposes a comprehensive approach to developing standardised data models for steel pre-fabrication, certain limitations remain. The establishment of these data models was driven by well-established frameworks for data modelling and ontology design. The primary focus of these models was on enabling interoperability and increasing the data basis for meeting the increasing requirements of steel fabrication. This process was guided by a structured requirements analysis conducted in close collaboration with stakeholders from the German steel fabrication industry. While this provided focused and practice-oriented input, it also inherently limited the diversity of perspectives, particularly in light of differing international practices, regulatory frameworks, and levels of technological readiness.

The beam sample and associated fabrication features considered in this work represent only a narrow section of the broader steel pre-fabrication landscape. While they were deliberately chosen to demonstrate the applicability of two new data models as successors to the DSTV-NC file format, they cannot fully account for the wide variety of components and processes present in industrial practice. Therefore, generalisability is limited, and further case studies are required to validate the model's robustness across diverse use cases. The provided IFC Fabrication View does not cover all requirements, nor haven been IDS or mvdXML developed to fully provide all documents for a holistic standardisation approach in buildingSMART. While the thesis builds upon existing industry standards, it does not redefine core data structures, and several interoperability mechanisms remain underdeveloped. The integration of Linked Data principles, though innovative, has not yet been tested in large, multi-stakeholder environments.

The experimental setup was conducted on a relatively small scale, using a single KUKA robotic robot within a laboratory environment. While this enabled controlled and detailed analysis, it does not reflect the full complexity of an industrial-scale fabrication line, where requirements regarding process synchronisation, timing, and machine-to-machine data handover are significantly more demanding. Consequently, scalability remains an open question, and further testing in real-world manufacturing environments is necessary. Additional testing, especially across a broader range of fabrication processes, may reveal further insights into which data and structural elements are critical to model. It is likely that such practical implementations will uncover missing semantics, limitations in current property set definitions, and the need for improved interoperability mechanisms. These insights will be essential to refining the model and ensuring its relevance and effectiveness for industry-wide application.

A fundamental challenge in advancing horizontal and vertical digitalisation within the steel fabrication sector lies in translating technical progress into industrial implementation. The adoption of new data structures, demands not only technical validation but also sector-wide acceptance and standardisation. This, in turn, requires active collaboration among software vendors, machinery manufacturers, and end-users, as well as the definition of commonly accepted data exchange specifications. The next chapter will address this challenge. Equally important is the question of long-term governance and responsibility. As the reuse of steel components in construction becomes an increasingly relevant objective, particularly in light of circular economy principles, so too does the need for reliable data about a component's history. This introduces new roles and responsibilities, especially regarding data storage, lifecycle documentation, and traceability. Yet, it remains unclear who should bear the responsibility and associated costs for the collection, management, and maintenance of such data.

9.3 Response to RQ 3

The evaluation confirms that both the IFC-based Fabrication View and the ontology-driven DSTV model can be implemented to successfully execute robotic fabrication processes and store the resulting data. Each approach supports the transfer of required information for machine operations and allows process data to be recorded for further analysis. However, notable differences exist in terms of implementation complexity, data consistency, and scalability. The IFC-based approach integrates with established design environments and enables the definition of custom property sets. Yet, its implementation is more complex due to rigid schema structures, reliance on manual workflows, and file-based data handling, which increases the risk of inconsistencies and redundancy.
In contrast, the ontology-based approach offers a more structured and extensible solution. By leveraging semantic technologies and storing data as RDF triples in a triplestore, it enables consistent, machine-readable data management and easier collaboration. The modular design allows for scalable extension of process logic and supports continuous feedback from the fabrication process. In summary, while both models are technically viable for robotic fabrication, the ontology-based approach demonstrates clear advantages in terms of flexibility, data consistency, and scalability, making it a more robust foundation for future applications in digital steel construction.

The final evaluation highlights how the developed data models successfully address the requirements defined in response to Research Questions 1 and 2. The conceptual model established in RQ1 identified key data categories necessary to support interoperability and integrated automated fabrication in steel construction. These categories provided the foundation for both the IFC-based and ontology-based implementations.
The final evaluation highlights how the developed data models successfully address the requirements defined in response to Research Questions 1 and 2. The conceptual model established in RQ1 identified the key data categories needed to integrate for increased digitalisation in steel pre-fabrication. These categories guided the development of both the IFC-based and ontology-based implementations.

Chapter 10

Communication

buildingSMART and the Linked Building Data Community Group (LBDCG), which is part of the World Wide Web Consortium (W3C) are two key organisations that create and maintain standards for digital data exchange in the built environment. While both play a critical role in structuring and managing building-related information, they adopt fundamentally different approaches. The following chapter examines the characteristics and workflows of both organisations (see Sections 10.1 and 10.2), focusing on the practical steps and challenges involved in implementing the data model within the existing design and steel fabrication process. The analysis aims to show how the developments within this thesis can be supported by one or the other to contribute to standardisation, integration into industry practice and suggest a final procedure in Section 10.3.

This chapter covers the final step *Communication* of the applied DSR research method and answers the last research question within this thesis, which aims to communicate the findings to the wider community and encourage the implementation of the research. This step also involves transferring the results back to the knowledge base, enabling users in the relevant field to access them. This has been achieved by publishing the updated version of the ontology ([25]) and releasing the initial property set on the bSDD.

10.1 buildingSMART and IFC

buildingSMART is an international organisation that facilitates the development and implementation of open digital standards and services on a global scale. One of the standards that it promotes is Industry Foundation Classes [75]. To counter proprietary BIM solutions and their limit of data exchange within a specific software ecosystem, openBIM is a universal approach to collaborative design, construction, and operation of buildings based on open standards and workflows. It ensures that data remains accessible and usable across different platforms and stakeholders [219]. Besides the IFC, openBIM comprises additional standards and services, such as buildingSMART Data Dictionary (bSDD), Information Delivery Specification (IDS) or buildingSMART Use Case Management (bSUCM).

The provision of these services is generally aimed at ensuring interoperability. In the context of projects adopting an openBIM approach, there are workflows for both the application of existing concepts [220] and the definition of new terms within construction domain [221].

Fundamentally, the exchange of information must first be clearly defined, for which the Information Delivery Specification (IDS) can be utilised. IDS serves as a standard for defining information requirements in a format that is both human-readable and machine-interpretable [84]. Within the framework of the buildingSMART Data Dictionary (BSDD), it is then possible to verify whether general definitions already exist for a given requirement. If suitable definitions are available, a dataset can be created in IFC, which can subsequently be validated against the existing schema. The use of tools such as the BIM Collaboration Format (BCF) enhances collaboration among multiple stakeholders, facilitating the identification of conflicts and the communication of necessary adjustments. However, in cases where standards for a specific exchange requirements have not yet been established, an alternative process must be developed.

buildingSMART follows a structured process for developing consensus-based standards and solutions [221]. Broadly, this process comprises three key phases: initiation, solution development, and approval. The initiation phase involves the development of standard proposals outlining user needs and concepts, supported by use cases, work plans and stakeholder engagement strategies. During the development phase, a consortium carries out the work, conducting expert reviews to ensure alignment with technical, commercial and user requirements. Once endorsed by the relevant domain or group, the draft is reviewed by the Standards Committee Executive and, if approved, becomes a buildingSMART International (bSI) Candidate Standard. In the final approval phase, stakeholder consensus must be documented, with formal voting and notification rules guiding the process.

As this process of defining new property sets for the steel fabrication domain has been ongoing throughout the duration of the thesis, additional details of the process are provided below.

Initially, the identified gap within existing definitions must be formulated as a use case and registered in the buildingSMART Use Case Management system. Although buildingSMART provides a template for presenting the use case, this template is already quite comprehensive and includes not only a Business Process Model and Notation (BPMN) model to illustrate the information exchange process, but also a listing of the core information to be presented. For initiatives of general interest to buildingSMART, overarching meetings are organised by the respective responsible parties. However, the focus groups are independently responsible for determining both the technical and content aspects, such as which entities, attributes and properties already exist and can be used, or which new ones can be added without violating the schema specifications. They also need to work with domain experts who can help define broad, general requirements. Once an agreement has been reached and new property sets have been defined, as no changes to entities or attributes are allowed, contact must be made with buildingSMART to upload the definition to the bSDD and make it available to all. However, this alone is not sufficient as the new definitions are not yet known and have not been implemented in the existing software programs. The work of the focus group is finalised, once all required information and documents are summarised within the defined use case. This includes the final IDS with a reference to the definition in the bSDD and an mvdXML with the rules defined for this Fabrication View as a constraint on the existing schema for the use case. The published specifications are initially a preview on the bSDD to allow for changes after a testing and review period. After final approval, the specifications are set to active and from there only versions referencing the initial publications can be uploaded.

It is clear that the process of defining and, in particular, standardising IFC Fabrication Views, is neither straightforward nor simple. Although an infrastructure exists, it is inflexible and time consuming. In addition, an expert group needs to be established for both technical and content development.

10.2 Linked Building Data Community and ontologies

The World Wide Web Consortium (W3C) plays a key role in the development of web technologies and semantic data standards. In the built environment, the Linked Building Data (LBD) community extends this vision by creating and maintaining open, interoperable ontologies and data standards. These standards leverage semantic web technologies to improve data integration and interoperability across multiple domains within the Architecture, Engineering, and Construction (AEC) industry.

A fundamental principle of Linked Building Data is its modular approach to ontology development. This ensures flexibility and scalability in the representation of building-related information. At its core, the Building Topology Ontology (BOT) serves as a foundational framework, structuring essential building elements and providing a common basis for linking additional domain-specific ontologies. Through this approach, LBD enables a more connected and semantically rich digital representation of the built environment.

Processes, standardisation and exchange within Linked Building Data are organised through regular meetings, dedicated working groups and collaboration with buildingSMART as ifcOWL is officially registered with bS. ifcOWL is an official mapping of IFC schema EXPRESS into OWL [105, 198]. Monthly meetings are currently held where interested parties can discuss requirements, update existing ontologies and explore the feasibility of representing the building domain using semantic triplets. In addition, the annual Linked Data in Architecture and Construction (LDAC) conference brings together experts and newcomers to share recent work and contributions. In recent years, the LDAC Summer School has been established to provide hands-on training in Linked Data and ontologies.

Unlike IFC, which relies on predefined schemas, ontologies provide a more flexible structure while adhering to established semantic web standards such as RDF, RDFS, and OWL for defining and structuring ontologies. Tools such as the Chowlk plugin for draw.io and the Chowlk converter provide a visual approach to develop new ontologies, making the process more accessible. Recently, the LBD community launched a dedicated website where existing ontologies can be explored and referenced [26]. However, despite the community's ongoing efforts, awareness of ontologies and Linked Building Data within the construction industry remains limited. Many professionals are still unfamiliar with these concepts, and some are even unaware of BIM and IFC. Therefore, a key challenge is not only to publish and standardise new developments, but also to effectively promote their adoption within the industry.

10.3 Standardisation proposal - RQ 4

The construction industry has traditionally been slow to adopt new technologies. However, the growing need to improve productivity, address labour shortages, and respond to pressing environmental challenges is driving stakeholders to take action.

Despite the technical support in developing new standardised data representations for the rising requirements throughout the entire construction value chain for a linked and interoperable data model, the integration and adoption within industry remains the most challenging one. Based on the characteristics and workflows described for existing entities active in unifying the data exchange in construction, Figure 10.1 shows the proposed standardisation strategy for addressing current challenges. Steel Fabrication will now be used as an example to describe the overall process.

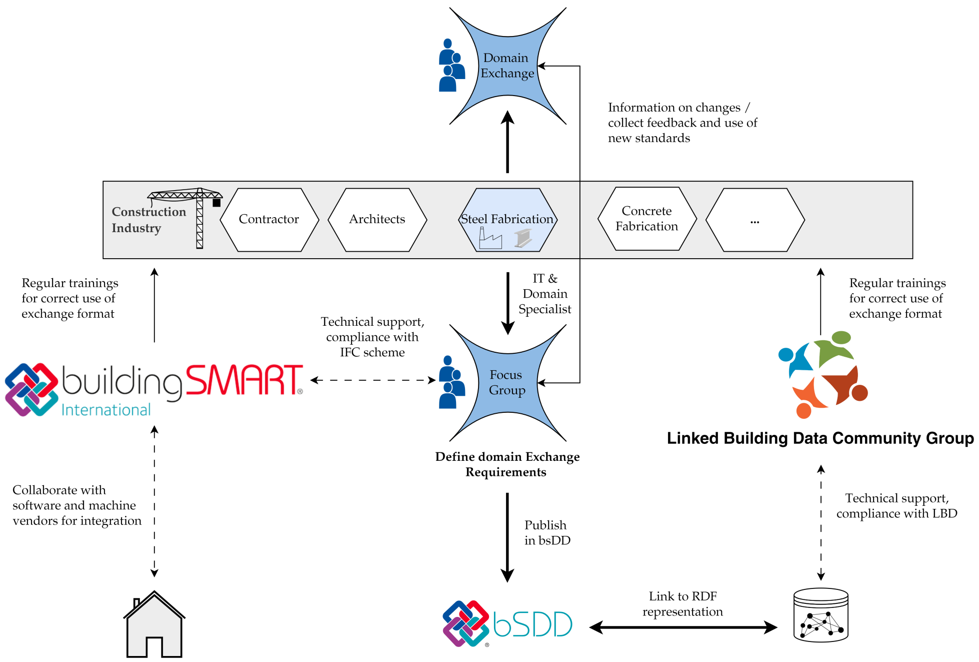


FIGURE 10.1: Proposed strategy for standardisation and implementation of exchange requirements in industry

Based on the bS process ([221]) a key approach to promoting new technologies is the formation of focus groups that represent the industry and actively support and drive technical developments. This ensures not only that the individuals and companies involved are aware at an early stage of the benefits of standardised data exchange within steel fabrication and its upstream and downstream processes but also actively define the requirements. An important aspect is that also IT experts are part of this group and technical support is enabled the buildingSMART to adhere compliance with the schema by defining new specifications. Ideally, new developments should be evaluated using real demonstrators to examine their practicality. Once successfully tested, the new exchange requirements are transferred to the buildingSMART Data Dictionary (bSDD). At this stage, a link can be made to Linked Building Data, as the standardised description of data is independent of its format. This allows properties to be represented in RDF. Through collaboration with the Linked Building Data Community Group, this foundation supports companies and stakeholders working with Linked Building Data. Although bS and LBDCG already collaborate, primarily through their agreement on the ifcOWL, future joint publications on data and information representation could further enhance alignment. As discussed in chapter 9, the use of domain-specific ontologies provides a structured approach to describing data and managing processes. This enables integration with the overarching IFC model, where only general stakeholder-relevant information is stored, while domain-specific data remains within its respective ontology. To ensure accessibility, complementary formats such as .jsonld can be used for data exchange. This approach provides a familiar and user-friendly format for most construction professionals, reducing the need to adopt alternative formats such as .ttl.

However, the publication of standardised exchange definitions, such as specifications for fabrication features or processes, does not automatically ensure their widespread adoption. To facilitate their implementation in practice, it is essential to work with both software vendors and industry associations to promote these developments effectively. Exchanging within domain allows feedback to be gathered to assess whether further refinement is needed. The results should be presented at national and international industry conferences, such as bauforumstahl e.V. committee meetings or sector-specific events such as Stahlbautag, and published in relevant trade journals, such as Stahlbauzeitung. These platforms also provide an opportunity to involve other industry stakeholders who may wish to contribute to further technical advances. In addition, close cooperation with machine suppliers is essential to ensure that their systems can correctly interpret and process the standardised input.

Tutorials or training sessions could also be offered, ideally in partnership with buildingSMART, bauforumstahl e.V. or the LBD community. This could be funded through membership fees for general training or through tailored, targeted programmes. It is also important to remember that many steel fabricators are small and medium-sized enterprises (SMEs) with limited IT and R&D resources, making it difficult for them to drive or implement such developments independently.

Another approach to promoting standardisation is through regulatory enforcement, where governments mandate the use of standardised definitions for data exchange. By implementing legal frameworks or industry regulations, authorities can ensure that all stakeholders in the construction and steel fabrication sectors adopt common data standards. Such regulatory measures could take various forms, including public procurement requirements that specify compliance with standardised data exchange formats, national or international building codes that incorporate digital data standards, or certification schemes that require compliance with specific ontologies or data models. Governments and regulators could also work with industry associations, such as bauforumstahl e.V. or buildingSMART, to develop guidelines and ensure that mandated standards are practical, technically feasible and beneficial to the industry.

The proposed strategy provides an initial answer to research question four by addressing key aspects of standardisation in the construction industry. However, its scope is limited, as this process has partly only been carried out once and in relation to a single industry in Germany. In order to determine its effectiveness in promoting standardisation and its implementation in the construction industry, it needs to be applied to other use cases and domains, including in other countries. The use of a standardised vocabulary ensures a clear and consistent structure for information retrieval, facilitating data reuse for deconstruction and interoperability. This enables users to find relevant information efficiently, while ensuring a common understanding among different stakeholders. As a result, ambiguity is minimised and the reliability of data interpretation is increased.

Chapter 11

Conclusion

This thesis focused on the development and evaluation of a multi-modal data modelling approach as a successor to the DSTV-NC format for steel fabrication data exchange. Emphasising interoperability, the approach integrates the data models to meet the evolving requirements of automated fabrication and align with the broader construction project data ecosystem. The proposed framework provides a system that supports process optimisation while ensuring accessibility and usability of fabrication data, process documentation, and steel element information for future reuse applications. By enhancing data availability and semantic consistency, this approach strengthens the foundation for assessing and facilitating the potential for steel component reuse.

Building upon the initial requirements analysis and the development of both the IFC Steel Fabrication View and the extended DSTV ontology, the resulting data models were successfully validated through robotic demonstrations. A subsequent evaluation of development efforts and outcomes against a range of criteria revealed that a hybrid approach offers the most suitable alternative to the existing DSTV-NC format. In this concept, core data relevant to all stakeholders is represented and stored in IFC, while domain-specific information can be more flexibly and efficiently modelled using ontologies. Key information can be integrated back into the overall data structure. However, a prerequisite for this is the standardisation of the necessary classes and relationships to ensure that all industry actors input and interpret data within property sets and entities consistently. buildingSMART provides a foundational framework for such standardisation through IFC, which can be adapted and expanded through collaboration with technical and domain experts in steel fabrication. A corresponding concept was outlined in Chapter 10.

The definitions and developments presented in this thesis contribute to the standardisation of steel-specific data exchange requirements, as well as to the data demands of robotic fabrication processes. Based on this foundation, and in collaboration with buildingSMART and bauforumstahl e.V., the first property sets for steel have already been published in the buildingSMART Data Dictionary. Furthermore, the DSTV ontology was updated to a first usable version, now publicly available and discoverable via a dedicated ontology search platform [25, 26]. Nonetheless, the practical adoption of both approaches will depend on targeted training, widespread awareness, and continuous maintenance.

The research questions formulated at the outset of this work were successfully addressed over the course of the thesis. The findings demonstrate that both the methodological approach and the close collaboration with industry partners were critical in generating meaningful insights. The results contribute to the ongoing standardisation and digital transformation of the steel construction industry and provide a foundation for further research.

Finally, I return to the work of Shannon, who asserted that the interpretation of data as information serves to reduce uncertainty [36]. In the context of steel fabrication, the availability of data enables not only the interpretation of design specifications for a steel plate and the determination of the real-world dimensions of a workpiece by a robot, thus avoiding collisions, but also the understanding by fabrication personnel of which tool was used at what speed for a given process. Moreover, process parameters such as energy consumption can be passed to an LCA database, and even decades later, inspectors may trace back when a steel beam was produced and what properties it possessed, assuming the beam can be reliably identified.

11.1 Outlook

Figure 11.1 illustrates the future objective of ensuring that critical data, from initial planning through pre-fabrication, first use, deconstruction, and secondary reuse, is centrally stored, linked, and accessible. Achieving this goal will require several steps, with a focus on standardising and maintaining open data exchange between stakeholders and project phases.

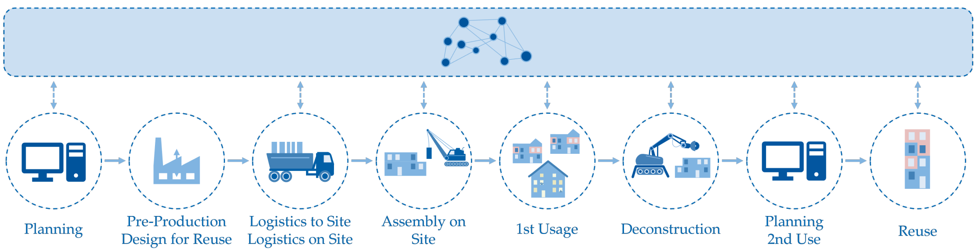


FIGURE 11.1: Towards a linked construction value chain enabling circularity

This thesis presented an initial approach using steel pre-fabrication, where two alternative exchange formats were developed, analysed and linked to the overarching BIM model as a replacement for an existing proprietary data format. Future research needs to focus on defining and implementing the data exchange and storage requirements for subsequent processes and phases. This applies not only to steelwork, but also to other disciplines such as civil engineering, reinforced concrete or building services. Various working groups within buildingSMART are already involved in defining, implementing and standardising the exchange requirements. However, as outlined in the Chapter 10, an essential aspect is also the dissemination and adoption of the defined standards to ensure that the proposed approach is effectively applied in practice.

An important aspect is the long-term storage of data to ensure its availability at the time of potential deconstruction. This raises the question of who is responsible for maintaining the data and who bears the associated costs. In this context, it becomes evident that not all recorded fabrication data are essential for integration into the overarching project exchange model or for long-term preservation. For instance, information on the tools used and their operational duration, which is valuable for internal process optimisation and maintenance within the fabrication facility, holds little relevance for other stakeholders. Therefore, only critical data should be retained over the long term. As analysed in Chapter 4, this includes key information such as the date of fabrication, material properties, actual positions and dimensions of features, and, most importantly, connection types and locations, which are essential for the future deconstruction of elements. Design for Disassembly [150] also plays a crucial role in enabling connections that can be dismantled without damage. In this context, bolted connections in steel construction, along with steel as a material, are particularly well suited for reuse within the framework of circular construction [222].

For the long-term storage of building data, it is crucial to adhere to uniform data formats and the principles of the openBIM approach to ensure that the data remains accessible over time and is not tied to specific software products. A potential solution for storing data and documents related to a BIM project is the Information Container for linked Document Delivery (ICDD) [223], as defined in DIN EN ISO 21597 [224, 225]. As a container format, it enables the structured and standardised organisation and delivery of all information, ensuring easy accessibility and usability. The ICDD allows not only the storage of data and documents, but also of ontologies, preserving the data structure and enabling the understanding of semantic relationships over time. In addition, the container facilitates the addition of new data, allowing the documentation of maintenance activities or changes to the object structure and individual building elements. Such data is crucial for assessing the potential for deconstruction and reuse at a later stage. However, the question remains as to who is responsible for storing and maintaining the availability of these data sets. For public buildings, a central body could be designated to manage building and infrastructure data sets. In the private sector, this is more challenging as ownership and responsibility may change over time. Nevertheless, regulatory frameworks could require that the database is always transferred and kept up to date during ownership transitions.

A collective effort is required to successfully drive forward the implementation of horizontal and vertical digitalisation, along with automation and efficiency improvements, in the construction industry. To achieve this, solutions must be understandable, especially for SMEs, with clear benefits and integration into existing workflows and structures. Collaboration between technical and domain experts, combined with a structured approach to standardisation and implementation, offers a promising strategy for transforming the construction industry.

Bibliography

[1] H. You et al. "Horizontal integration for steel fabrication compliance process". In: *Manufacturing Letters* 35 (2023). 51st SME North American Manufacturing Research Conference (NAMRC 51), pp. 1236–1245. ISSN: 2213-8463. DOI: https://doi.org/10.1016/j.mfglet.2023.08.067.

[2] K. Peffers et al. "A Design Science Research Methodology for Information Systems Research". In: *Journal of Management Information Systems* 24.3 (2007), pp. 45–77. DOI: 10.2753/MIS0742-1222240302.

[3] Y. T. Lee et al. "Information Modeling and Model Implementation". In: *Proceedings of the International Simulation Conference 2006* (2006). Accessed: 29.07.25. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=822353.

[4] M. Amirhosseini et al. "OntoAbsolute as a ontology evaluation methodology in analysis of the structural domains in upper, middle and lower level ontologies". In: *2011 International Conference on Semantic Technology and Information Retrieval*. (2011), pp. 26–33. DOI: 10.1109/STAIR.2011.5995760.

[5] J. Hietanen et al. "IFC model view definition format". In: *International Alliance for Interoperability* (2006). Accessed: 29.07.2025, pp. 1–29. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=59dbe4bd813c301aea5ce887f497211fe6a5a4ce.

[6] N. F. Noy et al. "Ontology Development 101: A Guide to Creating Your First Ontology". In: *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880* (2001). Accessed: 29.07.2025. URL: http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

[7] U. Awan et al. "Industry 4.0 and the circular economy: A literature review and recommendations for future research". In: *Business Strategy and the Environment* 30 (2021), pp. 2038–2060. DOI: 10.1002/bse.2731.

[8] M. Kozlovska et al. "Impact of Industry 4.0 Platform on the Formation of Construction 4.0 Concept: A Literature Review". In: *Sustainability* 13.5 (2021). ISSN: 2071-1050. DOI: 10.3390/su13052683.

[9] Q. K. Jahanger et al. "Potential positive impacts of digitalization of construction-phase information management for project owners". In: *Journal of Information Technology in Construction* 26 (2021), pp. 1–22. DOI: 10.36680/j.itcon.2021.001.

[10] S. Elhouar et al. "Leveraging advances in automation to benefit the construction industry in the structural steel sector". In: *IOP Conference Series: Materials Science and Engineering* 1218.1 (2022). DOI: 10.1088/1757-899X/1218/1/012008.

[11] S. Banihashemi et al. "Circular economy in construction: The digital transformation perspective". In: *Cleaner Engineering and Technology* 18 (2024). ISSN: 2666-7908. DOI: https://doi.org/10.1016/j.clet.2023.100715.

[12] United Nations Environment Programme. *2022 Global Status Report for Buildings and Construction*. Nairobi, (2022).

[13] M. Kuhnhenne et al. "Kreislaufwirtschaft im Stahl– und Metallleichtbau". In: *Stahlbau* 91.4 (2022), pp. 236–246. ISSN: 0038-9145. DOI: 10.1002/stab.202200013.

[14] A. Kanyilmaz et al. "Reuse of Steel in the Construction Industry: Challenges and Opportunities". In: *International Journal of Steel Structures* 23.5 (2023), pp. 1399–1416. ISSN: 1598-2351. DOI: 10.1007/s13296-023-00778-4.

[15] D. Densley Tingley et al. "Understanding and overcoming the barriers to structural steel reuse, a UK perspective". In: *Journal of Cleaner Production* 148 (2017), pp. 642–652. ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2017.02.006.

[16] bauforumstahl e.V. *Standardbeschreibung von Stahlbauteilen für die NC-Steuerung (ASCII)*. Accessed: 29.07.2025. (2003). URL: https://bauforumstahl.de/wp-content/uploads/2024/04/109_Standartbeschreibung_von_Stahlbau-Teilen_8.0.pdf.

[17] bauforumstahl e.V. *Standardbeschreibung von Stahlbauteilen für die NC-Steuerung (XNC): BFS-RL 03-105*. Accessed: 29.07.2025. (2006). URL: https://bauforumstahl.de/wp-content/uploads/2024/04/105_nc-xml.pdf.

[18] J. Jia et al. "Integration of Industry Foundation Classes and Ontology: Data, Applications, Modes, Challenges, and Opportunities". In: *Buildings* 14.4 (2024). ISSN: 2075-5309. DOI: 10.3390/buildings14040911.

[19] S. Brell-Cokcan et al., eds. *IoC - Internet of Construction: Informationsnetzwerke zur unternehmensübergreifenden Kollaboration in den Fertigungsketten des Bauwesens*. 1. Auflage 2024. Wiesbaden: Springer Fachmedien Wiesbaden GmbH and Springer Vieweg, (2024). ISBN: 9783658425432. DOI: https://doi.org/10.1007/978-3-658-42544-9.

[20] S. Stumm et al. "Robotergestütztes Schweißen – Verteilte Produktionstechnik für dynamische Automatisierung". In: *IoC - Internet of Construction : Informationsnetzwerke zur unternehmensübergreifenden Kollaboration in den Fertigungsketten des Bauwesens*. Ed. by S. Brell-Cokcan et al. Wiesbaden: Springer Fachmedien Wiesbaden, (2024), pp. 505–565. ISBN: 978-3-658-42544-9. DOI: 10.1007/978-3-658-42544-9_15.

[21] L. Kirner et al. "Enhancing robotic steel prefabrication with semantic digital twins driven by established industry standards". In: *Automation in Construction* 167 (2024). ISSN: 09265805. DOI: 10.1016/j.autcon.2024.105699.

[22] V. Jung. *D5.2 Prototype system for controlled deconstruction process*. (2024). DOI: 10.5281/zenodo.14615798.

[23] V. Jung et al. "Enabling Interoperability in Steel Construction – New Concepts for Using IFC for Steel Fabrication". In: *NSCC Sweden, NORDIC STEEL Construction Conference 2024 Luleå, Schweden*. (2024). DOI: https://doi.org/10.5281/zenodo.12509159.

[24] V. Jung et al. "openBIM – IFC in der Stahlbaufertigung". In: *Stahlbau* 94 (2025). ISSN: 0038-9145. DOI: 10.1002/stab.202500008.

[25] V. Jung et al. *DSTV: Steel Construction Ontology. Revision 1.0*. RWTH Aachen University, Chair of Individualized Production. Revision 1.0, Accessed: 04.08.2025. (2025). URL: https://ip.pages.rwth-aachen.de/ioc/dstv-ontologie/.

[26] Modelling & Standards (M&S) committee of EC3. *Built Environment Ontology Lookup Service (BE-OLS)*. Accessed: 29.07.2025. (2025). URL: https://cyberbuildlab.github.io/BE-OLS/.

[27] A. R. Hevner et al. "Design Science in Information Systems Research". In: *MIS Quarterly* 28.1 (2004), pp. 75–105. ISSN: 02767783. DOI: https://doi.org/10.2307/25148625.

[28] S. Mak. "A model of information management for construction using information technology". In: *Automation in Construction* 10.2 (2001). Multimedia in Construction, pp. 257–263. ISSN: 0926-5805. DOI: https://doi.org/10.1016/S0926-5805(99)00035-7.

[29] O. Tanga et al. "Data Management Risks: A Bane of Construction Project Performance". In: *Sustainability* 14.19 (2022). ISSN: 2071-1050. DOI: 10.3390/su141912793.

[30] H. Weber. "Methoden der Informationsmodellierung". In: *Data Engineering 4.0: Kompositionale Informationsmodelle für industrielle Anwendungen*. Wiesbaden: Springer Fachmedien Wiesbaden, (2021), pp. 57–202. ISBN: 978-3-658-33185-6. DOI: 10.1007/978-3-658-33185-6_3.

[31] R. L. Ackoff. "From Data to Wisdom". In: *Journal of Applied Systems Analysis* 16 (1989), pp. 3–9.

[32] R. J. Boland. "The in-formation of information systems". In: *Critical Issues in Information Systems Research*. USA: John Wiley & Sons, Inc., (1987), pp. 363–394. ISBN: 0471912816.

[33] P. Beynon-Davies. "What Is Information?" In: *Information Modelling: A Pragmatic Approach*. Cham: Springer International Publishing, (2022), pp. 9–34. ISBN: 978-3-030-98805-0. DOI: 10.1007/978-3-030-98805-0_2.

[34] F. I. Dretske. *Knowledge and the Flow of Information*. Cambridge, (1981).

[35] M. McCreadie et al. "Trends in analyzing access to information. Part I: cross-disciplinary conceptualizations of access". In: *Information Processing Management* 35.1 (1999), pp. 45–76. ISSN: 0306-4573. DOI: https://doi.org/10.1016/S0306-4573(98)00037-5.

[36] C. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27 (1948), pp. 379–423, 623–656.

[37] C. Shannon et al. *The Mathematical Theory of Communication*. 10. print. Urbana: Univ. of Illinois Pr., (1964).

[38] J. Jünger et al. "Datenformate". In: *Computational Methods für die Sozial- und Geisteswissenschaften*. Wiesbaden: Springer Fachmedien Wiesbaden, (2023), pp. 53–82. ISBN: 978-3-658-37747-2. DOI: 10.1007/978-3-658-37747-2_3.

[39] A. Pras et al. *On the Difference between Information Models and Data Models*. RFC 3444. (2003). DOI: 10.17487/RFC3444.

[40] D. C. Tsichritzis et al. *Data Models*. Prentice Hall Professional Technical Reference, (1982). ISBN: 0131964283.

[41] C. Koch et al. "Data Modeling". In: *Building Information Modeling: Technology Foundations and Industry Practice*. Ed. by A. Borrmann et al. Cham: Springer International Publishing, (2018), pp. 43–62. ISBN: 978-3-319-92862-3. DOI: 10.1007/978-3-319-92862-3_3.

[42] G. Booch et al. *Object oriented analysis and design with applications*. Third edition. The Addison-Wesley object technology series. Upper Saddle River, NJ: Addison-Wesley, (2007). ISBN: 0-201-89551-X.

[43] P. Shoval et al. "Entity-relationship and object-oriented data modeling — An experimental comparison of design quality". In: *Data Knowledge Engineering* 21.3 (1997), pp. 297–315. ISSN: 0169-023X. DOI: https://doi.org/10.1016/S0169-023X(97)88935-5.

[44] B. Thalheim. "Foundations of entity-relationship modeling". In: *Annals of Mathematics and Artificial Intelligence* 7 (1993), pp. 197–256. DOI: 10.1007/BF01556354.

[45] P. P.-S. Chen. "The entity-relationship model—toward a unified view of data". In: *ACM Transactions on Database Systems* 1.1 (1976), pp. 9–36. ISSN: 0362-5915. DOI: 10.1145/320434.320440.

[46] C. Koch. "Objektorientierte Modellierung". In: *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Ed. by A. Borrmann et al. Wiesbaden: Springer Fachmedien Wiesbaden, (2015), pp. 43–56. ISBN: 978-3-658-05605-6. DOI: 10.1007/978-3-658-05606-3_3.

[47] G. Engels et al. "Object-oriented modeling: a roadmap". In: *Proceedings of the Conference on the Future of Software Engineering*. New York, NY, USA: Association for Computing Machinery, (2000), pp. 103–116. ISBN: 1581132530. DOI: 10.1145/336512.336541.

[48] G. Booch et al. "The unified modeling language". In: *Unix Review* 14.13 (1996), p. 5.

[49] M. Udrică et al. "Issues Concerning the Use of UML Diagrams to Define the Underlying Process Model Simulation". In: *Annals of the University of Petrosani. Economics* 11.4 (2011). Accessed: 29.07.2025, pp. 305–314. URL: https://www.upet.ro/annals/economics/pdf/annals-2011-part4.pdf#page=305.

[50] M. Mani et al. "Semantic Data Modeling Using XML Schemas". In: *Conceptual Modeling — ER 2001*. Ed. by H. S.Kunii et al. Berlin, Heidelberg: Springer Berlin Heidelberg, (2001), pp. 149–163. ISBN: 978-3-540-45581-3. DOI: https://doi.org/10.1007/3-540-45581-7_13.

[51] P. V. Biron et al. *XML Schema Part 2: Datatypes Second Edition*. Accessed: 29.07.2025. (2012). URL: https://www.w3.org/TR/xmlschema11-2.

[52] C. Eastman. *Building product models: Computer environments supporting design and construction*. Boca Raton, FL, USA: CRC Press, (1999). ISBN: 9780849302596.

[53] International Organization for Standardization. *ISO 10303-225:1999 – Industrial automation systems and integration – Product data representation and exchange – Part 225: Application protocol: Building elements using explicit shape representation*. Accessed: 29.07.2025. (1999). URL: https://www.iso.org/standard/25091.html.

[54] A. Borrmann et al. "Industry Foundation Classes – Ein herstellerunabhängiges Datenmodell für den gesamten Lebenszyklus eines Bauwerks". In: *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Ed. by A. Borrmann et al. Wiesbaden: Springer Fachmedien Wiesbaden, (2021), pp. 95–146. ISBN: 978-3-658-33361-4. DOI: 10.1007/978-3-658-33361-4_6.

[55] bauforumstahl e.V. *Standartbeschreibung von Strukturen in der Stahlbau-Fertigung (Anbauschnittstelle): BFS-RL 03-108*. Accessed: 29.07.2025. (2013). URL: https://dstv.

deutscherstahlbau.de/fileadmin/user_upload/bauforumstahl.de/wissen/normen-und-richtlinien/bfs-richtlinien/108_Anbauschnittstelle-2013-10-24.pdf.
[56] N. Čuš Babič et al. "Culture change in construction industry: from 2d toward bim based construction". In: *Journal of Information Technology in Construction* 21 (2016). Accessed: 29.07.2025, pp. 86–99. URL: https://www.itcon.org/2016/6.
[57] F. Lévy. *BIM in small-scale sustainable design*. John Wiley & Sons, (2011). ISBN: 10:0470590890.
[58] J. Ingram. *Understanding BIM*. Routledge, (2020). ISBN: 9780429282300. DOI: 10.1201/9780429282300.
[59] L. F. Rochaa et al. "Beam for the Steel Fabrication Industry Robotic Systems". In: *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*. Ed. by M.-Y. Cheng et al. Taipei, Taiwan, (2017). DOI: http://dx.doi.org/10.22260/ISARC2017/0089.
[60] S. Ahmed. "Barriers to Implementation of Building Information Modeling (BIM) to the Construction Industry: A Review". In: *Journal of Civil Engineering and Construction* 7.2 (2018), pp. 107–113. DOI: 10.32732/jcec.2018.7.2.107.
[61] A. Borrmann et al. *Building information modeling: Technologische Grundlagen und industrielle Praxis*. VDI-Buch. Wiesbaden: Springer Vieweg, (2015). ISBN: 978-3-658-05605-6. DOI: 10.1007/978-3-658-05606-3.
[62] S. Stumm et al. "Towards Life Cycle Complete BIM". In: *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*. Ed. by M.-Y. Cheng et al. Taipei, Taiwan, (2017). DOI: 10.22260/ISARC2017/0088.
[63] N. O. Nawari. "BIM Standard in Off-Site Construction". In: *Journal of Architectural Engineering* 18.2 (2012), pp. 107–113. ISSN: 1076-0431. DOI: 10.1061/(ASCE)AE.1943-5568.0000056.
[64] *Nationales Zentrum für die Digitalisierung des Bauwesens (BIM) | BIM Deutschland*. Accessed: 29.07.2025. (2025). URL: https://www.bimdeutschland.de/.
[65] A. F. Kineber et al. "Challenges to the Implementation of Building Information Modeling (BIM) for Sustainable Construction Projects". In: *Applied Sciences* 13.6 (2023). DOI: https://doi.org/10.3390/app13063426.
[66] J. F. Tchouanguem Djuedja et al. "Interoperability Challenges in Building Information Modelling (BIM)". In: *Enterprise Interoperability VIII*. Ed. by K. Popplewell et al. Cham: Springer International Publishing, (2019), pp. 275–282. ISBN: 978-3-030-13693-2. DOI: https://doi.org/10.1007/978-3-030-13693-2_23.
[67] H. Kerosuo et al. "Challenges of the expansive use of Building Information Modeling (BIM) in construction projects". In: *Production* 25.2 (2015), pp. 289–297. DOI: 10.1590/0103-6513.106512.
[68] A. Borrmann et al. *TwinGen - Technologien zur Generierung digitaler Zwillinge als Grundlage für Betrieb und Instandhaltung baulicher Infrastruktur: Abschlussbericht*. Accessed: 29.07.2025. (2023). URL: https://publications.rwth-aachen.de/record/862155.
[69] B. Daniotti et al. "Workshop: BIM4EEB: A BIM-Based Toolkit for Efficient rEnovation in Buildings". In: *The 8th Annual International Sustainable Places Conference (SP2020) Proceedings* (2020). ISSN: 2504-3900. DOI: https://doi.org/10.3390/proceedings2020065017.

[70] L. Kirner et al. "Internet of Construction: Research Methods for Practical Relevance in Construction". In: *Technology | Architecture + Design* 5.2 (2021), pp. 146–152. ISSN: 2475-1448. DOI: 10.1080/24751448.2021.1967053.

[71] A. Mohammed et al. "Design for steel structures deconstruction: an analytics system for construction waste minimization in a circular economy through BIM technology". In: *Innovative Infrastructure Solutions* 9.11 (2024). ISSN: 2364-4176. DOI: 10.1007/s41062-024-01703-2.

[72] M. Cassano et al. "LOD Standardization for Construction Site Elements". In: *Procedia Engineering* 196 (2017). Creative Construction Conference 2017, Primosten, Croatia, pp. 1057–1064. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2017.08.062.

[73] International Organization for Standardization. *ISO 7817-1:2024 - Building Information Modelling — Level of Information Need, Part 1: Concepts and Principles*. Accessed: 29.07.2025. Geneva, Switzerland, (2024). URL: https://www.iso.org/standard/82914.html#lifecycle.

[74] BIMForum. *Level of Development, LOD-Specification, Part-I*. Accessed: 29.07.2025. (2024). URL: https://bimforum.org/wp-content/uploads/2024/11/LOD-Spec-2024-Part-I-official-English.pdf.

[75] International Organization for Standardization. *ISO 16739-1:2021-11 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries –part 1: data schema*. Accessed: 29.07.2025. Geneva, Switzerland, (2021). URL: https://www.iso.org/standard/84123.html.

[76] buildingSMART International. *IFC 4x3 Standard Documentation,* Accessed: 29.07.2025. (2022). URL: https://ifc43-docs.standards.buildingsmart.org/.

[77] M. Keller. "Informationstechnisch unterstützte Kooperation bei Bauprojekten". PhD thesis. TU Dresden, (2007). ISBN: 978-3-86780-004-4.

[78] S. Brell-Cokcan et al. "Transparency and Value of Data in Construction". In: *The Monetization of Technical Data: Innovations from Industry and Research*. Ed. by D. Trauth et al. Berlin, Heidelberg: Springer Berlin Heidelberg, (2023), pp. 539–558. ISBN: 978-3-662-66509-1. DOI: 10.1007/978-3-662-66509-1_30.

[79] Deutsches Institut für Normung e.V. (Hrsg.) *DIN EN ISO 29481-1:2018-01, Bauwerksinformationsmodelle_- Handbuch der Informationslieferungen_- Teil_1: Methodik und Format (ISO_29481-1:2016); Deutsche Fassung EN ISO 29481-1:2017*. Berlin, (2018). DOI: 10.31030/2773897.

[80] buildingSMART International. *Model View Definition (MVD) - An Introduction*. Accessed: 29.07.2025. (2025). URL: https://technical.buildingsmart.org/standards/ifc/mvd/.

[81] buildingSMART International. *An Integrated Process for Delivering IFC Based Data Exchange*. Accessed: 29.07.2025. (2012). URL: https://standards.buildingsmart.org/documents/IDM/IDM_guide-IntegratedProcess-2012_09.pdf.

[82] C. Zhang et al. "Towards model view definition on semantic level: A state of the art review". In: *Proceedings of the 20th International Workshop: Intelligent Computing in Engineering*. (2013).

[83] C. Eastman et al. “Introducing a new methodology to develop the information delivery manual for AEC projects”. In: *CIB W78* (2010), pp. 16–18.

[84] buildingSMART International. *Information Delivery Specification IDS - buildingSMART International*. Accessed: 29.07.2025. (2025). URL: `https://technical.buildingsmart.org/projects/information-delivery-specification-ids/`.

[85] J. Beetz et al. *Bimserver.org - an Open Source IFC model server*. Accessed: 29.07.2025. (2010). URL: `https://github.com/opensourceBIM/BIMserver`.

[86] T. Berners-Lee et al. *WorldWideWeb: Proposal for a HyperText project*. Accessed: 29.07.2025. (1990). URL: `https://www.w3.org/Proposal.html`.

[87] T. Berners-Lee, ed. *Weaving the web: The original design and ultimate destiny of the world wide web by its inventor*. 1. ed. San Francisco: Harper, (1999). ISBN: 978-0-06-251587-2.

[88] T. Berners-Lee. *Linked Data*. W3C Design Issues, Accessed: 29.07.2025. (2006). URL: `https://www.w3.org/DesignIssues/LinkedData`.

[89] L. Kirner. “An ontology-based approach for developing digital process twins in construction”. PhD thesis. RWTH Aachen University, (2024). DOI: `10.18154/RWTH-2024-10076`.

[90] R. Guns. “Tracing the origins of the semantic web”. In: *Journal of the American Society for Information Science and Technology* 64.10 (2013). DOI: `10.1002/asi.22907`.

[91] H. Sack. “Linked Data Technologien – Ein Überblick”. In: *Linked Enterprise Data: Management und Bewirtschaftung vernetzter Unternehmensdaten mit Semantic Web Technologien*. Ed. by T. Pellegrini et al. Springer Berlin Heidelberg, (2014), pp. 21–62. ISBN: 978-3-642-30274-9. DOI: `10.1007/978-3-642-30274-9_2`.

[92] P. Pauwels et al. “Linked Data”. In: *Building Information Modeling: Technology Foundations and Industry Practice*. Ed. by A. Borrmann et al. Springer International Publishing, (2018), pp. 181–197. ISBN: 978-3-319-92862-3. DOI: `10.1007/978-3-319-92862-3_10`.

[93] M. H. Rasmussen et al. “BOT: The building topology ontology of the W3C linked building data group”. In: *Semantic Web* 12.1 (2021), pp. 143–161. ISSN: 15700844. DOI: `10.3233/SW-200385`.

[94] J. Werbrouck et al. “Towards a decentralised common data environment using linked building data and the solid ecosystem”. English. In: *Proceedings of the 36th CIB W78 Conference on Information Technology in Construction*. Ed. by B. Kumar et al. (2019), pp. 113–123. ISBN: 9781861354877.

[95] T. Gruber. “A translation approach to portable ontology specifications”. In: *Knowledge Acquisition* 5.2 (1993). ISSN: 1042-8143. DOI: `10.1006/knac.1993.1008`.

[96] T. Gruber. “Ontology”. In: *Encyclopedia of Database Systems*. Ed. by L. LIU et al. Boston, MA: Springer US, (2009), pp. 1963–1965. ISBN: 978-0-387-39940-9. DOI: `10.1007/978-0-387-39940-9_1318`.

[97] *LargeTripleStores - W3C Wiki*. Accessed: 29.07.2025. (2025). URL: `https://www.w3.org/wiki/LargeTripleStores`.

[98] Z. Zhou et al. “Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry”. In: *Journal of Computing in Civil Engineering* 30.6 (2016). DOI: `10.1061/(ASCE)CP.1943-5487.0000594`.

[99] C. J. Anumba et al. "Ontology-based information and knowledge management in construction". In: *Construction Innovation* 8.3 (2008), pp. 218–239. DOI: 10.1108/14714170810888976.

[100] S. Lohmann et al. "Visualizing ontologies with VOWL". In: *Semantic Web* 7.4 (2016), pp. 399–419. ISSN: 15700844. DOI: 10.3233/SW-150200.

[101] R. Cyganiak et al., eds. *RDF 1.1 Concepts and Abstract Syntax*. Accessed: 29.07.2025. (2023). URL: https://www.w3.org/TR/rdf11-concepts/.

[102] D. Bickley et al., eds. *RDF Schema 1.1*. Accessed: 29.07.2025. (2023). URL: https://www.w3.org/TR/rdf-schema/.

[103] W3C OWL Working Group, ed. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Accessed: 29.07.2025. (2017). URL: https://www.w3.org/TR/owl2-overview/.

[104] T. Berners-Lee et al. *Uniform Resource Identifier (URI): Generic Syntax. RC 3986*. Accessed: 29.07.2025. (2005). URL: https://datatracker.ietf.org/doc/html/rfc3986.

[105] P. Pauwels et al. "EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology". In: *Automation in Construction* 63 (2016), pp. 100–133. ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2015.12.003.

[106] D. Beckett et al. *Turtle - Terse RDF Triple Language*. Accessed: 29.07.2025. (2011). URL: https://www.w3.org/TeamSubmission/turtle/.

[107] T. Berners-Lee et al. *Notation3 (N3): A readable RDF syntax*. Accessed: 29.07.2025. (2011). URL: https://www.w3.org/TeamSubmission/n3/.

[108] P. Brůha. *Semantic Web and Classic Data Structures*. (2013). DOI: 10.13140/RG.2.2.18962.35525.

[109] Ontotext. *GraphDB – The Leading RDF Database for Knowledge Graphs*. Accessed: 29.07.2025. (2024). URL: https://www.ontotext.com/products/graphdb/.

[110] R. Barbau et al. "OntoSTEP: Enriching product model data using ontologies". In: *Computer-Aided Design* 44.6 (2012), pp. 575–590. ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2012.01.008.

[111] J. Beetz et al. "IfcOWL: A case of transforming EXPRESS schemas into ontologies". In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23.1 (2009), pp. 89–101. ISSN: 0890-0604. DOI: 10.1017/S0890060409000122.

[112] buildingSMART International. *ifcOWL - buildingSMART International*. Accessed: 29.07.2025. (2025). URL: https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/.

[113] J. Oraskari et al. *IFCtoLBD: IFCtoLBD v 2.44.0*. Version 2.44.0. Accessed: 29.07.2025. (2024). URL: https://github.com/jyrkioraskari/IFCtoLBD.

[114] P. Pauwels et al., eds. *Buildings and Semantics: Data Models and Web Technologies for the Built Environment*. English. CRC Press, (2022). DOI: 10.1201/9781003204381.

[115] M. Bonduel et al. "The IFC to Linked Building Data Converter - Current Status". English. In: *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2018* 2159 (2018). Accessed: 29.07.2025, pp. 34–43. ISSN: 1613-0073. URL: https://ceur-ws.org/Vol-2159/04paper.pdf.

[116] J. Wix et al. *Information Delivery Manual: Guide to Components and Development Method*. Accessed: 29.07.2025. (2010). URL: https://standards.buildingsmart.org/documents/IDM/IDM_guide-CompsAndDevMethods-IDMC_004-v1_2.pdf.

[117] National BIM Standard Project Committee. "National building information modeling standard, version 1, Part 1: Overview, principles, and methodologies". In: *National Institutute of Building Sciences* (2007).

[118] J. Karlshoej. "Process and building information modelling in the construction industry by using information delivery manuals and model view definitions". In: *Proceedings of 9th European Conference on Product and Process Modeling*. (2012), pp. 305–309.

[119] T. Aditya Sai Srinivas et al. "Data Standardization: Key to Effective Data Integration". In: *Advanced Innovations in Computer Programming Languages* 6.1 (2023). DOI: https://zenodo.org/records/10060920.

[120] M. Winfield. "Construction 4.0 and ISO 19650: a panacea for the digital revolution?" In: *Proceedings of the Institution of Civil Engineers - Management, Procurement and Law* 173.4 (2020), pp. 175–181. ISSN: 1751-4304. DOI: 10.1680/jmapl.19.00051.

[121] Deutsches Institut für Normung e.V. (Hrsg.) *DIN EN ISO 19650-1:2019-08, Organisation und Digitalisierung von Informationen zu Bauwerken und Ingenieurleistungen, einschließlich Bauwerksinformationsmodellierung_(BIM)_- Informationsmanagement mit BIM_- Teil_1: Begriffe und Grundsätze (ISO_19650-1:2018); Deutsche Fassung EN_ISO_19650-1:2018*. Berlin, (2019). DOI: 10.31030/3030494.

[122] Deutsches Institut für Normung e.V. (Hrsg.) *DIN EN ISO 19650-2:2019-08, Organisation und Digitalisierung von Informationen zu Bauwerken und Ingenieurleistungen, einschließlich Bauwerksinformationsmodellierung_(BIM)_- Informationsmanagement mit BIM_- Teil_2: Planungs-, Bau- und Inbetriebnahmephase (ISO_19650-2:2018); Deutsche Fassung EN_ISO_19650-2:2018*. Berlin, (2019). DOI: 10.31030/3030497.

[123] buildingSMART International. *Use Case Management*. Accessed: 29.07.2025. (2025). URL: https://ucm.buildingsmart.org/de/use-case-management.

[124] buildingSMART International. *buildingSMART Data Dictionary*. Accessed: 29.07.2025. (2025). URL: https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/.

[125] buildingSMART International. *buildingSMART Data Dictionary (bSDD) - Search*. Accessed: 29.07.2025. (2025). URL: https://search.bsdd.buildingsmart.org/.

[126] C. Zhang et al. "Interoperable validation for IFC building models using open standards". In: *Journal of Information Technology in Construction - Special Issue* 20.ECPPM-2014 (2015). Accessed: 29.07.2025, pp. 24–39. URL: http://www.itcon.org/2015/2.

[127] T. Chipman et al. *mvdXML: Specification of a standardized format to define and exchange Model View Definitions with Exchange Requirements and Validation Rules*. Accessed: 29.07.2025. (2016). URL: https://standards.buildingsmart.org/MVD/RELEASE/mvdXML/v1-1/mvdXML_V1-1-Final.pdf.

[128] buildingSMART International. *Index of /MVD/RELEASE/mvdXML/v1-1*. Accessed: 29.07.2025. (2021). URL: https://standards.buildingsmart.org/MVD/RELEASE/mvdXML/v1-1/.

[129] M. Lopez et al. "Building a chemical ontology using Methontology and the Ontology Design Environment". In: *IEEE Intelligent Systems and their Applications* 14.1 (1999), pp. 37–46. DOI: 10.1109/5254.747904.

[130] S. Staab et al. *Handbook on ontologies*. 2. ed. International handbooks on information systems. Dordrecht: Springer, (2009). ISBN: 978-3-540-70999-2. DOI: 10.1007/978-3-540-92673-3.

[131] S. Tartir et al. "Ontological Evaluation and Validation". In: *Theory and Applications of Ontology: Computer Applications*. Ed. by R. Poli et al. Dordrecht: Springer Netherlands, (2010), pp. 115–130. ISBN: 978-90-481-8847-5. DOI: 10.1007/978-90-481-8847-5_5.

[132] C. M. Steiner et al. "Validating domain ontologies: A methodology exemplified for concept maps". In: *Cogent Education* 4.1 (2017). Ed. by S. Wang. DOI: 10.1080/2331186X.2016.1263006.

[133] A. Gómez-Pérez. "Ontology Evaluation". In: *Handbook on Ontologies*. Ed. by S. Staab et al. Springer Berlin Heidelberg, (2004), pp. 251–273. ISBN: 978-3-540-24750-0. DOI: 10.1007/978-3-540-24750-0_13.

[134] A. Degbelo. "A Snapshot of Ontology Evaluation Criteria and Strategies". In: *Proceedings of the 13th International Conference on Semantic Systems*. Semantics 2017. Amsterdam, Netherlands: Association for Computing Machinery, New York, USA, (2017), pp. 1–8. ISBN: 9781450352963. DOI: 10.1145/3132218.3132219.

[135] O. Corcho et al. "Methodologies, tools and languages for building ontologies. Where is their meeting point?" In: *Data Knowledge Engineering* 46.1 (2003), pp. 41–64. ISSN: 0169-023X. DOI: https://doi.org/10.1016/S0169-023X(02)00195-7.

[136] M. A. Musen. "The protégé project: a look back and a look forward". In: *AI Matters* 1.4 (2015), pp. 4–12. DOI: 10.1145/2757001.2757003.

[137] S. Chávez-Feria et al. "Chowlk: from UML-Based Ontology Conceptualizations to OWL". In: *The Semantic Web*. Ed. by P. Groth et al. Cham: Springer International Publishing, (2022), pp. 338–352. DOI: https://doi.org/10.1007/978-3-031-06981-9_20.

[138] A. S. Kuhlmann. "Bewertung der Optimierungsmodelle". In: *Konstruktion und Implementierung eines Optimierungsmodells für den Kombinierten Güterverkehr: mit der Fokussierung auf ein Umschlagterminal*. Wiesbaden: Springer Fachmedien Wiesbaden, (2013), pp. 164–178. ISBN: 978-3-658-02473-4. DOI: https://doi.org/10.1007/978-3-658-02473-4_4.

[139] A. Mohammed et al. "Life cycle environmental impact assessment of steel structures using building information modeling". In: *Innovative Infrastructure Solutions* 10.1 (2025). ISSN: 2364-4176. DOI: 10.1007/s41062-024-01840-8.

[140] M. Feldmann et al. "Wiederverwendung im Stahl– und Metallleichtbau". In: *2023 Stahlbau Kalender*. Ed. by U. Kuhlmann. Wiley, (2023), pp. 651–684. ISBN: 9783433033876. DOI: 10.1002/9783433611302.ch8.

[141] European Parliament and Council. *Directive 2008/98/EC on Waste and Repealing Certain Directives*. Accessed: 29.07.2025. (2008). URL: http://data.europa.eu/eli/dir/2008/98/2018-07-05.

[142] A. M. G. Coelho et al. *PROGRESS - Provisions for greater reuse of steel structures: European Recommendations for Reuse of Steel Products in Single-Storey Buildings*. Tech. rep. Accessed:

29.07.2025. (2020). URL: https://www.steelconstruct.com/wp-content/uploads/PROGRESS_Design_guide_final-version.pdf.

[143] The European Convention for Constructional Steelwork. *Case studies of the EU project PROGRESS*. Accessed: 29.07.2025. (2022). URL: https://www.steelconstruct.com/eu-projects/progress/case-studies/.

[144] M. Gorgolewski. "Designing with reused building components: some challenges". In: *Building Research & Information* 36.2 (2008), pp. 175–188. DOI: 10.1080/09613210701559499.

[145] Deutsches Institut für Normung e.V. *DIN 1050 Blatt 2 (1947) Altstahl im Hochbau – Richtlinien für Aufarbeitung und Verwendung*. s.l.: s.n. (1947).

[146] American Institute of Steel Construction (AISC). *Specification for Structural Steel Buildings*. Accessed: 29.07.2025. (2016). URL: https://www.aisc.org/globalassets/aisc/publications/standards/a360-16w-rev-june-2019.pdf.

[147] M. Feldmann et al. "Erarbeitung europäischer Bemessungsvorschriften für wiederverwendete Stahlbauteile". In: *Stahlbau* 93.3 (2024), pp. 192–203. ISSN: 00389145. DOI: 10.1002/stab.202400005.

[148] H. Bartsch et al. "On the development of regulations for the increased reuse of steel structures". In: *Life-Cycle of Structures and Infrastructure Systems*. Ed. by F. Biondini et al. London: CRC Press, (2023), pp. 1287–1294. ISBN: 9781003323020. DOI: 10.1201/9781003323020-158.

[149] Deutsche Institut für Normung e.V. *DIN CEN/TS 17440:2020-10, Bewertung und Ertüchtigung von bestehenden Tragwerken; Deutsche Fassung CEN/TS_17440:2020*. Berlin, (2020). DOI: 10.31030/3110663.

[150] I. Bertin et al. "A BIM-Based Framework and Databank for Reusing Load-Bearing Structural Elements". In: *Sustainability* 12 (2020). DOI: 10.3390/su12083147.

[151] C. Fivet. "Design of load-bearing systems for open-ended downstream reuse". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 225. 1. IOP Publishing. (2019). DOI: 10.1088/1755-1315/225/1/012031.

[152] I. Bertin et al. "Construction, deconstruction, reuse of the structural elements: The circular economy to reach zero carbon". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 323. 1. IOP Publishing. (2019). DOI: 10.1088/1755-1315/323/1/012020.

[153] Deutsche Industrie- und Handelskammer. *DIHK-Report Fachkräfte 2023/2024*. Accessed: 29.07.2025. Berlin, (2023). URL: https://www.dihk.de/resource/blob/107882/f8e2f248f04aaf10e622d5a0fcb38df9/dihk-fachkraeftereport-2023-data.pdf.

[154] K. S. Saidi et al. "Robotics in Construction". In: *Springer Handbook of Robotics*. Ed. by B. Siciliano et al. Springer Handbooks. Cham: Springer International Publishing, (2016), pp. 1493–1520. ISBN: 978-3-319-32550-7. DOI: 10.1007/978-3-319-32552-1_57.

[155] M. Vannucci et al. "Robotic systems in the European steel industry: state-of-art and use cases". In: *Industry 4.0 and the Road to Sustainable Steelmaking in Europe: Recasting the Future*. Springer International Publishing Cham, (2024), pp. 77–96. DOI: 10.1007/978-3-031-35479-3_5.

[156] M. Weinel et al. "Introduction: The Historic Importance and Continued Relevance of Steel-Making in Europe". In: *Industry 4.0 and the Road to Sustainable Steelmaking in Europe*. Ed. by D. Stroud et al. Topics in Mining, Metallurgy and Materials Engineering.

Cham: Springer International Publishing, (2024), pp. 1–14. ISBN: 978-3-031-35478-6. DOI: 10.1007/978-3-031-35479-3_1.

[157] Organisation for Economic Co-operation and Development (OECD). *The Digital Transformation of SMEs*. (2021). DOI: 10.1787/bdb9256a-en.

[158] P. F. Rocha et al. "Impacts of Prefabrication in the Building Construction Industry". In: *Encyclopedia* 3.1 (2023), pp. 28–45. ISSN: 2673-8392. DOI: 10.3390/encyclopedia3010003.

[159] S. Stumm et al. "Robotik im Stahlbau 4.0: Von der digitalen Planung zu Produktion und Bau". In: *Stahlbau Kalender 2019*. John Wiley Sons, Ltd, (2019), pp. 733–778. ISBN: 9783433609873. DOI: https://doi.org/10.1002/9783433609873.ch12.

[160] H. Wilmsmeyer et al. "Intelligente Korrektur eines Schweißroboters". In: *Kommunikation und Bildverarbeitung in der Automation*. Ed. by J. Jasperneite et al. Springer Berlin Heidelberg, (2018), pp. 283–294. ISBN: 978-3-662-55232-2. DOI: https://doi.org/10.1007/978-3-662-55232-2_22.

[161] S. Kumar et al. *Fountain – an intelligent contextual assistant combining knowledge representation and language models for manufacturing risk identification*. (2023). DOI: https://doi.org/10.48550/arXiv.2308.00364.

[162] M. Rickert et al. "Industrieroboter für KMU". In: *Industrie 4.0 Management* 32.2 (2016). Accessed: 29.07.2025, pp. 46–49. URL: https://industry-science.com/artikel/industrieroboter-fuer-kmu-flexible-und-intuitive-prozessbeschreibung/.

[163] A. Perzylo et al. "SMErobotics: Smart Robots for Flexible Manufacturing". In: *IEEE Robotics Automation Magazine* 26.1 (2019), pp. 78–90. DOI: 10.1109/MRA.2018.2879747.

[164] T. Dietz et al. "Programming system for efficient use of industrial robots for deburring in SME environments". In: *ROBOTIK 2012, Proceedings of 7th German Conference on Robotics*. (2012). DOI: 10.24406/publica-fhg-375949.

[165] M. R. Pedersen et al. "Robot skills for manufacturing: From concept to industrial deployment". In: *Robotics and Computer-Integrated Manufacturing* 37 (2016), pp. 282–291. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2015.04.002.

[166] X. Zhao et al. "Development of a Robotic Structural Steel Cutting System". In: *IOP Conference Series: Materials Science and Engineering* 538.1 (2019). DOI: 10.1088/1757-899X/538/1/012042.

[167] J. Hippe. "Entwicklung einer prototypischen Schnittstelle zur Ansteuerung von Industrierobotern über den DSTV-NC". Master thesis. RWTH Aachen University, (2021).

[168] J. Braumann et al. "Parametric robot control". In: *Integrated CAD/CAM for Architectural Design. ACADIA* 11 (2011), pp. 242–251. DOI: 10.52842/conf.acadia.2011.242.

[169] P. Zeman. "Industrie und Stahlbau 4.0 – ein paar Gedanken!" In: *Stahlbau* 86.1 (2017), pp. 84–86. ISSN: 00389145. DOI: 10.1002/stab.201720450.

[170] S. Talebi et al. "Tolerance Management in Construction: A Conceptual Framework". In: *Sustainability* 12.3 (2020). DOI: 10.3390/su12031039.

[171] S. Talebi et al. "Critical Review of Tolerance Management in Construction". In: *24th Annual Conference of the International Group for Lean Construction*. Accessed: 30.07.2025. (2016). URL: http://www.iglc.net/papers/details/1275.

[172] R. Müller et al. "Tolerance Management for Assembly – Not a Matter of Product Size". In: *Precision Assembly Technologies and Systems*. Ed. by S. Ratchev. Vol. 371. IFIP Advances in Information and Communication Technology. Berlin, Heidelberg: Springer Berlin Heidelberg, (2012), pp. 97–104. ISBN: 978-3-642-28162-4. DOI: https://doi.org/10.1007/978-3-642-28163-1_13.

[173] P. Miermeister. "Fraunhofer-Gesellschaft Beherrschung von Toleranzen in der Robotik". In: *Seminar "Zukunft der Industrierobotik" 2013*. Aceesed: 29.07.2025. Stuttgart, (2013). URL: http://publica.fraunhofer.de/dokumente/N-277654.html.

[174] D. Ni et al. "Haptic and Visual Augmented Reality Interface for Programming Welding Robots". In: *Advances in Manufacturing* 5.3 (2017), pp. 191–198. ISSN: 2095-3127. DOI: 10.1007/s40436-017-0184-7.

[175] K. D. Willis et al. "Interactive Fabrication: New Interfaces for Digital Fabrication". In: *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*. New York, NY, USA: ACM, (2011), pp. 69–72. ISBN: 978-1-4503-0478-8. DOI: 10.1145/1935701.1935716.

[176] H. Lee et al. "A Robot Teaching Framework for a Redundant Dual Arm Manipulator with Teleoperation from Exoskeleton Motion Data". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. (2014), pp. 1057–1062. DOI: 10.1109/HUMANOIDS.2014.7041495.

[177] M. R. Pedersen et al. "Gesture-Based Extraction of Robot Skill Parameters for Intuitive Robot Programming". In: *Journal of Intelligent & Robotic Systems* 80 (2015). DOI: https://doi.org/10.1007/s10846-015-0219-x.

[178] C. Yang et al. "Development of a Robotic Teaching Interface for Human to Human Skill Transfer". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (2016), pp. 710–716. DOI: 10.1109/IROS.2016.7759130.

[179] Y. Liu. "Toward Intelligent Welding Robots: Virtualized Welding Based Learning of Human Welder Behaviors". In: *Weld World* 60.4 (2016), pp. 719–729. ISSN: 0043-2288. DOI: 10.1007/s40194-016-0340-x.

[180] H. Guido et al. "A Stateful Robotic Weldment Geometry Measuring System". In: *International Journal of Materials and Product Technology* 48.1-4 (2014), pp. 167–178. ISSN: 0268-1900. DOI: 10.1504/IJMPT.2014.059044.

[181] F. Habibkhah et al. "Application of machine learning for seam profile identification in robotic welding". In: *Machine Learning with Applications* 20 (2025). ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2025.100633.

[182] T. Adams et al. "Praktiken der Vorfertigung im Stahlbau". In: *IoC - Internet of Construction : Informationsnetzwerke zur unternehmensübergreifenden Kollaboration in den Fertigungsketten des Bauwesens*. Ed. by S. Brell-Cokcan et al. Springer Fachmedien Wiesbaden, (2024), pp. 41–76. ISBN: 978-3-658-42544-9. DOI: 10.1007/978-3-658-42544-9_3.

[183] W. Lu et al. "Digital technologies for construction sustainability: Status quo, challenges, and future prospects". In: *npj Materials Sustainability* 2.1 (2024). DOI: 10.1038/s44296-024-00010-2.

[184] M. Abuhussain et al. "Integrating Building Information Modeling (BIM) for optimal lifecycle management of complex structures". In: *Structures* 60 (2024). DOI: 10.1016/j.istruc.2023.105831.

[185] B. Gordon. "Prefabricated Steel Structures: Streamlining Construction Processes". In: *Journal of Steel Structures and Constructions* 10:3 (2024). ISSN: 2472-0437. DOI: 10.37421/2472-0437.2024.10.180.

[186] K. Menzel et al. "Linked Data and Ontologies for Semantic Interoperability". In: *Innovative Tools and Methods Using BIM for an Efficient Renovation in Buildings*. Ed. by B. Daniotti et al. Cham: Springer International Publishing, (2022), pp. 17–28. ISBN: 978-3-031-04670-4. DOI: 10.1007/978-3-031-04670-4_2.

[187] M. Heinisuo et al. "Product modelling and data exchange for constructional steelwork". In: *Proceedings of CIT2000, CIB-W78, IABSE, EG-SEA-AI - Construction Information Technology*. Accessed: 29.07.2025. Reykjavik, Iceland, (2000), pp. 448–456. URL: http://itc.scix.net/paper/w78-2000-448.

[188] D. Brown et al. *Design for Construction*. Ed. by The Steel Construction Institute. Accessed: 29.07.2025. Berkshire, (1997). URL: https://steelconstruction.info/images/2/24/SCI_P178.pdf.

[189] A. J. Crowley et al. *CIMsteel Integration Standards: Release 2: Second Edition*. Ed. by The Steel Construction Institute. Berkshire, (2003).

[190] E. Holtzhauer et al. "Product modelling in the steel construction domain". In: *International Conference on Computing in Civil and Building Engineering: proceedings (ICCCBE) , Weimar, vol. 2004* (2004). DOI: 10.25643/BAUHAUS-UNIVERSITAET.241.

[191] R. Lipman. "Mapping Between the CIMSteel Integration Standards and Industry Foundation Classes Product Models for Structural Steel". In: *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*. Accessed: 29.07.2025. Montréal, Canada, (2006). URL: https://itc.scix.net/pdfs/w78-2006-tf476.pdf.

[192] American Institute of Steel Construction (AISC). *EM11 Model View Documentation*. Accessed: 29.07.2025. (2012). URL: https://www.aisc.org/technical-resources/ifc/.

[193] Y. Zheng et al. "A shared ontology suite for digital construction workflow". In: *Automation in Construction* 132 (2021). ISSN: 09265805. DOI: 10.1016/j.autcon.2021.103930.

[194] M. Sabou et al. "Semantics for Cyber-Physical Systems: A cross-domain perspective". In: *Semantic Web* 11 (2020), pp. 115–124. ISSN: 2210-4968. DOI: 10.3233/SW-190381.

[195] A. Wong et al. "Ontology mapping for the interoperability problem in network management". In: *IEEE Journal on Selected Areas in Communications* 23.10 (2005), pp. 2058–2068. DOI: 10.1109/JSAC.2005.854130.

[196] A. Gyrard et al. "Building IoT-Based Applications for Smart Cities: How Can Ontology Catalogs Help?" In: *IEEE Internet of Things Journal* 5.5 (2018), pp. 3978–3990. DOI: 10.1109/JIOT.2018.2854278.

[197] Q. Cao et al. "Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach". In: *Cybernetics and Systems* 50.2 (2019), pp. 82–96. ISSN: 0196-9722. DOI: 10.1080/01969722.2019.1565118.

[198] P. Pauwels et al. *ifcOWL ontology*. Accessed: 29.07.2025. (2019). URL: `https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL/index.html`.

[199] S. Törmä. "Semantic Linking of Building Information Models". In: *2013 IEEE Seventh International Conference on Semantic Computing*. (2013), pp. 412–419. DOI: `10.1109/ICSC.2013.80`.

[200] A. Wagner et al. "BPO: The Building Product Ontology for assembled products". In: *Proceedings of the 7th Linked Data in Architecture and Construction Workshop - LDAC2019*. Accessed: 29.07.2025. (2019). URL: `https://api.semanticscholar.org/CorpusID:198161048`.

[201] P. Bonsma et al. "Handling huge and complex 3D geometries with Semantic Web technology". In: *IOP Conference Series: Materials Science and Engineering* 364.1 (2018). DOI: `10.1088/1757-899X/364/1/012041`.

[202] M. Bonduel. "A Framework for a Linked Data-based Heritage BIM". Accessed 29.07.2025, Available at `https://lirias.kuleuven.be/3416395&lang=en`. PhD thesis. Faculty of Engineering Technology, KU Leuven, (2021).

[203] N. M. El-Gohary et al. "Domain ontology for processes in infrastructure and construction". In: *Journal of construction engineering and management* 136.7 (2010), pp. 730–744. DOI: `https://doi.org/10.1061/(ASCE)CO.1943-7862.0000178`.

[204] L. Kirner et al. "ioc:process – ein neuer Ansatz einer Bauprozess-Ontologie für die unternehmensübergreifende Kollaboration". In: *IoC - Internet of Construction : Informationsnetzwerke zur unternehmensübergreifenden Kollaboration in den Fertigungsketten des Bauwesens*. Ed. by S. Brell-Cokcan et al. Springer Fachmedien Wiesbaden, (2024), pp. 229–278. ISBN: 978-3-658-42544-9. DOI: `https://doi.org/10.1007/978-3-658-42544-9_9`.

[205] L. Kirner et al. *Internet of Construction Process Ontology (ioc)*. Accessed: 29.07.2025. (2023). URL: `http://w3id.org/ioc`.

[206] P. Hitzler et al. "The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation". In: *Semant. Web* 10.1 (2019), pp. 9–32. ISSN: 1570-0844. DOI: `10.3233/SW-180320`.

[207] A. Haller et al. *Semantic Sensor Network Ontology*. Accessed: 29.07.2025. (2017). URL: `https://www.w3.org/TR/vocab-ssn/`.

[208] E. V. Kalemi et al. "ifcOWL-DfMA a new ontology for the offsite construction domain". In: *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2020*. Accessed: 29.07.2025. (2020). URL: `https://ceur-ws.org/Vol-2636/08paper.pdf`.

[209] J. L. Martinez Lastra et al. "Ontologies for Production Automation". In: *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*. Ed. by T. S. Dillon et al. Berlin Heidelberg: Springer Berlin Heidelberg, (2009), pp. 276–289. ISBN: 978-3-540-89784-2. DOI: `https://doi.org/10.1007/978-3-540-89784-2_11`.

[210] J. Oraskari et al. "A Method to Unify Custom Properties in IFC to Linked Building Data Conversion". In: *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2024*. Accessed: 29.07.2025. (2024). URL: `https://ceur-ws.org/Vol-3824/short2.pdf`.

[211] H. Knitt et al. “Prozessmodellierung für den robotischen Zusammenbau im Stahlbau”. In: *Stahlbau* 94 (2025). ISSN: 0038-9145. DOI: 10.1002/stab.202500007.

[212] buildingSMART International. *mvdXML - buildingSMART International*. Accessed: 29.07.2025. (2025). URL: https://technical.buildingsmart.org/standards/ifc/mvd/mvdxml/.

[213] schema.org. *Schema.org - schemas and supporting software*. Accessed: 29.07.2025. (2025). URL: https://github.com/schemaorg/schemaorg.

[214] FAIRsharing Team. *Quantities, Units, Dimensions and Types Ontology (QUDT)*. FAIRsharing. FAIRsharing record: Quantities, Units, Dimensions and Types. (2015). DOI: 10.25504/FAIRsharing.d3pqw7.

[215] T. Lebo et al. *PROV-O: The PROV Ontology*. Ed. by W3C. Accessed: 29.07.2025. (2017). URL: https://www.w3.org/TR/prov-o/.

[216] M. Poveda-Villalón et al. “OOPS! (Ontology Pitfall Scanner!): An on-line tool for ontology evaluation”. In: *International Journal on Semantic Web and Information Systems* 10.2 (2014). Accessed: 29.07.2025, pp. 7–34. URL: https://oops.linkeddata.es.

[217] E. Kerber et al. “Dynamic WAAM: adaptive processes for equivalent contact surface (ECS) optimization”. In: *Construction Robotics* 7.3–4 (2023), pp. 309–328. ISSN: 2509-8780. DOI: 10.1007/s41693-023-00113-7.

[218] KIT - Institute for Automation and Applied Informatics. *KITModelViewer*. Accessed: 29.07.2025. (2025). URL: https://www.iai.kit.edu/ifc.

[219] buildingSMART International. *openBIM*. Accessed: 29.07.2025. (2025). URL: https://www.buildingsmart.org/about/openbim/.

[220] buildingSMART International. *Clarifying the "openBIM Workflow": A Practical Breakdown*. Accessed: 29.07.2025. (2025). URL: https://www.buildingsmart.org/clarifying-the-openbim-workflow/.

[221] buildingSMART International. *bSI Process*. Accessed: 29.07.2025. (2025). URL: https://www.buildingsmart.org/about/bsi-process/.

[222] C. Davis et al. “Reuse, Remanufacturing and Recycling in the Steel Sector”. In: *Philosophical Transactions of the Royal Society A* 382 (2024). DOI: 10.1098/rsta.2023.0244.

[223] V. Jung. “Enabling Circular Construction: Evaluating ICDD for Effective Deconstruction Processes”. In: *35. Forum Bauinformatik, fbi 2024*. Technische Universität Hamburg, Institut für Digitales und Autonomes Bauen, (2024). DOI: 10.15480/882.13525.

[224] Deutsches Institut für Normung e.V. (Hrsg.) *DIN EN ISO 21597-1:2021-07, Informationscontainer zur Datenübergabe_- Austausch-Spezifikation_- Teil_1: Container (ISO_21597-1:2020);* Berlin, (2021). DOI: 10.31030/3137795.

[225] Deutsches Institut für Normung e.V. (Hrsg.) *DIN EN ISO 21597-2:2021-07, Informationscontainer zur Datenübergabe_- Austausch-Spezifikation_- Teil_2: Dynamische Semantik (ISO_21597-2:2020);* Berlin, (2021). DOI: 10.31030/3192763.

Appendix A

Requirements

The following appendix completes the requirements defined in Chapter 4 which must be displayed in the data models.

TABLE A.1: Requirements steel detailing based on DSTV-NC file

Requirement Set	**Specification**	**Description**
Workpiece	Drawing number	Drawing number of the element
	Part-number	A unique identifier assigned to each component
	Position	Location of a part within an assembly or structure
	Material	Specifies the type of material
	Weightpm	Weight per meter
	Surface	Lateral surface
	Single-part number	Specifies whether part number or position number
Creator	System	System which creates the detailing and NC files
	Release	Version of the system
	Company	Fabricating company
	User	User of the system / creator of the file
Project	Order	Order number
	Orderer	The person/ entity that places order for fabrication
	Object	Related object
Plate	Name	ID of the plate
	Thickness	Dimensions specifying size characteristics
	Width	Dimensions specifying size characteristics
	Length	Dimensions specifying size characteristics
Profile	Name	ID of the profile
	Family	Profile Type
	Length	Dimensions specifying size characteristics
	Sawing length	The length specified for cutting operations
	Height	Dimensions specifying size characteristics
	Flange-height	Dimensions specifying size characteristics
	Flange-thickness	Dimensions specifying size characteristics
	Web-thickness	Dimensions specifying size characteristics
	Radius	Dimensions specifying size characteristics

TABLE A.2: Requirements machine fabrication based on DSTV-NC file-1

Requirement Set	**Specification**	**Description**
Surface Treatment	Top	Top Coat applied to the surface
	Base	Base Coat applied
Tolerances	Min	Minimum allowable value for a given dimension
	Max	Maximum allowable value for a given dimension
Profile Definition	Location	Location of the profile in the assembly or structure
	Radius	Radius of curvature for the profile
Plane Definition	Name	Name of the plane used in defining the geometry
	Point	Reference point on the plane
Bending	Edge angle	Angle between edge of profile and adjacent surfaces
	Vertex Y	Y-coordinate of the edge line on the profile
	Vertex X	X-coordinate of the edge line on the profile
	Edge radius	Radius of curvature on the edge of the profile
Through Hole	Reference	Reference on what side of the profile to fabricate
	Vertex X	X coordinate of the hole feature location
	Vertex Y	Y coordinate of the hole feature location
	Diameter	Diameter of the hole
	Quality	Quality of the hole fabrication (e.g., tolerance, finish)
Threaded Hole	Reference	Reference on what side of the profile to fabricate
	Vertex X	X coordinate of the hole feature location
	Vertex Y	Y coordinate of the hole feature location
	Diameter	Diameter of the hole
	Direction	Direction of the thread
Blind Hole	Reference	Reference on what side of the profile to fabricate
	Vertex X	X coordinate of the hole feature location
	Vertex Y	Y coordinate of the hole feature location
	Diameter	Diameter of the hole
	Depth	Depth of the blind hole
Sink Hole	Reference	Reference on what side of the profile to fabricate
	Vertex X	X coordinate of the hole feature location
	Vertex Y	Y coordinate of the hole feature location
	Diameter	Diameter of the hole
	Depth	Depth of the sink hole
	Angle	Angle of the sink hole (e.g., 90 degrees, conical)

TABLE A.3: Requirements machine fabrication based on DSTV-NC file-2

Requirement Set	**Specification**	**Description**
Sawing	Length	Length of the profile to be cut
	Point	Reference point on the profile for cutting operations
	Vector	Direction vector for the cutting operation
	Mitre	Angle of the cut for mitre or angular cuts
Center Punch	Reference	Reference on what side of the profile to fabricate
	Vertex X	X coordinate of the hole feature location
	Vertex Y	Y coordinate of the hole feature location
	Radius	Radius of the center punch hole
Outer and Inner Contours	Level	Contour level (e.g., top, bottom, middle)
	Reference	Reference on what side of the profile to fabricate
	Location	Location of the contour relative to the profile
	Radius	Radius of the contour curvature
	Chamfer Y	Y value of the chamfer (edge treatment)
	Chamfer Phi	Angle of the chamfer in radians
	Notch X	X coordinate of the notch location
	Notch Y	Y coordinate of the notch location
	Notch R	Radius of the notch (e.g., fillet radius)
	Type	Defines if the contour is inner or outer
Signing	Reference	Reference on what side of the profile to fabricate
	Level	Level where the sign is applied
	Text-height	Height of the text in the sign
	Text	Text content of the sign
	Angle	Angle of the sign with respect to the profile
	Trans	Translation applied (e.g., offset)
	Vertex X	X coordinate of the sign feature location
	Vertex Y	Y coordinate of the sign feature location
Cambering	Vertex X	X coordinate of the cambering for the profile
	Vertex Y	Y coordinate of the cambering for the profile
	Level	Level or height at which the cambering is applied
Punching/ Powder	Level	Level of the punching or powdering operation
	Type	Type of punching or powdering operation
	Reference	Reference on what side of the profile to fabricate
	Vertex X	X coordinate of the feature location
	Vertex Y	Y coordinate of the feature location
	Radius	Radius of the punching or powdering feature

Manufacturing Tolerances for Rolled I- and H-Beams in Structural Steel (DIN EN 10034)							
Profile Height h		Flange Width b		Web Thickness s		Flange Thickness t	
Nominal Size	Tolerance	Nominal Size	Tolerance	Nominal Size	Tolerance	Nominal Size	Tolerance
$h \leq 180$	$+3.0/-2.0$	$b \leq 110$	$+4.0/-1.0$	$s \leq 7$	$+0.7/-0.7$	$t \leq 6.5$	$+1.5/-0.5$
$180 < h \leq 400$	$+4.0/-2.0$	$110 < b \leq 210$	$+4.0/-2.0$	$7 < s \leq 10$	$+1.0/-1.0$	$6.5 < t \leq 10$	$+2.0/-1.0$
$400 < h \leq 700$	$+5.0/-3.0$	$210 < b \leq 325$	$+4.0/-4.0$	$10 < s \leq 20$	$+1.5/-1.5$	$10 < t \leq 20$	$+2.5/-1.5$
$h > 700$	$+5.0/-5.0$	$b > 325$	$+8.0/-5.0$	$20 < s \leq 40$	$+2.0/-2.0$	$20 < t \leq 30$	$+2.5/-2.0$
				$40 < s \leq 60$	$+2.5/-2.5$	$30 < t \leq 40$	$+2.5/-2.5$
				$s > 60$	$+3.0/-3.0$	$40 < t \leq 60$	$+3.0/-3.0$
						$t > 60$	$+4.0/-4.0$

TABLE A.4: Tolerances for rolled I- and H-beams in structural steel according to DIN EN 10034 in [mm].

Process – Tolerances (DIN 1090-2)				
Feature	Parameter	Basic Tolerance	Additional Tolerance Cl.1	Additional Tolerance Cl.2
Bolt Hole Position	Deviation of Centerline	$+2.0/-2.0$	$+2.0/-2.0$	$+1.0/-1.0$
	Deviation of Distance a			
	If $a < 3d$	$+0.0/-0.0$	$+3.0/-0.0$	$+2.0/-0.0$
	If $a \geq 3d$	$+3.0/-3.0$	$+3.0/-3.0$	$+2.0/-2.0$
Hole Group Position	Deviation of a Hole Group	$+2.0/-2.0$	$+2.0/-2.0$	$+1.0/-1.0$
Distance Between Hole Groups	Deviation of Distance c			
	General Case	No Requirement	$+5.0/-5.0$	$+2.0/-2.0$
	Individual Part	No Requirement	$+2.0/-2.0$	$+1.0/-1.0$
Hole Diameter	Mechanical Fastener	$+0.5/-0.5$		
Cutout	Depth	No Requirement	$+3.0/-0.0$	$+2.0/-0.0$
	Length	No Requirement	$+3.0/-0.0$	$+2.0/-0.0$

TABLE A.5: Excerpt of process – tolerances (DIN 1090-2) in [mm]

TABLE A.6: Process feedback -1

Requirement Set	Specification	Description
Drilling	Speed	Rotational speed at which the drill operates.
	Feed rate	Rate at which the drill bit advances into the material.
	Hole type	Specifies the shape of the hole being drilled.
	Coolant consumption	Amount of coolant used during the drilling process.
	Coolant type	Specifies the type of coolant used during drilling.
Milling	Speed	Rotational speed of the milling cutter.
	Feed per tooth	Distance the tool advances per tooth in one revolution.
	Cutting speed	Speed at which milling tool cuts.
	Coolant consumption	Amount of coolant used during the milling process.
	Coolant type	Specifies the type of coolant used during milling.
Punching	Speed	Speed at which the punching tool operates.
	Punching force	Amount of force exerted by the punching tool.
Waterjet Cutting	Water pressure	Pressure at which water is ejected from the nozzle.
	Cutting nozzles	Type of nozzle used in the waterjet cutting process.
	Cutting speed	Speed at which the waterjet cuts.
	Abrasive material	Type of abrasive material.
	Water quality	Purity of the water used in the cutting process.
	Distance	Distance between nozzle and material being cut.
	Cutting direction	Direction of the cutting process.
Laser Processing	Cutting gas	Type of gas used.
	Cutting speed	Speed at which the laser cuts through the material.
	Focal length	Focal length of the laser used in cutting.
	Source strength	Power output of the laser source.
Plasma Cutting	Gas pressure	Pressure of the gas used in plasma cutting.
	Shielding gas	Gas used to protect the plasma arc and prevent oxidation.
	Cutting speed	Speed of the plasma cutter.
	Current	Electric current used in the plasma cutting process.
Bending	Bending force	Force applied to bend the material.
	Bending speed	Speed at which the material is bent.
	Bending position	Position of the material during the bending process.
	Gauge position	Position of the back gauge during bending.
	Clamping force	Force applied to clamp the bending tool.

TABLE A.7: Process Feedback -2

Requirement Set	**Specification**	**Description**
Sawing	Saw blade type	Type of saw blade used in the sawing process.
	Saw blade size	Size of the saw blade used.
	Saw blade teeth	Number of teeth on the saw blade.
	Saw blade speed	Speed at which the saw blade rotates during cutting.
	Feed rate	Rate at which the material moves through the saw.
	Coolant type	Type of coolant used during the sawing process.
	Coolant consumption	Amount of coolant used during the sawing process.
Galvanising	Galvanising method	Method used in the galvanising process.
	Pretreatment	Type of pretreatment applied before galvanising.
	Zinc layer thickness	Thickness of the zinc layer applied.
	Post-treatment	Process applied after galvanising to finish the surface.
MIG / MAG Welding	Current	Electric current used during the welding process.
	Wire feed speed	Speed at which the welding wire is fed into the arc.
	Voltage	Voltage applied during welding.
	Gas type	Type of shielding gas used.
	Gas pressure	Pressure of the shielding gas during welding.
	Wire diameter	Diameter of the welding wire used.
	Welding position	Orientation of the welding process (e.g., flat, horizontal, vertical, overhead).

TABLE A.8: Requirements for deconstruction

Requirement Set	**Specification**	**Description**
Connection Details	Type of connection	Type of connection used (e.g., weld, bolt, rivet).
	Connection strength	Strength or load-bearing capacity of the connection.
	Joint design	The design of the joint (e.g., butt weld, lap joint).
	Material compatibility	Compatibility of the materials used for the connection.
Sequence of Construction / Deconstruction	Assembly order	The order in which components are assembled during construction.
	Disassembly order	The order in which components are to be removed during deconstruction.
Centre of Gravity	Centre of gravity location	The position of the centre of gravity of the steel elements for proper handling during transport.
	Weight distribution	How the weight is distributed in relation to the centre of gravity.
Unique ID	Identification code	Unique code or tag for identifying each steel element.
	Traceability data	Data associated with the element's history, including origin, manufacturing date, and prior use.

TABLE A.9: Regulatory requirements for reuse

Requirement Set	**Specification**	**Description**
Data Availability	Data accessibility	Ensuring critical data is available at key stages
Reuse History	Previous use	Documentation of the component's original structural function, including loads and application context.
	Fabrication and deconstruction records	Information about manufacturing date, fabrication processes, and method of disassembly.
Assessment	Damage history	Record of any incidents, overloads, or material fatigue relevant to structural safety.
	Testing data	Results from non-destructive or destructive tests (e.g., ultrasonic, tensile tests) used to validate integrity.
Compliance Documentation	Certification	Certificates, third-party assessments, or other verification of material and structural performance.
	Standards conformity	Evidence of compliance with relevant standards (e.g., DIN 1050, ANSI/AISC 360-16, prEN 1090).
	Assessment procedures	Procedures prescribed by regulation for evaluating suitability of components based on data availability.

Appendix B

Property Sets

TABLE B.1: Pset_ThreadedHole

Property Set	Property	Description
Pset_ ThreadedHole	Diameter	Planned diameter.
	Measured Diameter	Actual measured diameter of the threaded hole.
	Diameter Tolerance Min	Permitted negative deviation of the actual value from the planned value.
	Diameter Tolerance Max	Permitted positive deviation of the actual value from the planned value.
	Direction	The orientation of the thread (e.g., clockwise, counterclockwise).
	Thread Type	Type of threading used, such as UNC (Unified National Coarse) or M (Metric).
	Gradient	Angle or taper of the thread (in degrees).
	Depth	Depth of the threaded hole.
	DIN Reference	DIN standard defining the tolerance for the threaded hole.

TABLE B.2: Pset_BlindHole

Property Set	Property	Description
Pset_ BlindHole	Diameter	Planned diameter of the blind hole.
	Measured Diameter	Actual measured diameter of the blind hole.
	Diameter Tolerance Min	Permitted negative deviation of the actual value from the planned value.
	Diameter Tolerance Max	Permitted positive deviation of the actual value from the planned value.
	Depth	Depth of the blind hole.
	DIN Reference	DIN standard defining the tolerance for the blind hole.

TABLE B.3: Pset_CountersinkHole

Property Set	**Property**	**Description**
Pset_ Countersink Hole	Diameter	Planned diameter of the countersink hole.
	Measured Diameter	Actual measured diameter of the countersink hole after fabrication.
	Diameter Tolerance Min	Permitted negative deviation of the actual value from the planned value.
	Diameter Tolerance Max	Permitted positive deviation of the actual value from the planned value.
	Depth	Depth of the countersink hole.
	Angle	Angle of the countersink hole's walls.
	DIN Reference	DIN standard defining the tolerance for the countersink hole.

Appendix C

DSTV Ontology

LISTING C.1: DSTV Ontology

```
@prefix : <http://w3id.org/dstv#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#> .
@prefix ioc: <http://w3id.org/ioc#> .
@prefix mod: <https://w3id.org/mod#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dstv: <http://w3id.org/dstv#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema1: <http://schema.org#> .
@base <http://w3id.org/dstv#> .

<http://w3id.org/dstv#> rdf:type owl:Ontology ;
                         <http://purl.org/dc/terms/creator> "Victoria Jung, Lukas Kirner, Jyrki Oraskari, Timur Kuzu, Individualized Production RWTH Aachen" ;
                         <http://purl.org/dc/terms/description> "dstv"@en ;
                         <http://purl.org/dc/terms/issued> "2025-08-04"@en ;
                         <http://purl.org/dc/terms/title> "DSTV:Steel Construction Ontology"@en ;
                         <http://purl.org/vocab/vann/preferredNamespacePrefix> "dstv" ;
                         <http://purl.org/vocab/vann/preferredNamespaceUri> "http://w3id.org/dstv#" ;
                         owl:versionInfo "0.0.3" .

#################################################################
#    Annotation properties
#################################################################

###  http://purl.org/dc/elements/1.1/creator
dc:creator rdf:type owl:AnnotationProperty .

```

```
###  http://purl.org/dc/elements/1.1/title
dc:title rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
<http://purl.org/dc/terms/creator> rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
<http://purl.org/dc/terms/description> rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
<http://purl.org/dc/terms/issued> rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
<http://purl.org/dc/terms/title> rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespacePrefix
<http://purl.org/vocab/vann/preferredNamespacePrefix> rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
<http://purl.org/vocab/vann/preferredNamespaceUri> rdf:type owl:AnnotationProperty .


###  http://www.w3.org/2002/07/owl#versionInfo
owl:versionInfo rdf:type owl:AnnotationProperty .


###  https://w3id.org/mod#createdWith
mod:createdWith rdf:type owl:AnnotationProperty .


#################################################################
#    Object Properties
#################################################################

###  http://w3id.org/dstv#hasAdjustedProperty
dstv:hasAdjustedProperty rdf:type owl:ObjectProperty ;
                         rdfs:domain [ rdf:type owl:Class ;
                                       owl:unionOf ( dstv:Feature
                                                     dstv:MeasurableProperties
                                                     dstv:Process
                                                   )
                                     ] ;
                         rdfs:range dstv:PropertyState ;
                         rdfs:comment "Refers to a property that has been modified or
    recalculated based on measurements or processing results." ;
                         rdfs:label "has adjusted property" .
```

```


###  http://w3id.org/dstv#hasComment
dstv:hasComment rdf:type owl:ObjectProperty ;
                rdfs:domain [ rdf:type owl:Class ;
                              owl:unionOf ( dstv:Feature
                                            dstv:Process
                                          )
                            ] ;
                rdfs:range dstv:Comment ;
                rdfs:comment "Links a feature or process to a descriptive comment or annotation." ;
                rdfs:label "has comment" .


###  http://w3id.org/dstv#hasCreator
dstv:hasCreator rdf:type owl:ObjectProperty ;
                rdfs:domain dstv:Workpiece ;
                rdfs:range dstv:Creator ;
                rdfs:comment "Indicates the entity (person or organization) responsible for creating the workpiece." ;
                rdfs:label "has creator" .


###  http://w3id.org/dstv#hasDefinedToleranceProperty
dstv:hasDefinedToleranceProperty rdf:type owl:ObjectProperty ;
                                 rdfs:domain [ rdf:type owl:Class ;
                                               owl:unionOf ( dstv:Feature
                                                             dstv:MeasurableProperties
                                                             dstv:Process
                                                           )
                                             ] ;
                                 rdfs:range dstv:DefinedTolerance ;
                                 rdfs:comment "Connects a feature or process to a defined tolerance specification from a standard or design requirement." ;
                                 rdfs:label "has defined tolerance property" .


###  http://w3id.org/dstv#hasDiameter
dstv:hasDiameter rdf:type owl:ObjectProperty ;
                 rdfs:domain dstv:Feature ;
                 rdfs:range dstv:Diameter ;
                 rdfs:comment "Defines the diameter of a circular feature, such as a hole" ;
                 rdfs:label "has diameter" .


###  http://w3id.org/dstv#hasHeight
dstv:hasHeight rdf:type owl:ObjectProperty ;
               rdfs:domain [ rdf:type owl:Class ;
                             owl:unionOf ( dstv:Feature
                                           dstv:ProfileDef
                                         )
```

```
                           ] ;
               rdfs:range dstv:Height ;
               rdfs:comment "Specifies the height of a feature or a profile." ;
               rdfs:label "has height" .


###  http://w3id.org/dstv#hasInformaiton
dstv:hasInformaiton rdf:type owl:ObjectProperty ;
                    rdfs:domain dstv:Workpiece ;
                    rdfs:range dstv:Information ;
                    rdfs:comment "Links a workpiece to supplementary information." ;
                    rdfs:label "has information" .


###  http://w3id.org/dstv#hasLowerToleranceLimit
dstv:hasLowerToleranceLimit rdf:type owl:ObjectProperty ;
                            rdfs:domain dstv:ToleranceLimit ;
                            rdfs:range dstv:PropertyState ;
                            rdfs:comment "Specifies the lower boundary of an acceptable dimensional range." ;
                            rdfs:label "has lower tolerance limit" .


###  http://w3id.org/dstv#hasMaterial
dstv:hasMaterial rdf:type owl:ObjectProperty ;
                 rdfs:domain dstv:Workpiece ;
                 rdfs:range dstv:Material ;
                 rdfs:comment "Specifies the material from which the workpiece is made." ;
                 rdfs:label "has material" .


###  http://w3id.org/dstv#hasMaterialInventory
dstv:hasMaterialInventory rdf:type owl:ObjectProperty ;
                          rdfs:domain dstv:Material ;
                          rdfs:range dstv:MaterialInventory ;
                          rdfs:comment "Connects a material to its inventory or stock information." ;
                          rdfs:label "has material inventory" .


###  http://w3id.org/dstv#hasMeasuredProperty
dstv:hasMeasuredProperty rdf:type owl:ObjectProperty ;
                         rdfs:domain [ rdf:type owl:Class ;
                                       owl:unionOf ( dstv:Feature
                                                     dstv:MeasurableProperties
                                                     dstv:Process
                                                   )
                                     ] ;
                         rdfs:range dstv:PropertyState ;
                         rdfs:comment "Links a measurable property that was recorded during or after production." ;
                         rdfs:label "has measured property" .
```

```
### http://w3id.org/dstv#hasNominalsize
dstv:hasNominalsize rdf:type owl:ObjectProperty ;
                    rdfs:domain dstv:ToleranceLimit ;
                    rdfs:range dstv:NominalSize ;
                    rdfs:comment "Indicates the target or nominal size from which
    tolerances are derived." ;
                    rdfs:label "has nominalsize" .


### http://w3id.org/dstv#hasNormReference
dstv:hasNormReference rdf:type owl:ObjectProperty ;
                      rdfs:domain dstv:DefinedTolerance ;
                      rdfs:range dstv:NormReference ;
                      rdfs:comment "References a normative document or standard that
    defines the tolerance." ;
                      rdfs:label "has norm reference" .


### http://w3id.org/dstv#hasPlannedProperty
dstv:hasPlannedProperty rdf:type owl:ObjectProperty ;
                        rdfs:domain [ rdf:type owl:Class ;
                                      owl:unionOf ( dstv:Feature
                                                    dstv:MeasurableProperties
                                                    dstv:Process
                                                  )
                                    ] ;
                        rdfs:range dstv:PropertyState ;
                        rdfs:comment "Specifies a property value that is planned
    during the design or production phase." ;
                        rdfs:label "has planned property" .


### http://w3id.org/dstv#hasPosition
dstv:hasPosition rdf:type owl:ObjectProperty ;
                 rdfs:domain [ rdf:type owl:Class ;
                               owl:unionOf ( dstv:Feature
                                             dstv:Process
                                           )
                             ] ;
                 rdfs:range [ rdf:type owl:Class ;
                              owl:unionOf ( dstv:VertexX
                                            dstv:VertexY
                                          )
                            ] ;
                 rdfs:comment "Indicates the position of a feature or process within
    the coordinate system of the workpiece." ;
                 rdfs:label "has position" .


### http://w3id.org/dstv#hasProfileDef
dstv:hasProfileDef rdf:type owl:ObjectProperty ;
```

```
                  rdfs:domain dstv:Workpiece ;
                  rdfs:range dstv:ProfileDef ;
                  rdfs:comment "Connects a workpiece or structural part to its
    profile definition, such as an I-beam or U-profile." ;
                  rdfs:label "has profile def" .


###  http://w3id.org/dstv#hasProject
dstv:hasProject rdf:type owl:ObjectProperty ;
               rdfs:domain dstv:Workpiece ;
               rdfs:range dstv:Project ;
               rdfs:comment "Associates the workpiece with a specific construction or
    production project." ;
               rdfs:label "has project" .


###  http://w3id.org/dstv#hasProperties
dstv:hasProperties rdf:type owl:ObjectProperty ;
                  rdfs:domain [ rdf:type owl:Class ;
                                owl:unionOf ( dstv:Feature
                                              dstv:Process
                                              dstv:ProfileDef
                                            )
                              ] ;
                  rdfs:range [ rdf:type owl:Class ;
                               owl:unionOf ( dstv:ProcessProperties
                                             ifc:ProfileProperties
                                           )
                             ] ;
                  rdfs:comment "Links an object to a set of technical or geometric
    properties relevant for manufacturing or analysis." ;
                  rdfs:label "has properties" .


###  http://w3id.org/dstv#hasReference
dstv:hasReference rdf:type owl:ObjectProperty ;
                 rdfs:domain dstv:Feature ;
                 rdfs:range dstv:DimensionalReference ;
                 rdfs:comment "Defines a reference that a feature is based on." ;
                 rdfs:label "has dimensional reference" .


###  http://w3id.org/dstv#hasToleranceLimit
dstv:hasToleranceLimit rdf:type owl:ObjectProperty ;
                      rdfs:domain dstv:DefinedTolerance ;
                      rdfs:range dstv:ToleranceLimit ;
                      rdfs:comment "Links a tolerance specification to its actual
    dimensional limit values." ;
                      rdfs:label "has tolerance limit" .


###  http://w3id.org/dstv#hasUpperToleranceLimit
dstv:hasUpperToleranceLimit rdf:type owl:ObjectProperty ;
```

```
                            rdfs:domain dstv:ToleranceLimit ;
                            rdfs:range dstv:PropertyState ;
                            rdfs:comment "Specifies the upper boundary of an
    acceptable dimensional range." ;
                            rdfs:label "has upper tolerance limit" .


###  http://w3id.org/dstv#hasView
dstv:hasView rdf:type owl:ObjectProperty ;
             rdfs:domain [ rdf:type owl:Class ;
                           owl:unionOf ( dstv:Feature
                                         dstv:Process
                                       )
                         ] ;
             rdfs:range dstv:Reference ;
             rdfs:comment "Points to the view of the profile where the process and
    feature is planned on." ;
             rdfs:label "has reference view" .


###  http://w3id.org/dstv#hasWidth
dstv:hasWidth rdf:type owl:ObjectProperty ;
              rdfs:domain dstv:Feature ;
              rdfs:range dstv:Width ;
              rdfs:comment "Defines the width of a feature or a profile." ;
              rdfs:label "has width" .


###  http://w3id.org/dstv#hasfeature
dstv:hasfeature rdf:type owl:ObjectProperty ;
                rdfs:domain dstv:Workpiece ;
                rdfs:range dstv:Feature ;
                rdfs:comment "Associates a workpiece with its geometric or functional
    features." ;
                rdfs:label "has feature" .


###  http://w3id.org/dstv#hasfeaturevalues
dstv:hasfeaturevalues rdf:type owl:ObjectProperty ;
                      rdfs:domain dstv:Feature ;
                      rdfs:range dstv:FeatureValues ;
                      rdfs:comment "Points to concrete values that describe the
    feature, such as dimensions or angles." ;
                      rdfs:label "has feature values" .


###  http://w3id.org/dstv#hasgeneralinformation
dstv:hasgeneralinformation rdf:type owl:ObjectProperty ;
                           rdfs:domain dstv:Workpiece ;
                           rdfs:range dstv:GeneralInformation ;
                           rdfs:comment "Links a workpiece to general metadata such as
    date, client, or job description." ;
                           rdfs:label "has general information" .
```

```
### http://w3id.org/dstv#preDefinedType
dstv:preDefinedType rdf:type owl:ObjectProperty ;
                    rdfs:domain dstv:ToleranceLimit ;
                    rdfs:range dstv:ToleranceClasses ;
                    rdfs:comment "Refers to a predefined tolerance class." ;
                    rdfs:label "pre defined type" .


### http://w3id.org/ioc#hasInputElement
ioc:hasInputElement rdf:type owl:ObjectProperty ;
                    rdfs:domain dstv:Process ;
                    rdfs:range dstv:Workpiece ;
                    rdfs:comment "Denotes the input workpiece to a process step." ;
                    rdfs:label "has input element" .


### http://w3id.org/ioc#hasOutputElement
ioc:hasOutputElement rdf:type owl:ObjectProperty ;
                     rdfs:domain dstv:Process ;
                     rdfs:range dstv:Feature ;
                     rdfs:comment "Indicates the result or product of a process,
     typically a new or modified feature." ;
                     rdfs:label "has output element" .


### https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#Identification
ifc:Identification rdf:type owl:ObjectProperty ;
                   rdfs:domain dstv:Workpiece ;
                   rdfs:range ifc:Identifier ;
                   rdfs:comment "Provides a unique identifier for the workpiece in
     compliance with IFC standards." ;
                   rdfs:label "identification" .


#################################################################
#    Data properties
#################################################################

### http://schema.org#Value
schema1:Value rdf:type owl:DatatypeProperty ;
              rdfs:domain dstv:PropertyState ;
              rdfs:range rdfs:Literal ;
              rdfs:comment "Represents the actual value of a planned, measured,
     calculated or defined tolerance property." ;
              rdfs:label "value" .


### http://w3id.org/dstv#AdditionalTolerance1
dstv:AdditionalTolerance1 rdf:type owl:DatatypeProperty ;
                          rdfs:domain dstv:ToleranceClasses ;
                          rdfs:range rdfs:Literal ;
```

```
                          rdfs:comment "Specifies the first additional tolerance value
    applied to the basic tolerance." ;
                          rdfs:label "additional tolerance 1" .


### http://w3id.org/dstv#AdditionalTolerance2
dstv:AdditionalTolerance2 rdf:type owl:DatatypeProperty ;
                          rdfs:domain dstv:ToleranceClasses ;
                          rdfs:range rdfs:Literal ;
                          rdfs:comment "Specifies the second additional tolerance
    value applied to the basic tolerance." ;
                          rdfs:label "additional tolerance 2" .


### http://w3id.org/dstv#BasicTolerance
dstv:BasicTolerance rdf:type owl:DatatypeProperty ;
                    rdfs:domain dstv:ToleranceClasses ;
                    rdfs:range rdfs:Literal ;
                    rdfs:comment "Defines the base tolerance value for a particular
    tolerance class." ;
                    rdfs:label "basic tolerance" .


### http://w3id.org/dstv#Company
dstv:Company rdf:type owl:DatatypeProperty ;
             rdfs:domain dstv:Creator ;
             rdfs:range rdfs:Literal ;
             rdfs:comment "Name of the company or organization that created the
    element." ;
             rdfs:label "company" .


### http://w3id.org/dstv#DrawingNumber
dstv:DrawingNumber rdf:type owl:DatatypeProperty ;
                   rdfs:domain dstv:Information ;
                   rdfs:range rdfs:Literal ;
                   rdfs:comment "Identifies the engineering or technical drawing
    number related to the workpiece." ;
                   rdfs:label "drawing number" .


### http://w3id.org/dstv#MaterialDelivery
dstv:MaterialDelivery rdf:type owl:DatatypeProperty ;
                      rdfs:domain dstv:MaterialInventory ;
                      rdfs:range rdfs:Literal ;
                      rdfs:comment "Indicates the delivery status or date of the
    material." ;
                      rdfs:label "material delivery" .


### http://w3id.org/dstv#MaterialGrade
dstv:MaterialGrade rdf:type owl:DatatypeProperty ;
                   rdfs:domain dstv:Material ;
```

```
                   rdfs:range rdfs:Literal ;
                   rdfs:comment "Specifies the classification or quality grade of the
     material." ;
                   rdfs:label "material grade" .


###  http://w3id.org/dstv#MaterialID
dstv:MaterialID rdf:type owl:DatatypeProperty ;
                rdfs:domain dstv:Material ;
                rdfs:range rdfs:Literal ;
                rdfs:comment "Unique identifier for the material used in the
     workpiece." ;
                rdfs:label "material i d" .


###  http://w3id.org/dstv#MaterialQuality
dstv:MaterialQuality rdf:type owl:DatatypeProperty ;
                     rdfs:domain dstv:Material ;
                     rdfs:range rdfs:Literal ;
                     rdfs:comment "Describes the quality condition or specification of
     the material." ;
                     rdfs:label "material quality" .


###  http://w3id.org/dstv#MaterialStock
dstv:MaterialStock rdf:type owl:DatatypeProperty ;
                   rdfs:domain dstv:MaterialInventory ;
                   rdfs:range rdfs:Literal ;
                   rdfs:comment "Amount of material currently in inventory or stock." ;
                   rdfs:label "material stock" .


###  http://w3id.org/dstv#MaterialSupplier
dstv:MaterialSupplier rdf:type owl:DatatypeProperty ;
                      rdfs:domain dstv:MaterialInventory ;
                      rdfs:range rdfs:Literal ;
                      rdfs:comment "Name or identifier of the supplier providing the
     material." ;
                      rdfs:label "material supplier" .


###  http://w3id.org/dstv#Object
dstv:Object rdf:type owl:DatatypeProperty ;
            rdfs:domain dstv:Project ;
            rdfs:range rdfs:Literal ;
            rdfs:comment "Specifies the construction or production object name related
     to the project." ;
            rdfs:label "object" .


###  http://w3id.org/dstv#Order
dstv:Order rdf:type owl:DatatypeProperty ;
           rdfs:domain dstv:Project ;
```

```
          rdfs:range rdfs:Literal ;
          rdfs:comment "Represents the production or project order identifier." ;
          rdfs:label "order" .


###  http://w3id.org/dstv#OrderNumber
dstv:OrderNumber rdf:type owl:DatatypeProperty ;
                rdfs:domain dstv:Information ;
                rdfs:range rdfs:Literal ;
                rdfs:comment "Unique number used to track or identify the order
    associated with the component." ;
                rdfs:label "order number" .


###  http://w3id.org/dstv#Orderer
dstv:Orderer rdf:type owl:DatatypeProperty ;
            rdfs:domain dstv:Project ;
            rdfs:range rdfs:Literal ;
            rdfs:comment "Entity (person or organization) who placed the order for
    the project." ;
            rdfs:label "orderer" .


###  http://w3id.org/dstv#PartNumber
dstv:PartNumber rdf:type owl:DatatypeProperty ;
               rdfs:domain dstv:Information ;
               rdfs:range rdfs:Literal ;
               rdfs:comment "Unique part number assigned to the workpiece or
    component." ;
               rdfs:label "part number" .


###  http://w3id.org/dstv#PositionNumber
dstv:PositionNumber rdf:type owl:DatatypeProperty ;
                   rdfs:domain dstv:Information ;
                   rdfs:range rdfs:Literal ;
                   rdfs:comment "Identifies the position number of the component
    within the assembly or drawing. Here, the part can be identified as either a main
    part or a sub-part." ;
                   rdfs:label "position number" .


###  http://w3id.org/dstv#Release
dstv:Release rdf:type owl:DatatypeProperty ;
            rdfs:domain dstv:Creator ;
            rdfs:range xsd:dateTimeStamp ;
            rdfs:comment "Specifies the date and time when the detailing of the steel
    element was officially released." ;
            rdfs:label "release" .


###  http://w3id.org/dstv#SupplyCondition
dstv:SupplyCondition rdf:type owl:DatatypeProperty ;
```

```
                    rdfs:domain dstv:Material ;
                    rdfs:range rdfs:Literal ;
                    rdfs:comment "Describes the delivery or supply condition of the
    raw material." ;
                    rdfs:label "supply condition" .


###  http://w3id.org/dstv#System
dstv:System rdf:type owl:DatatypeProperty ;
            rdfs:domain dstv:Creator ;
            rdfs:range rdfs:Literal ;
            rdfs:comment "Refers to the system or application used during creation of
    the component or model." ;
            rdfs:label "system" .


###  http://w3id.org/dstv#User
dstv:User rdf:type owl:DatatypeProperty ;
          rdfs:domain dstv:Creator ;
          rdfs:range rdfs:Literal ;
          rdfs:comment "Identifies the user who created or edited the element." ;
          rdfs:label "user" .


###  http://w3id.org/dstv#Weightpm
dstv:Weightpm rdf:type owl:DatatypeProperty ;
              rdfs:domain dstv:Information ;
              rdfs:range rdfs:Literal ;
              rdfs:comment "Weight per meter of the component or profile." ;
              rdfs:label "weightpm" .


###  http://www.w3.org/ns/prov#GeneratedAtTime
prov:GeneratedAtTime rdf:type owl:DatatypeProperty ;
                    rdfs:domain dstv:PropertyState ;
                    rdfs:range xsd:dateTimeStamp ;
                    rdfs:comment "Timestamp when the property state or measurement
    was generated." ;
                    rdfs:label "generated at time" .


#################################################################
#    Classes
#################################################################

###  http://w3id.org/dstv#AdjustmentProcess
dstv:AdjustmentProcess rdf:type owl:Class ;
                      rdfs:subClassOf dstv:Process ;
                      rdfs:comment "A process in which existing properties are
    corrected or adjusted, e.g. alignment or re-measurement after initial
    fabrication." ;
                      rdfs:label "Adjustment Process" .

```

```
### http://w3id.org/dstv#Assembly
dstv:Assembly rdf:type owl:Class ;
              rdfs:subClassOf dstv:ProductionProcess ;
              rdfs:comment "A production process where multiple components are joined
     or assembled into a final workpiece." ;
              rdfs:label "Assembly" .


### http://w3id.org/dstv#Beam
dstv:Beam rdf:type owl:Class ;
          rdfs:subClassOf dstv:Profile ;
          rdfs:comment "A Beam is usually a horizontal or almost horizontal component
     that can be loaded primarily by bending. For steel construction fabrication, the
     Beam must have a parametric profile and the volume geometry must consist of an
     extrusion. The extrusion direction must correspond to the x-direction of the local
     coordinate system." ;
          rdfs:isDefinedBy ifc:IfcBeam ;
          rdfs:label "Beam" .


### http://w3id.org/dstv#Bending
dstv:Bending rdf:type owl:Class ;
             rdfs:subClassOf dstv:ProductionProcess ;
             rdfs:comment "A process where material is plastically deformed into a
     curved shape." ;
             rdfs:label "Bending" .


### http://w3id.org/dstv#BlindHoleDrill
dstv:BlindHoleDrill rdf:type owl:Class ;
                    rdfs:subClassOf dstv:Drill ;
                    rdfs:comment "A drilling process that creates a hole which does
     not pass completely through the material." ;
                    rdfs:label "Blind Hole Drill" .


### http://w3id.org/dstv#Bolting
dstv:Bolting rdf:type owl:Class ;
             rdfs:subClassOf dstv:Assembly ;
             rdfs:comment "An assembly process using bolts to join components." ;
             rdfs:label "Bolting" .


### http://w3id.org/dstv#BottomReference
dstv:BottomReference rdf:type owl:Class ;
                     rdfs:subClassOf dstv:DimensionalReference ;
                     rdfs:comment "A reference datum positioned at the bottom edge or
     surface of a component." ;
                     rdfs:isDefinedBy <http://w3id.org/dstv#> ;
                     rdfs:label "Bottom Reference" .
```

```
###  http://w3id.org/dstv#BottomView
dstv:BottomView rdf:type owl:Class ;
                rdfs:subClassOf dstv:Reference ;
                rdfs:comment "A view representing the bottom side of a workpiece." ;
                rdfs:label "Bottom View" .


###  http://w3id.org/dstv#Column
dstv:Column rdf:type owl:Class ;
            rdfs:subClassOf dstv:Profile ;
            rdfs:comment "A Column is a vertical structural member which often is aligned with a structural grid intersection. For steel construction fabrication, the Column must have a parametric profile and the volume geometry must consist of an extrusion. The extrusion direction must correspond to the x-direction of the local coordinate system." ;
            rdfs:isDefinedBy ifc:IfcColumn ;
            rdfs:label "Column" .


###  http://w3id.org/dstv#Comment
dstv:Comment rdf:type owl:Class ;
             rdfs:subClassOf dstv:GeneralInformation ;
             rdfs:comment "An annotation or remark associated with a feature, process, or workpiece. Specifications between two entities can be exchanged here" ;
             rdfs:label "Comment" .


###  http://w3id.org/dstv#CountersinkHoleDrill
dstv:CountersinkHoleDrill rdf:type owl:Class ;
                          rdfs:subClassOf dstv:Drill ;
                          rdfs:comment "A drilling process that creates a countersunk hole, typically for flush bolt heads." ;
                          rdfs:label "Countersink Hole Drill" .


###  http://w3id.org/dstv#Creator
dstv:Creator rdf:type owl:Class ;
             rdfs:subClassOf dstv:GeneralInformation ;
             rdfs:comment "Information about the person or organization that produced the workpiece." ;
             rdfs:label "Creator" .


###  http://w3id.org/dstv#Cutting
dstv:Cutting rdf:type owl:Class ;
             rdfs:subClassOf dstv:ProductionProcess ;
             rdfs:comment "A production process where material is removed or separated by cutting." ;
             rdfs:label "Cutting" .


###  http://w3id.org/dstv#DefinedTolerance
dstv:DefinedTolerance rdf:type owl:Class ;
```

```
                      rdfs:comment "A specification of allowable deviation from
     nominal dimensions." ;
                      rdfs:label "Defined Tolerance" .


###  http://w3id.org/dstv#Depth
dstv:Depth rdf:type owl:Class ;
           rdfs:subClassOf dstv:FeatureValues ;
           rdfs:comment "A dimensional value representing how deep a feature extends
     into the material." ;
           rdfs:label "Depth" .


###  http://w3id.org/dstv#Diameter
dstv:Diameter rdf:type owl:Class ;
              rdfs:subClassOf dstv:FeatureValues ;
              rdfs:comment "A dimensional value describing the width across a circular
     feature." ;
              rdfs:label "Diameter" .


###  http://w3id.org/dstv#DimensionalReference
dstv:DimensionalReference rdf:type owl:Class ;
                          rdfs:comment "A reference geometry or datum used to define
     dimensions relative to features." ;
                          rdfs:label "Dimensional Reference" .


###  http://w3id.org/dstv#Drill
dstv:Drill rdf:type owl:Class ;
           rdfs:subClassOf dstv:ProductionProcess ;
           rdfs:comment "A fabrication process to produce holes using a drill tool." ;
           rdfs:label "Drill" .


###  http://w3id.org/dstv#Feature
dstv:Feature rdf:type owl:Class ;
             rdfs:comment "A geometric or functional element of a workpiece which
     represents a steel fabrication output element of a specific process" ;
             rdfs:label "Feature" .


###  http://w3id.org/dstv#FeatureValues
dstv:FeatureValues rdf:type owl:Class ;
                   rdfs:comment "A container class for planned, actual measured or
     nominal geometric dimensions of a feature." ;
                   rdfs:label "Feature Values" .


###  http://w3id.org/dstv#FrontView
dstv:FrontView rdf:type owl:Class ;
               rdfs:subClassOf dstv:Reference ;
               rdfs:comment "A view representing the front side of a workpiece." ;
```

```
                rdfs:label "Front View" .


### http://w3id.org/dstv#GeneralInformation
dstv:GeneralInformation rdf:type owl:Class ;
                        rdfs:comment "A superclass for metadata or descriptive
     information about entities." ;
                        rdfs:label "General Information" .


### http://w3id.org/dstv#Height
dstv:Height rdf:type owl:Class ;
            rdfs:subClassOf dstv:FeatureValues ;
            rdfs:comment "A dimensional value measuring vertical extent of a feature
     or profile" ;
            rdfs:label "Height" .


### http://w3id.org/dstv#Information
dstv:Information rdf:type owl:Class ;
                 rdfs:subClassOf dstv:GeneralInformation ;
                 rdfs:comment "General metadata describing aspects of a workpiece." ;
                 rdfs:label "Information" .


### http://w3id.org/dstv#InternalContouring
dstv:InternalContouring rdf:type owl:Class ;
                        rdfs:subClassOf dstv:Cutting ;
                        rdfs:comment "A cutting process to produce internal shapes or
     voids within a workpiece." ;
                        rdfs:label "Internal Contouring" .


### http://w3id.org/dstv#InternalCountour
dstv:InternalCountour rdf:type owl:Class ;
                      rdfs:subClassOf dstv:VoidingFeature ;
                      rdfs:comment "An internal cutout (creating an opening) or
     external cutout (creating a recess) of arbitrary shape. The edges between cutting
     planes may be overcut or undercut, i.e. rounded." ;
                      rdfs:label "Internal Countour" .


### http://w3id.org/dstv#Material
dstv:Material rdf:type owl:Class ;
              rdfs:subClassOf ifc:Material ;
              rdfs:comment "A class representing physical material of a workpiece." ;
              rdfs:label "Material" .


### http://w3id.org/dstv#MaterialInventory
dstv:MaterialInventory rdf:type owl:Class ;
                       rdfs:subClassOf dstv:GeneralInformation ;
```

```
                       rdfs:comment "Metadata about material stock, delivery, and supplier." ;
                       rdfs:label "Material Inventory" .


###  http://w3id.org/dstv#MeasurableProcessProperties
dstv:MeasurableProcessProperties rdf:type owl:Class ;
                                 rdfs:subClassOf dstv:MeasurableProperties ;
                                 rdfs:comment "Quantifiable attributes of a process, such as feed rate or speed." ;
                                 rdfs:label "Measurable Process Properties" .


###  http://w3id.org/dstv#MeasurableProfileProperties
dstv:MeasurableProfileProperties rdf:type owl:Class ;
                                 rdfs:subClassOf dstv:MeasurableProperties ;
                                 rdfs:comment "Quantifiable properties of a profile or a feature" ;
                                 rdfs:label "Measurable Profile Properties" .


###  http://w3id.org/dstv#MeasurableProperties
dstv:MeasurableProperties rdf:type owl:Class ;
                          rdfs:comment "Superclass grouping measurable features or properties." ;
                          rdfs:label "Measurable Properties" .


###  http://w3id.org/dstv#Milling
dstv:Milling rdf:type owl:Class ;
             rdfs:subClassOf dstv:ProductionProcess ;
             rdfs:comment "A machining process using rotary cutting tools to remove material." ;
             rdfs:label "Milling" .


###  http://w3id.org/dstv#Mitre
dstv:Mitre rdf:type owl:Class ;
           rdfs:subClassOf dstv:ProductionProcess ;
           rdfs:comment "A process producing angled cuts (miter joints) between components." ;
           rdfs:label "Mitre" .


###  http://w3id.org/dstv#NominalSize
dstv:NominalSize rdf:type owl:Class ;
                 rdfs:subClassOf dstv:DefinedTolerance ;
                 rdfs:comment "The target dimension from which tolerances are defined." ;
                 rdfs:label "Nominal Size" .


###  http://w3id.org/dstv#NormReference
```

```
dstv:NormReference rdf:type owl:Class ;
                   rdfs:subClassOf dstv:DefinedTolerance ;
                   rdfs:comment "A reference to a normative standard defining tolerance classes." ;
                   rdfs:label "Norm Reference" .


###  http://w3id.org/dstv#Plate
dstv:Plate rdf:type owl:Class ;
           rdfs:subClassOf dstv:Workpiece ;
           rdfs:comment "An Plate is a planar and often flat part with constant thickness. A plate may carry loads between or beyond points of support, or provide stiffening. The location of the plate (being horizontal, vertical or sloped) is not relevant to its definition. For steel construction fabrication, the Plate must have a  profile and the volume geometry must consist of an extrusion." ;
           rdfs:isDefinedBy ifc:IfcPlate ;
           rdfs:label "Plate" .


###  http://w3id.org/dstv#Process
dstv:Process rdf:type owl:Class ;
             rdfs:comment "The generic class process serves as the central element of the construction process ontology. It can represent a classical process or a subfield common in some definitions such as an activity or a task. Here it is used to describe processes in steel fabrication." ;
             rdfs:isDefinedBy ioc:Process ;
             rdfs:label "Process" .


###  http://w3id.org/dstv#ProcessProperties
dstv:ProcessProperties rdf:type owl:Class ;
                       rdfs:subClassOf dstv:MeasurableProcessProperties ;
                       rdfs:comment "Specific measurable parameters of a process step." ;
                       rdfs:label "Process Properties" .


###  http://w3id.org/dstv#ProductionProcess
dstv:ProductionProcess rdf:type owl:Class ;
                       rdfs:subClassOf dstv:Process ;
                       rdfs:comment "A process directly involved in fabrication or modification of the workpiece." ;
                       rdfs:label "Production Process" .


###  http://w3id.org/dstv#Profile
dstv:Profile rdf:type owl:Class ;
             rdfs:subClassOf dstv:Workpiece ;
             rdfs:comment "A linear structural member with a standardized cross-section." ;
             rdfs:label "Profile" .
```

```
###  http://w3id.org/dstv#ProfileDef
dstv:ProfileDef rdf:type owl:Class ;
                rdfs:comment "Definition of a profile shape according to IFC
     standard." ;
                rdfs:isDefinedBy
     "https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcProfileDef" ;
                rdfs:label "Profile Def" .


###  http://w3id.org/dstv#Project
dstv:Project rdf:type owl:Class ;
             rdfs:subClassOf dstv:GeneralInformation ;
             rdfs:comment "Metadata about the production or construction project." ;
             rdfs:label "Project" .


###  http://w3id.org/dstv#PropertyState
dstv:PropertyState rdf:type owl:Class ;
                   rdfs:comment "Represents an instance of a measurable property at a
     given time or condition." ;
                   rdfs:label "Property State" .


###  http://w3id.org/dstv#Punching
dstv:Punching rdf:type owl:Class ;
              rdfs:subClassOf dstv:ProductionProcess ;
              rdfs:comment "A process removing material using a punch press to create
     holes or shapes." ;
              rdfs:label "Punching" .


###  http://w3id.org/dstv#QualityCheck
dstv:QualityCheck rdf:type owl:Class ;
                  rdfs:subClassOf dstv:Process ;
                  rdfs:comment "A process step intended to assess or validate
     conformity of features." ;
                  rdfs:label "Quality Check" .


###  http://w3id.org/dstv#RearView
dstv:RearView rdf:type owl:Class ;
              rdfs:subClassOf dstv:Reference ;
              rdfs:comment "A view representing the rear side of a workpiece." ;
              rdfs:label "Rear View" .


###  http://w3id.org/dstv#Reference
dstv:Reference rdf:type owl:Class ;
               rdfs:comment "A general viewpoint or reference projection used for
     feature specification." ;
               rdfs:label "Reference View" .
```

```
###  http://w3id.org/dstv#Sawing
dstv:Sawing rdf:type owl:Class ;
            rdfs:subClassOf dstv:ProductionProcess ;
            rdfs:comment "A cutting process using a saw to separate material." ;
            rdfs:label "Sawing" .


###  http://w3id.org/dstv#Scanning
dstv:Scanning rdf:type owl:Class ;
              rdfs:subClassOf dstv:QualityCheck ;
              rdfs:comment "A process where measurement equipment captures feature
    dimensions or geometry." ;
              rdfs:label "Measurement Process" .


###  http://w3id.org/dstv#Signing
dstv:Signing rdf:type owl:Class ;
             rdfs:subClassOf dstv:ProductionProcess ;
             rdfs:comment "The process of marking or labeling components, e.g.
    engraving or stamping identification." ;
             rdfs:label "Signing" .


###  http://w3id.org/dstv#SurfaceFeature
dstv:SurfaceFeature rdf:type owl:Class ;
                    rdfs:subClassOf dstv:Feature ;
                    rdfs:comment "A surface feature is a modification at (onto, or
    into) of the surface of an element. Parts of the surface of the entire surface may
    be affected. The volume and mass of the element may be increased, remain
    unchanged, or be decreased by the surface feature, depending on manufacturing
    technology. However, any increase or decrease of volume is small compared to the
    total volume of the element. In steel fabrication each surface feature is
    representing a fabrication process step which manipulates the surface of the
    workpiece e.g. blasting, painting, marking." ;
                    rdfs:isDefinedBy ifc:IfcSurfaceFeature ;
                    rdfs:label "Surface Feature" .


###  http://w3id.org/dstv#SurfaceTreatment
dstv:SurfaceTreatment rdf:type owl:Class ;
                      rdfs:subClassOf dstv:ProductionProcess ;
                      rdfs:comment "Processes such as painting, galvanizing, or
    coating applied to the surface." ;
                      rdfs:label "Surface Treatment" .


###  http://w3id.org/dstv#SymmetricalReference
dstv:SymmetricalReference rdf:type owl:Class ;
                          rdfs:subClassOf dstv:DimensionalReference ;
                          rdfs:comment "A reference datum centered or mirrored along a
    symmetry plane." ;
                          rdfs:isDefinedBy <http://w3id.org/dstv#> ;
                          rdfs:label "Symmetrical Reference" .
```

```
### http://w3id.org/dstv#ThreadedHoleDrill
dstv:ThreadedHoleDrill rdf:type owl:Class ;
                       rdfs:subClassOf dstv:Drill ;
                       rdfs:comment "A drilling process that creates a threaded hole
    with internal screw threads." ;
                       rdfs:label "Threaded Hole Drill" .


### http://w3id.org/dstv#ThroughHole
dstv:ThroughHole rdf:type owl:Class ;
                 rdfs:subClassOf dstv:VoidingFeature ;
                 rdfs:comment "A circular or slotted or threaded hole, typically but
    not necessarily of smaller dimension than what would be considered a cutout." ;
                 rdfs:isDefinedBy
    "https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#HOLE" ;
                 rdfs:label "Through Hole" .


### http://w3id.org/dstv#ThroughHoleDrill
dstv:ThroughHoleDrill rdf:type owl:Class ;
                      rdfs:subClassOf dstv:Drill ;
                      rdfs:comment "A drilling process that produces a hole through
    the entire workpiece." ;
                      rdfs:label "Through Hole Drill" .


### http://w3id.org/dstv#ToleranceClasses
dstv:ToleranceClasses rdf:type owl:Class ;
                      rdfs:subClassOf dstv:DefinedTolerance ;
                      rdfs:comment "Classes categorizing tolerance ranges" ;
                      rdfs:label "Tolerance Classes" .


### http://w3id.org/dstv#ToleranceLimit
dstv:ToleranceLimit rdf:type owl:Class ;
                    rdfs:subClassOf dstv:DefinedTolerance ;
                    rdfs:comment "A specific upper or lower bound defined for a
    tolerance class." ;
                    rdfs:label "Tolerance Limit" .


### http://w3id.org/dstv#TopReference
dstv:TopReference rdf:type owl:Class ;
                  rdfs:subClassOf dstv:DimensionalReference ;
                  rdfs:comment "A reference datum at the top edge or surface of a
    component." ;
                  rdfs:isDefinedBy <http://w3id.org/dstv#> ;
                  rdfs:label "Top Reference" .


### http://w3id.org/dstv#TopView
```

```
dstv:TopView rdf:type owl:Class ;
            rdfs:subClassOf dstv:Reference ;
            rdfs:comment "A view representing the top side of a workpiece." ;
            rdfs:label "Top View" .


###  http://w3id.org/dstv#ValidationProcess
dstv:ValidationProcess rdf:type owl:Class ;
                      rdfs:subClassOf dstv:QualityCheck ;
                      rdfs:comment "A specific quality check process verifying
    compliance to specifications." ;
                      rdfs:label "Validation Process" .


###  http://w3id.org/dstv#VertexX
dstv:VertexX rdf:type owl:Class ;
            rdfs:subClassOf dstv:FeatureValues ;
            rdfs:comment "The Xcoordinate value of a vertex point defining a feature
    location." ;
            rdfs:label "Vertex X" .


###  http://w3id.org/dstv#VertexY
dstv:VertexY rdf:type owl:Class ;
            rdfs:subClassOf dstv:FeatureValues ;
            rdfs:comment "The Ycoordinate value of a vertex point defining a feature
    location." ;
            rdfs:label "Vertex Y" .


###  http://w3id.org/dstv#VoidingFeature
dstv:VoidingFeature rdf:type owl:Class ;
                   rdfs:subClassOf dstv:Feature ;
                   rdfs:comment "A voiding feature is a modification of an element
    which reduces its volume. Such a feature may be manufactured in different ways,
    for example by cutting, drilling, or milling of members made of various materials,
    or by inlays into the formwork of cast members made of materials such as concrete.
    In steel fabrication each voiding element representation a fabrication process
    step, which significantly reduces the volume e.g. drilling, milling, sawing." ;
                   rdfs:isDefinedBy ifc:IfcVoidingFeature ;
                   rdfs:label "Voiding Feature" .


###  http://w3id.org/dstv#Welding
dstv:Welding rdf:type owl:Class ;
            rdfs:subClassOf dstv:Assembly ;
            rdfs:comment "An assembly process using fusion to join metal components."
    ;
            rdfs:label "Welding" .


###  http://w3id.org/dstv#Width
dstv:Width rdf:type owl:Class ;
```

```
        rdfs:subClassOf dstv:FeatureValues ;
        rdfs:comment "A dimensional value representing how wide a feature or
    profile is." ;
        rdfs:label "Width" .


### http://w3id.org/dstv#Workpiece
dstv:Workpiece rdf:type owl:Class ;
            rdfs:comment "A physical element or component to be manufactured within
    a structural project." ;
            rdfs:isDefinedBy ifc:IfcBuildingElement ;
            rdfs:label "Workpiece" .


### https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#Identifier
ifc:Identifier rdf:type owl:Class ;
            rdfs:subClassOf dstv:GeneralInformation ;
            rdfs:comment "An identifier is an alphanumeric string which allows an
    individual thing to be identified. It may not provide natural-language meaning." ;
            rdfs:isDefinedBy ifc:IfcIdentifier ;
            rdfs:label "Identifier" .


### https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#Material
ifc:Material rdf:type owl:Class ;
           rdfs:subClassOf dstv:GeneralInformation ;
           rdfs:comment "Material is a homogeneous or inhomogeneous substance that
    can be used to form elements (physical products or their components). Material is
    the basic entity for material designation and definition; this includes
    identification by name and classification (via reference to an external
    classification). Material must have a name (=material grade e.g. S355JR) and a
    category (=material e.g. Steel, Aluminium etc.). Material defines the
    planned/proposed material. The finally used material is stored in the property
    set." ;
           rdfs:label "Material" .


### https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#ProfileProperties
ifc:ProfileProperties rdf:type owl:Class ;
                   rdfs:subClassOf dstv:MeasurableProfileProperties ;
                   rdfs:comment "IFC-defined measurable properties associated with
    profile elements." ;
                   rdfs:isDefinedBy
    "https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcProfileProperties"
    ;
                   rdfs:label "Profile Properties" .
```

Appendix D

Modified IFC File

Listing D.1: Modified IFC file in STEP format

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ViewDefinition [DesignTransferView]','ExchangeRequirement [Structural]','Option[Flat wide beams as plates:Off]','Option[Location by:Global]','Option[Locations from Organizer:On]','Option[Layer:Name]','Option[Object coloring:ByObjectClass]','Option[Export All:On]','Option[Export Pours:Off]'),'2;1');
FILE_NAME('IFC4.ifc','2025-03-03T12:43:14',('t.steinhardt'),('Trimble'),'IFC Export Version Nov  4 2024','Tekla Structures 2023 Service Pack 12','IFC4 test model');
FILE_SCHEMA(('IFC4'));
ENDSEC;
DATA;
#1=IFCPERSON('t.steinhardt',$,$,$,$,$,$,$);
#2=IFCORGANIZATION($,'Trimble',$,$,$);
#3=IFCPERSONANDORGANIZATION(#1,#2,$);
#4=IFCAPPLICATION(#2,'2023 Service Pack 12','Tekla Structures','Multi material modeling');
#5=IFCOWNERHISTORY(#3,#4,$,.NOCHANGE.,$,$,$,1740999600);
#6=IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#7=IFCMEASUREWITHUNIT(IFCRATIOMEASURE(304.8),#6);
#8=IFCDIMENSIONALEXPONENTS(1,0,0,0,0,0,0);
#9=IFCCONVERSIONBASEDUNIT(#8,.LENGTHUNIT.,'FOOT',#7);
#10=IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#11=IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.09290304),#10);
#12=IFCDIMENSIONALEXPONENTS(2,0,0,0,0,0,0);
#13=IFCCONVERSIONBASEDUNIT(#12,.AREAUNIT.,'SQUARE FOOT',#11);
#14=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#15=IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.028316846592),#14);
#16=IFCDIMENSIONALEXPONENTS(3,0,0,0,0,0,0);
#17=IFCCONVERSIONBASEDUNIT(#16,.VOLUMEUNIT.,'CUBIC FOOT',#15);
#18=IFCSIUNIT(*,.MASSUNIT.,.KILO.,.GRAM.);
#19=IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
#20=IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
#21=IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.0174532925199433),#20);
#22=IFCDIMENSIONALEXPONENTS(0,0,0,0,0,0,0);
#23=IFCCONVERSIONBASEDUNIT(#22,.PLANEANGLEUNIT.,'DEGREE',#21);
#24=IFCSIUNIT(*,.SOLIDANGLEUNIT.,$,.STERADIAN.);
#25=IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,.DEGREE_CELSIUS.);
#26=IFCSIUNIT(*,.LUMINOUSINTENSITYUNIT.,$,.LUMEN.);
#27=IFCUNITASSIGNMENT((#6,#10,#14,#18,#19,#20,#24,#25,#26));
```

```
#28=IFCCARTESIANPOINT((0.,0.,0.));
#29=IFCDIRECTION((1.,0.,0.));
#30=IFCDIRECTION((0.,0.,1.));
#31=IFCAXIS2PLACEMENT3D(#28,#30,#29);
#32=IFCDIRECTION((0.,1.));
#33=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,0.2,#31,#32);
#34=IFCPROJECT('2W0vjfNnX7y9Ku3WUhLDah',#5,'Unknown',$,$,$,$,(#33),#27);
#35=IFCLOCALPLACEMENT($,#31);
#36=IFCSITE('2iqpZRqeXELxFIjfQF2AfA',#5,'Unknown',$,$,#35,$,$,.ELEMENT.,$,$,$,$,$);
#37=IFCRELAGGREGATES('1z4tL7Tf16VOEUAxe_XGl3',#5,$,$,#34,(#36));
#38=IFCLOCALPLACEMENT(#35,#31);
#39=IFCBUILDING('2eHqvxMDP6I8Bu9QZ2Ya8s',#5,'Unknown',$,$,#38,$,$,.ELEMENT.,$,$,$);
#40=IFCRELAGGREGATES('0Kmaw6zun3reSrVQ1zMiIW',#5,$,$,#36,(#39));
#41=IFCCARTESIANPOINT((0.,-2000.));
#42=IFCCARTESIANPOINT((0.,18000.));
#43=IFCPOLYLINE((#41,#42));
#44=IFCCARTESIANPOINT((6000.,-2000.));
#45=IFCCARTESIANPOINT((6000.,18000.));
#46=IFCPOLYLINE((#44,#45));
#47=IFCCARTESIANPOINT((12000.,-2000.));
#48=IFCCARTESIANPOINT((12000.,18000.));
#49=IFCPOLYLINE((#47,#48));
#50=IFCCARTESIANPOINT((18000.,-2000.));
#51=IFCCARTESIANPOINT((18000.,18000.));
#52=IFCPOLYLINE((#50,#51));
#53=IFCCARTESIANPOINT((24000.,-2000.));
#54=IFCCARTESIANPOINT((24000.,18000.));
#55=IFCPOLYLINE((#53,#54));
#56=IFCCARTESIANPOINT((-2000.,0.));
#57=IFCCARTESIANPOINT((26000.,0.));
#58=IFCPOLYLINE((#56,#57));
#59=IFCCARTESIANPOINT((-2000.,8000.));
#60=IFCCARTESIANPOINT((26000.,8000.));
#61=IFCPOLYLINE((#59,#60));
#62=IFCCARTESIANPOINT((-2000.,16000.));
#63=IFCCARTESIANPOINT((26000.,16000.));
#64=IFCPOLYLINE((#62,#63));
#65=IFCGRIDAXIS('1',#43,.T.);
#66=IFCGRIDAXIS('2',#46,.T.);
#67=IFCGRIDAXIS('3',#49,.T.);
#68=IFCGRIDAXIS('4',#52,.T.);
#69=IFCGRIDAXIS('5',#55,.T.);
#70=IFCGRIDAXIS('A',#58,.T.);
#71=IFCGRIDAXIS('B',#61,.T.);
#72=IFCGRIDAXIS('C',#64,.T.);
#73=IFCLOCALPLACEMENT(#38,#31);
#74=IFCGEOMETRICCURVESET((#43,#46,#49,#52,#55,#58,#61,#64));
#75=IFCGEOMETRICREPRESENTATIONSUBCONTEXT('FootPrint','Model',*,*,*,*,#33,$,.
    MODEL_VIEW.,$);
#76=IFCSHAPEREPRESENTATION(#75,'FootPrint','GeometricCurveSet',(#74));
#77=IFCPRODUCTDEFINITIONSHAPE($,$,(#76));
#78=IFCGRID('3bfTKVDcf4JPaH3q1TPIye',#5,'\X2\00B1\X0\0.00',$,$
    ,#73,#77,(#65,#66,#67,#68,#69),(#70,#71,#72),$,.RECTANGULAR.);
#79=IFCGRIDAXIS('1',#43,.T.);
#80=IFCGRIDAXIS('2',#46,.T.);
#81=IFCGRIDAXIS('3',#49,.T.);
#82=IFCGRIDAXIS('4',#52,.T.);
#83=IFCGRIDAXIS('5',#55,.T.);
#84=IFCGRIDAXIS('A',#58,.T.);
```

```
#85=IFCGRIDAXIS('B',#61,.T.);
#86=IFCGRIDAXIS('C',#64,.T.);
#87=IFCCARTESIANPOINT((0.,0.,1000.));
#88=IFCAXIS2PLACEMENT3D(#87,#30,#29);
#89=IFCLOCALPLACEMENT(#38,#88);
#90=IFCGEOMETRICCURVESET((#43,#46,#49,#52,#55,#58,#61,#64));
#91=IFCSHAPEREPRESENTATION(#75,'FootPrint','GeometricCurveSet',(#90));
#92=IFCPRODUCTDEFINITIONSHAPE($,$,(#91));
#93=IFCGRID('1uq2COSSj0iRPFN9ACa_9Q',#5,'+1.000',$,$,#89,#92,(#79,#80,#81,#82,#83)
    ,(#84,#85,#86),$,.RECTANGULAR.);
#94=IFCGRIDAXIS('1',#43,.T.);
#95=IFCGRIDAXIS('2',#46,.T.);
#96=IFCGRIDAXIS('3',#49,.T.);
#97=IFCGRIDAXIS('4',#52,.T.);
#98=IFCGRIDAXIS('5',#55,.T.);
#99=IFCGRIDAXIS('A',#58,.T.);
#100=IFCGRIDAXIS('B',#61,.T.);
#101=IFCGRIDAXIS('C',#64,.T.);
#102=IFCCARTESIANPOINT((0.,0.,6000.));
#103=IFCAXIS2PLACEMENT3D(#102,#30,#29);
#104=IFCLOCALPLACEMENT(#38,#103);
#105=IFCGEOMETRICCURVESET((#43,#46,#49,#52,#55,#58,#61,#64));
#106=IFCSHAPEREPRESENTATION(#75,'FootPrint','GeometricCurveSet',(#105));
#107=IFCPRODUCTDEFINITIONSHAPE($,$,(#106));
#108=IFCGRID('1T$Htujeb9$RPbYA$aKLvi',#5,'+6.000',$,$
    ,#104,#107,(#94,#95,#96,#97,#98),(#99,#100,#101),$,.RECTANGULAR.);
#109=IFCGRIDAXIS('1',#43,.T.);
#110=IFCGRIDAXIS('2',#46,.T.);
#111=IFCGRIDAXIS('3',#49,.T.);
#112=IFCGRIDAXIS('4',#52,.T.);
#113=IFCGRIDAXIS('5',#55,.T.);
#114=IFCGRIDAXIS('A',#58,.T.);
#115=IFCGRIDAXIS('B',#61,.T.);
#116=IFCGRIDAXIS('C',#64,.T.);
#117=IFCCARTESIANPOINT((0.,0.,6500.));
#118=IFCAXIS2PLACEMENT3D(#117,#30,#29);
#119=IFCLOCALPLACEMENT(#38,#118);
#120=IFCGEOMETRICCURVESET((#43,#46,#49,#52,#55,#58,#61,#64));
#121=IFCSHAPEREPRESENTATION(#75,'FootPrint','GeometricCurveSet',(#120));
#122=IFCPRODUCTDEFINITIONSHAPE($,$,(#121));
#123=IFCGRID('3gxAzczaXFLh0Q9ieDA8Ro',#5,'+6.500',$,$
    ,#119,#122,(#109,#110,#111,#112,#113),(#114,#115,#116),$,.RECTANGULAR.);
#124=IFCBUILDINGSTOREY('0eRy5T6DjFHRiK9EtiByYn',#5,'Unknown',$,$,#73,$,$,.ELEMENT
    .,0.);
#125=IFCDIRECTION((0.,1.,0.));
#126=IFCAXIS2PLACEMENT3D(#28,#29,#125);
#127=IFCLOCALPLACEMENT(#73,#126);
#128=IFCELEMENTASSEMBLY('1YmjF7Z5T6EA6m7cLK_Nat',#5,'Tr\X2\00E4\X0\ger',$,$,#127,$,
    $,$,.NOTDEFINED.);
#129=IFCELEMENTASSEMBLYTYPE('0cbCqt849EOvlWJDbpT0D_',#5,'Unknown',$,$,$,$,$,$,.
    NOTDEFINED.);
#130=IFCPROPERTYENUMERATEDVALUE('Status',$,(IFCLABEL('NEW')),$);
#131=IFCPROPERTYSINGLEVALUE('Reference',$,IFCIDENTIFIER('1'),$);
#132=IFCPROPERTYSET('1l92z2$DX0gevYGCTxqEaN',#5,'Pset_ElementAssemblyCommon',$
    ,(#130,#131));
#133=IFCLOCALPLACEMENT(#127,#31);
#134=IFCCARTESIANPOINT((0.,0.));
#135=IFCDIRECTION((1.,0.));
#136=IFCAXIS2PLACEMENT2D(#134,#135);
```

```
#137=IFCISHAPEPROFILEDEF(.AREA.,'IPE160',#136,82.,160.,5.,7.400000095,9.,0.,0.);
#138=IFCDIRECTION((-1.,0.,0.));
#139=IFCAXIS2PLACEMENT3D(#28,#138,#30);
#140=IFCDIRECTION((0.,0.,-1.));
#141=IFCCOLOURRGB($,0.356862745098039,0.592156862745098,0.137254901960784);
#142=IFCSURFACESTYLERENDERING(#141,0.,$,$,$,$,IFCNORMALISEDRATIOMEASURE(0.00390625),IFCSPECULAREXPONENT(10.),.NOTDEFINED.);
#143=IFCSURFACESTYLE('S235JR',.POSITIVE.,(#142));
#144=IFCEXTRUDEDAREASOLID(#137,#139,#140,1005.);
#145=IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,*,*,#33,$,.MODEL_VIEW.,$);
#146=IFCSHAPEREPRESENTATION(#145,'Body','SolidModel',(#144));
#147=IFCCARTESIANPOINT((1005.,0.,0.));
#148=IFCPOLYLINE((#28,#147));
#149=IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis','Model',*,*,*,*,#33,$,.MODEL_VIEW.,$);
#150=IFCSHAPEREPRESENTATION(#149,'Axis','Curve3D',(#148));
#151=IFCCARTESIANPOINT((502.5,0.,0.));
#152=IFCGEOMETRICREPRESENTATIONSUBCONTEXT('CoG','Model',*,*,*,*,#33,$,.MODEL_VIEW.,$);
#153=IFCSHAPEREPRESENTATION(#152,'CoG','Point',(#151));
#154=IFCPRODUCTDEFINITIONSHAPE($,$,(#146,#150,#153));
#155=IFCBEAM('2_3cDkCurFORwqHcwd5Suz',#5,'Tr\X2\00E4\X0\ger','IPE160',$,#133,#154,'S235JR',.NOTDEFINED.);
#156=IFCBEAMTYPE('0kf$FCTb1CF8x7OuJrnqP7',#5,'Tr\X2\00E4\X0\ger',$,$,$,$,$,$,$);
#157=IFCMATERIAL('S235JR',$,'STEEL');
#158=IFCMATERIALPROFILE('IPE160',$,#157,#137,$,$);
#159=IFCMATERIALPROFILESET($,$,(#158),$);
#160=IFCMATERIALPROFILESETUSAGE(#159,5,$);
#161=IFCPROPERTYSINGLEVALUE('IsExternal',$,IFCBOOLEAN(.F.),$);
#162=IFCPROPERTYSINGLEVALUE('LoadBearing',$,IFCBOOLEAN(.F.),$);
#163=IFCPROPERTYSINGLEVALUE('Slope',$,IFCPLANEANGLEMEASURE(0.),$);
#164=IFCPROPERTYSINGLEVALUE('Roll',$,IFCPLANEANGLEMEASURE(0.),$);
#165=IFCPROPERTYSET('2r4QIWXZz9ph6Xx1XhkXk3',#5,'Pset_BeamCommon',$,(#130,#161,#162,#163,#164));
#166=IFCQUANTITYLENGTH('Length',$,$,1010.,$);
#167=IFCQUANTITYAREA('CrossSectionArea',$,$,0.002,$);
#168=IFCQUANTITYAREA('OuterSurfaceArea',$,$,0.626,$);
#169=IFCQUANTITYAREA('GrossSurfaceArea',$,$,0.626,$);
#170=IFCQUANTITYAREA('NetSurfaceArea',$,$,0.623,$);
#171=IFCQUANTITYVOLUME('GrossVolume',$,$,0.002,$);
#172=IFCQUANTITYVOLUME('NetVolume',$,$,0.0019,$);
#173=IFCQUANTITYWEIGHT('GrossWeight',$,$,15.9,$);
#174=IFCQUANTITYWEIGHT('NetWeight',$,$,14.8,$);
#175=IFCELEMENTQUANTITY('3gEnhD5hPCPQgR4kCzNhDe',#5,'Qto_BeamBaseQuantities',$,$,(#166,#167,#168,#169,#170,#171,#172,#173,#174));
#176=IFCCARTESIANPOINT((725.,-40.,0.));
#177=IFCAXIS2PLACEMENT3D(#176,#30,#29);
#178=IFCLOCALPLACEMENT(#133,#177);
#179=IFCCARTESIANPOINTLIST2D(((0.,0.),(80.,0.),(80.,80.),(0.,80.)));
#180=IFCINDEXEDPOLYCURVE(#179,(IFCLINEINDEX((1,2,3,4,1))),$);
#181=IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'BL10',#180);
#182=IFCCARTESIANPOINT((0.,0.,5.));
#183=IFCAXIS2PLACEMENT3D(#182,#30,#29);
#184=IFCCOLOURRGB($,0.615686274509804,0.615686274509804,0.662745098039216);
#185=IFCSURFACESTYLERENDERING(#184,0.,$,$,$,$,IFCNORMALISEDRATIOMEASURE(0.00390625),IFCSPECULAREXPONENT(10.),.NOTDEFINED.);
#186=IFCSURFACESTYLE('S235JR',.POSITIVE.,(#185));
#187=IFCEXTRUDEDAREASOLID(#181,#183,#140,10.);
```

```
#188=IFCSHAPEREPRESENTATION(#145,’Body’,’SolidModel’,(#187));
#189=IFCPRODUCTDEFINITIONSHAPE($,$,(#188));
#191=IFCCARTESIANPOINT((200.,-40.,0.));
#192=IFCAXIS2PLACEMENT3D(#191,#30,#29);
#193=IFCLOCALPLACEMENT(#133,#192);
#194=IFCSHAPEREPRESENTATION(#145,’Body’,’SolidModel’,(#187));
#195=IFCPRODUCTDEFINITIONSHAPE($,$,(#194));
#197=IFCCARTESIANPOINT((50.,80.,0.));
#198=IFCAXIS2PLACEMENT3D(#197,#125,#29);
#199=IFCLOCALPLACEMENT(#127,#198);
#200=IFCCOLOURRGB($,0.698039215686274,0.4,0.);
#201=IFCSURFACESTYLERENDERING(#200,0.,$,$,$,$,IFCNORMALISEDRATIOMEASURE(0.00390625)
     ,IFCSPECULAREXPONENT(10.),.NOTDEFINED.);
#202=IFCSURFACESTYLE(’7990’,.POSITIVE.,(#201));
#203=IFCCIRCLEPROFILEDEF(.AREA.,’BOLT’,#136,4.);
#204=IFCAXIS2PLACEMENT3D(#28,#140,#125);
#205=IFCEXTRUDEDAREASOLID(#203,#204,#30,20.);
#206=IFCCOLOURRGB(’Black’,0.,0.,0.);
#207=IFCSURFACESTYLERENDERING(#206,0.,$,$,$,$,IFCNORMALISEDRATIOMEASURE(0.00390625)
     ,IFCSPECULAREXPONENT(10.),.NOTDEFINED.);
#208=IFCSURFACESTYLE($,.POSITIVE.,(#207));
#209=IFCCIRCLEPROFILEDEF(.AREA.,’BOLT’,#136,5.);
#210=IFCEXTRUDEDAREASOLID(#209,#204,#30,7.400000095);
#211=IFCCARTESIANPOINT((7.50555,0.));
#212=IFCCARTESIANPOINT((3.75278,6.5));
#213=IFCCARTESIANPOINT((-3.75278,6.5));
#214=IFCCARTESIANPOINT((-7.50555,0.));
#215=IFCCARTESIANPOINT((-3.75278,-6.5));
#216=IFCCARTESIANPOINT((3.75278,-6.5));
#217=IFCPOLYLINE((#211,#212,#213,#214,#215,#216,#211));
#218=IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#217);
#219=IFCCARTESIANPOINT((0.,0.,5.3));
#220=IFCAXIS2PLACEMENT3D(#219,#140,#125);
#221=IFCEXTRUDEDAREASOLID(#218,#220,#30,5.30000019073486);
#222=IFCCIRCLEPROFILEDEF(.AREA.,’BOLT’,#136,8.);
#223=IFCCARTESIANPOINT((0.,0.,-7.4));
#224=IFCAXIS2PLACEMENT3D(#223,#140,#125);
#225=IFCEXTRUDEDAREASOLID(#222,#224,#30,1.60000002384186);
#226=IFCCARTESIANPOINT((0.,0.,-9.));
#227=IFCAXIS2PLACEMENT3D(#226,#140,#125);
#228=IFCEXTRUDEDAREASOLID(#218,#227,#30,6.44000005722046);
#229=IFCSHAPEREPRESENTATION(#145,’Body’,’SweptSolid’,(#205,#210,#221,#225,#228));
#230=IFCREPRESENTATIONMAP(#31,#229);
#231=IFCCARTESIANPOINT((0.,-20.,0.));
#232=IFCCARTESIANTRANSFORMATIONOPERATOR3D($,$,#231,$,$);
#233=IFCMAPPEDITEM(#230,#232);
#234=IFCCARTESIANPOINT((0.,20.,0.));
#235=IFCCARTESIANTRANSFORMATIONOPERATOR3D($,$,#234,$,$);
#236=IFCMAPPEDITEM(#230,#235);
#237=IFCSHAPEREPRESENTATION(#145,’Body’,’MappedRepresentation’,(#233,#236));
#238=IFCPRODUCTDEFINITIONSHAPE($,$,(#237));
#239=IFCOWNERHISTORY(#3,#4,$,.NOCHANGE.,$,$,$,0);
#240=IFCMECHANICALFASTENER(’0LSt5fqPL6vAyKGVWaNqrD’,#239,’M8*20-24017’,$,$
     ,#199,#238,$,8.,20.,$);
#241=IFCMATERIAL(’7990’,$,’STEEL’);
#242=IFCMECHANICALFASTENERTYPE(’2uq1QX3Cb2EgkyXbM5zbjW’,#5,’7990’,$,$,$,$,$,$,$
     ,8.,20.);
#243=IFCPROPERTYSINGLEVALUE(’ThreadDiameter’,$,IFCPOSITIVELENGTHMEASURE(8.),$);
#244=IFCPROPERTYSINGLEVALUE(’ThreadLength’,$,IFCPOSITIVELENGTHMEASURE(20.),$);
```

```
#245= IFCPROPERTYSINGLEVALUE('WashersCount',$,IFCCOUNTMEASURE(1),$);
#246= IFCPROPERTYSINGLEVALUE('NutsCount',$,IFCCOUNTMEASURE(1),$);
#247= IFCPROPERTYSINGLEVALUE('HeadShape',$,IFCLABEL('Hexagon'),$);
#248= IFCPROPERTYSINGLEVALUE('KeyShape',$,IFCLABEL('Slot'),$);
#249= IFCPROPERTYSINGLEVALUE('NutShape',$,IFCLABEL('Hexagon'),$);
#250= IFCPROPERTYSINGLEVALUE('WasherShape',$,IFCLABEL('Standard'),$);
#251= IFCPROPERTYSET('1lz1h2$VH0qe$m_BMO_RdN',#5,'Pset_MechanicalFastenerBolt',$
      ,(#243,#244,#245,#246,#247,#248,#249,#250));
#252= IFCCARTESIANPOINT((955.,80.,0.));
#253= IFCAXIS2PLACEMENT3D(#252,#125,#138);
#254= IFCLOCALPLACEMENT(#127,#253);
#255= IFCSHAPEREPRESENTATION(#145,'Body','MappedRepresentation',(#233,#236));
#256= IFCPRODUCTDEFINITIONSHAPE($,$,(#255));
#257= IFCMECHANICALFASTENER('1Lvn8htH116gRITQhEY0kF',#239,'M8*20-24017',$,$
      ,#254,#256,$,8.,20.,$);
#258= IFCPROPERTYSINGLEVALUE('ThreadDiameter',$,IFCPOSITIVELENGTHMEASURE(8.),$);
#259= IFCPROPERTYSINGLEVALUE('ThreadLength',$,IFCPOSITIVELENGTHMEASURE(20.),$);
#260= IFCPROPERTYSET('24cKIPu7z6rRaQa$BnY2wm',#5,'Pset_MechanicalFastenerBolt',$
      ,(#258,#259,#245,#246,#247,#248,#249,#250));
#261= IFCRELAGGREGATES('1j$6iz7pb9E8r2ia3iPsMc',#5,$,$,#128,(#155,#240,#257));
#262= IFCRELCONTAINEDINSPATIALSTRUCTURE('14A3rIDXL639XuQR7Afst9',#5,$,$,(#128),#124)
      ;
#263= IFCRELAGGREGATES('0b63aadffF6w1lhuhlr_pR',#5,$,$,#39,(#124));
#264= IFCRELCONTAINEDINSPATIALSTRUCTURE('0rvLvwk510_8ybaLEgeB8e',#5,$,$
      ,(#78,#93,#108,#123),#39);
#265= IFCRELASSOCIATESMATERIAL('0nloRQApz3lRJUEzL4YIqc',#5,$,$,(#155),#160);
#266= IFCRELASSOCIATESMATERIAL('2oIu4dv9j70xnpbPK6XLWC',#5,$,$,(#240,#257),#241);
#267= IFCRELDEFINESBYPROPERTIES('06SnWGLc9CER2jseatij1S',#5,$,$,(#128),#132);
#268= IFCRELDEFINESBYPROPERTIES('1R5fKtTen82f2zJx9M_Kfy',#5,$,$,(#155),#165);
#269= IFCRELDEFINESBYPROPERTIES('1v5hDH19nF2R9XMJ3rPRwj',#5,$,$,(#155),#175);
#270= IFCRELDEFINESBYPROPERTIES('24Xm3kWYf2Hg5QzYygRazs',#5,$,$,(#240),#251);
#271= IFCRELDEFINESBYPROPERTIES('3hVPfYhbnCr8DiGS3zpvCT',#5,$,$,(#257),#260);
#272= IFCRELDEFINESBYTYPE('2XhV6oW7PEnwekwIsWPnZa',#5,$,$,(#128),#129);
#273= IFCRELDEFINESBYTYPE('12mwlIcXv7T8e$Bjf0GxCJ',#5,$,$,(#155),#156);
#274= IFCRELDEFINESBYTYPE('2BINHSYVL0o8gRcZaktXVA',#5,$,$,(#240,#257),#242);
#277= IFCSTYLEDITEM(#144,(#143),$);
#278= IFCSTYLEDITEM(#187,(#186),$);
#279= IFCSTYLEDITEM(#205,(#202),$);
#280= IFCSTYLEDITEM(#210,(#208),$);
#281= IFCSTYLEDITEM(#221,(#202),$);
#282= IFCSTYLEDITEM(#225,(#202),$);
#283= IFCSTYLEDITEM(#228,(#202),$);
#284= IFCPRESENTATIONLAYERASSIGNMENT('Grid +1.000',$,(#91),$);
#285= IFCPRESENTATIONLAYERASSIGNMENT('Grid +6.000',$,(#106),$);
#286= IFCPRESENTATIONLAYERASSIGNMENT('Grid +6.500',$,(#121),$);
#287= IFCPRESENTATIONLAYERASSIGNMENT('Grid \X2\00B1\X0\0.00',$,(#76),$);
#288= IFCPRESENTATIONLAYERASSIGNMENT('M8*20-24017',$,(#237,#255),$);
#289= IFCPRESENTATIONLAYERASSIGNMENT('Tr\X2\00E4\X0\ger',$,(#146),$);
#290= IFCVOIDINGFEATURE('3W2O2mdAL4dwQkU01hLgFg',$,'Cutout_01','BL10',$,#178,#189,$
      ,.CUTOUT.);
#291= IFCRELVOIDSELEMENT('1wEpl0VQnA08H0ZSJ7s5h3',$,$,$,#155,#290);
#292= IFCPROPERTYSET($,$,'Pset_Cutout',$
      ,(#293,#294,#295,#296,#297,#298,#299,#300,#301,#302));
#293= IFCPROPERTYSINGLEVALUE('Width',$,IFCPOSITIVELENGTHMEASURE(80.),$);
#294= IFCPROPERTYSINGLEVALUE('MeasuredWidth',$,IFCPOSITIVELENGTHMEASURE(80.3),$);
#295= IFCPROPERTYSINGLEVALUE('WidthToleranceMin',$,IFCLENGTHMEASURE(0.),$);
#296= IFCPROPERTYSINGLEVALUE('WidthToleranceMax',$,IFCLENGTHMEASURE(3.),$);
#297= IFCPROPERTYSINGLEVALUE('Height',$,IFCPOSITIVELENGTHMEASURE(80.),$);
#298= IFCPROPERTYSINGLEVALUE('MeasuredHeight',$,IFCPOSITIVELENGTHMEASURE(80.5),$);
```

```
#299=IFCPROPERTYSINGLEVALUE('HeightToleranceMin',$,IFCLENGTHMEASURE(0.),$);
#300=IFCPROPERTYSINGLEVALUE('HeightToleranceMax',$,IFCLENGTHMEASURE(3.),$);
#301=IFCPROPERTYSINGLEVALUE('DINReference',$,IFCLABEL('DIN 1090-2'),$);
#302=IFCPROPERTYSINGLEVALUE('Timestamp',$,IFCTIMESTAMP(1744819110),$);
#303=IFCPROPERTYSET($,$,'Pset_FabricationProcess',$,(#304,#305,#306,#307));
#304=IFCPROPERTYSINGLEVALUE('ProcessRequired',$,IFCLABEL('plasma cutting'),$);
#305=IFCPROPERTYSINGLEVALUE('ProcessForbidden',$,IFCLABEL(''),$);
#306=IFCPROPERTYSINGLEVALUE('Comment',$,IFCLABEL(''),$);
#307=IFCPROPERTYSINGLEVALUE('Timestamp',$,IFCTIMESTAMP(1744819110),$);
#308=IFCRELDEFINESBYPROPERTIES($,$,$,$,(#290),#292);
#309=IFCRELDEFINESBYPROPERTIES($,$,$,$,(#290),#303);
#310=IFCVOIDINGFEATURE('12vELqmArEPvOFlVWMHPLJ',$,'Cutout_02','BL10',$,#193,#195,$
    ,.CUTOUT.);
#311=IFCRELVOIDSELEMENT('30SRco9012NfDOGg_qu$$f',$,$,$,#155,#310);
#312=IFCPROPERTYSINGLEVALUE('Diameter',$,IFCREAL(10.),$);
#313=IFCPROPERTYSINGLEVALUE('MeasuredDiameter',$,IFCREAL(10.),$);
#314=IFCPROPERTYSINGLEVALUE('Diameter_tol_min',$,IFCREAL(-0.05),$);
#315=IFCPROPERTYSINGLEVALUE('Diameter_tol_max',$,IFCREAL(0.05),$);
#316=IFCPROPERTYSINGLEVALUE('ReferenceNorm',$,IFCLABEL('DIN 1090-2'),$);
#317=IFCPROPERTYSET('38GN4VCCn5gRx$cNtRMkZ3',#5,'Pset_ThroughHole',$
    ,(#312,#313,#314,#315,#316));
#318=IFCVOIDINGFEATURE('3zuVQQahX738TS_yEpbgK5',#5,'Hole_01','Voiding feature for
    fastener',$,#199,$,$,.HOLE.);
#319=IFCRELVOIDSELEMENT('3PyYfuNDT0agxJ36WihgVP',#5,'Hole_01 Relation','Voiding
    feature applied to fastener',#240,#318);
#320=IFCRELDEFINESBYPROPERTIES('1w3NiB1f57L9ucAVLzfurJ',#5,$,$,(#318),#317);
#321=IFCVOIDINGFEATURE('2MwWi_aVv72Qj10t2lI6IQ',#5,'Hole_02','Voiding feature for
    fastener',$,#254,$,$,.HOLE.);
#322=IFCRELVOIDSELEMENT('3phQnW$2L0QASel0fbWEQG',#5,'Hole_02 Relation','Voiding
    feature applied to fastener',#257,#321);
#323=IFCRELDEFINESBYPROPERTIES('0Ja96ttQz43xg_z4JUNbSM',#5,$,$,(#321),#317);
#324=IFCPROPERTYSINGLEVALUE('MeasuredHeight',$,IFCPOSITIVELENGTHMEASURE(162.1),$);
#325=IFCPROPERTYSINGLEVALUE('MeasuredWidth',$,IFCPOSITIVELENGTHMEASURE(81.4),$);
#326=IFCPROPERTYSINGLEVALUE('MeasuredLength',$,IFCPOSITIVELENGTHMEASURE(1002.),$);
#327=IFCPROPERTYSINGLEVALUE('HeightToleranceMin',$,IFCLENGTHMEASURE(-2.),$);
#328=IFCPROPERTYSINGLEVALUE('HeightToleranceMax',$,IFCLENGTHMEASURE(3.),$);
#329=IFCPROPERTYSINGLEVALUE('WidthToleranceMin',$,IFCLENGTHMEASURE(-1.),$);
#330=IFCPROPERTYSINGLEVALUE('WidthToleranceMax',$,IFCLENGTHMEASURE(4.),$);
#331=IFCPROPERTYSINGLEVALUE('DINReference',$,IFCLABEL('DIN EN 10034'),$);
#332=IFCRELDEFINESBYPROPERTIES('2YD9MmrX9APuZOJz4$0IXn',$,'Defines additional
    profile properties',$,(#155),#333);
#333=IFCPROPERTYSET('0xujQhH$XESxT1o46C3q6A',$,'AdditionalProfileProperties',$
    ,(#324,#325,#326,#327,#328,#329,#330,#331));
#334=IFCTASK('2tjlfR5X13787P9KKA3lDV',$,'Plasma Cutting Task',$,$,$,$,'NOTSTARTED',
    $,$,$,#335,$);
#335=IFCTASKTIME($,$,$,.WORKTIME.,'00:05:00','2025-03-27T18:30:00','2025-03-27T18
    :35:00',$,$,$,$,$,$,.F.,$,$,'2025-03-27T18:40:00','2025-03-27T18:43:00',$,$);
#336=IFCPROPERTYSET('09VmwTHmz8iwpJsZOVqFe5',$,'Pset_PlasmaCuttingProcess',$
    ,(#337,#338));
#337=IFCPROPERTYSINGLEVALUE('RobotCommand',$,$,$);
#338=IFCPROPERTYSINGLEVALUE('RobotSpeed',$,IFCREAL(4.75),$);
#339=IFCRELDEFINESBYPROPERTIES('3S0DpY579BrencSAMz4xP_',$,$,$,(#334),#336);
#340=IFCRELASSIGNSTOPROCESS('12fb_gpn1C5g7hPhEMvC2t',$,$,$,(#290),$,#334,$);
#341=IFCPROPERTYSET('0YPZ5itMD2$gzZdxDg2Lb$',$,'Pset_ManufacturerTypeInformation',$
    ,(#342,#343,#344));
#342=IFCPROPERTYSINGLEVALUE('ArticleNumber',$,IFCIDENTIFIER('707048'),$);
#343=IFCPROPERTYSINGLEVALUE('Manufacturer',$,IFCLABEL('KUKA'),$);
#344=IFCPROPERTYSINGLEVALUE('ModelReference',$,IFCLABEL('KR30-3 F'),$);
#345=IFCRELDEFINESBYPROPERTIES('27RftJOGv4W8WCy1cppu9g',$,$,$,(#334),#341);
```

```
#346=IFCTASK(’2Kex6Iogj5ov3af62gdEr7’,$,’Drilling Task’,$,$,$,$,’NOTSTARTED’,$,$,$
    ,#347,$);
#347=IFCTASKTIME($,$,$,.WORKTIME.,’00:30:00’,’2025-04-01T17:00:00’,’2025-04-01T17
    :30:00’,$,$,$,$,$,$,.F.,$,$,’2025-04-01T17:57:00’,’2025-04-01T18:25:00’,$,$);
#348=IFCPROPERTYSET(’0wJrkWrQ5EVxtkzZkYxGuf’,$,’Pset_DrillingProcess’,$,(#349,#350)
    );
#349=IFCPROPERTYSINGLEVALUE(’SpindleSpeed’,$,IFCREAL(800.),$);
#350=IFCPROPERTYSINGLEVALUE(’FeedRate’,$,IFCREAL(0.05),$);
#351=IFCRELDEFINESBYPROPERTIES(’0u2pjardv9W8PZlWFvQnpB’,$,$,$,(#321),#348);
#352=IFCRELDEFINESBYPROPERTIES(’3CIjVJBhH5UelYUhV5MRNl’,$,$,$,(#346),#348);
#353=IFCRELASSIGNSTOPROCESS(’2SNqWJK$zBO9O1pvX$GDZz’,$,$,$,(#321),$,#346,$);
#354=IFCPROPERTYSET(’2uddVRp6P9zP2ygPfaKGFf’,$,’Pset_FabricationProcess’,$
    ,(#355,#356,#357,#358));
#355=IFCPROPERTYSINGLEVALUE(’Position’,$,IFCLABEL(’x=955.0, y=21.0’),$);
#356=IFCPROPERTYSINGLEVALUE(’Process Required’,$,IFCLABEL(’drilling’),$);
#357=IFCPROPERTYSINGLEVALUE(’Process Forbidden’,$,IFCLABEL(’’),$);
#358=IFCPROPERTYSINGLEVALUE(’Comment’,$,IFCLABEL(’’),$);
#359=IFCRELDEFINESBYPROPERTIES(’0FS$V2NgbA3vAaCzyu1ict’,$,$,$,(#321),#354);
#360=IFCRELDEFINESBYPROPERTIES(’2dClZHtnT7IhaLqmkmhEif’,$,$,$,(#346),#341);
ENDSEC;
END-ISO-10303-21;
```

Appendix E

IFC Modification Codes

This appendix contains some of the relevant codes used to modify the IFC file according to the requirements and use case demonstrated in this thesis. It contains the code in addition to the listings in Chapter 7.

LISTING E.1: ifcOpenShell python code for changing *IfcOpeningElement* to *IfcVoding Feature* and adding additional property set

```
ifc_file = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\
    IFC4_Sample Beam.ifc")


opening_elements = ifc_file.by_type("IfcOpeningElement")

if not opening_elements:
    print("Keine IfcOpeningElement-Instanzen gefunden!")
else:
    for index, opening in enumerate(opening_elements, start=1):

        original_guid = opening.GlobalId


        voiding_feature = ifc_file.create_entity("IfcVoidingFeature")
        voiding_feature.GlobalId = original_guid
        voiding_feature.PredefinedType = "CUTOUT"


        voiding_feature.Name = f"Cutout_{index:02}"
        voiding_feature.Description = opening.Description
        voiding_feature.ObjectPlacement = opening.ObjectPlacement
        voiding_feature.Representation = opening.Representation

        relating_element = None
        for rel in ifc_file.by_type("IfcRelVoidsElement"):
            if rel.RelatedOpeningElement == opening:
                relating_element = rel.RelatingBuildingElement
                break

            for rel in ifc_file.by_type("IfcRelVoidsElement"):
                if rel.RelatedOpeningElement == opening:
                    ifc_file.remove(rel)

            voids_relation = ifc_file.create_entity("IfcRelVoidsElement")
            voids_relation.GlobalId = ifcopenshell.guid.new()
            voids_relation.RelatingBuildingElement = relating_element
            voids_relation.RelatedOpeningElement = voiding_feature
```

```
            ifc_file.add(voids_relation)

        ifc_file.remove(opening)

        if index == 1:
            current_timestamp = int(time.time())

            pset_cutout = ifc_file.create_entity("IfcPropertySet")
            pset_cutout.Name = "Pset_Cutout"
            pset_cutout.HasProperties = [
                ifc_file.create_entity("IfcPropertySingleValue", "Width", None, ifc_file.create_entity("IfcPositiveLengthMeasure", 80.0)),
                ifc_file.create_entity("IfcPropertySingleValue", "MeasuredWidth", None, ifc_file.create_entity("IfcPositiveLengthMeasure", 80.3)),
                ifc_file.create_entity("IfcPropertySingleValue", "WidthToleranceMin", None, ifc_file.create_entity("IfcLengthMeasure", 0.0)),
                ifc_file.create_entity("IfcPropertySingleValue", "WidthToleranceMax", None, ifc_file.create_entity("IfcLengthMeasure", 3.0)),
                ifc_file.create_entity("IfcPropertySingleValue", "Height", None, ifc_file.create_entity("IfcPositiveLengthMeasure", 80.0)),
                ifc_file.create_entity("IfcPropertySingleValue", "MeasuredHeight", None, ifc_file.create_entity("IfcPositiveLengthMeasure", 80.5)),
                ifc_file.create_entity("IfcPropertySingleValue", "HeightToleranceMin", None, ifc_file.create_entity("IfcLengthMeasure", 0.0)),
                ifc_file.create_entity("IfcPropertySingleValue", "HeightToleranceMax", None, ifc_file.create_entity("IfcLengthMeasure", 3.0)),
                ifc_file.create_entity("IfcPropertySingleValue", "DINReference", None, ifc_file.create_entity("IfcLabel", "DIN 1090-2")),
                ifc_file.create_entity("IfcPropertySingleValue", "Timestamp", None, ifc_file.create_entity("IfcTimeStamp", current_timestamp))
            ]

            pset_fabrication = ifc_file.create_entity("IfcPropertySet")
            pset_fabrication.Name = "Pset_FabricationProcess"
            pset_fabrication.HasProperties = [
                ifc_file.create_entity("IfcPropertySingleValue", "ProcessRequired", None, ifc_file.create_entity("IfcLabel", "plasma cutting")),
                ifc_file.create_entity("IfcPropertySingleValue", "ProcessForbidden", None, ifc_file.create_entity("IfcLabel", "")),
                ifc_file.create_entity("IfcPropertySingleValue", "Comment", None, ifc_file.create_entity("IfcLabel", "")),
                ifc_file.create_entity("IfcPropertySingleValue", "Timestamp", None, ifc_file.create_entity("IfcTimeStamp", current_timestamp))
            ]

            for pset in [pset_cutout, pset_fabrication]:
                rel_defines = ifc_file.create_entity("IfcRelDefinesByProperties")
                rel_defines.RelatingPropertyDefinition = pset
                rel_defines.RelatedObjects = [voiding_feature]
                ifc_file.add(rel_defines)


updated_file_path = r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_SampleBeam_updated.ifc"
ifc_file.write(updated_file_path)
```

LISTING E.2: ifcOpenShell python code for adding *IfcVoidingElement* to the fastener and adding additional property set

```
def add_voiding_feature_to_fasteners(ifc_file_path):

    ifc_file = ifcopenshell.open(ifc_file_path)
    fasteners = ifc_file.by_type("IfcMechanicalFastener")

    print(f"Anzahl gefundener IfcMechanicalFastener: {len(fasteners)}")

    if len(fasteners) < 2:
        raise ValueError("Not enough mechanical fasteners found in the IFC model. Need at least 2.")

    selected_fasteners = fasteners[:2]

    owner_history_list = ifc_file.by_type("IfcOwnerHistory")
    if owner_history_list:
        owner = owner_history_list[0]
    else:
        owner = ifc_file.create_entity("IfcOwnerHistory",
                                       OwningUser=None,
                                       OwningApplication=None,
                                       State=None,
                                       ChangeAction="ADDED",
                                       LastModifiedDate=None,
                                       CreationDate=0)

    def make_real_property(name, value):
        return ifc_file.create_entity("IfcPropertySingleValue",
                                      Name=name,
                                      NominalValue=ifc_file.create_entity("IfcReal", value),
                                      Unit=None)

    def make_label_property(name, value):
        return ifc_file.create_entity("IfcPropertySingleValue",
                                      Name=name,
                                      NominalValue=ifc_file.create_entity("IfcLabel", value),
                                      Unit=None)

    props = [
        make_real_property("Diameter", 10.0),
        make_real_property("MeasuredDiameter", 10.0),
        make_real_property("Diameter_tol_min", -0.05),
        make_real_property("Diameter_tol_max", 0.05),
        make_label_property("ReferenceNorm", "DIN 1090-2")
    ]

    pset = ifc_file.create_entity("IfcPropertySet",
                                  GlobalId=ifcopenshell.guid.new(),
                                  OwnerHistory=owner,
                                  Name="Pset_ThroughHole",
                                  HasProperties=list(props))

    for index, fastener in enumerate(selected_fasteners, start=1):
        placement = fastener.ObjectPlacement
```

```python

        void_name = f"Hole_{index:02}"

        voiding_feature = ifc_file.create_entity("IfcVoidingFeature",
                                                 GlobalId=ifcopenshell.guid.new(),
                                                 OwnerHistory=owner,
                                                 Name=void_name,
                                                 Description="Voiding feature for
    fastener",
                                                 PredefinedType="HOLE",
                                                 ObjectPlacement=placement)

        ifc_file.create_entity("IfcRelVoidsElement",
                               GlobalId=ifcopenshell.guid.new(),
                               OwnerHistory=owner,
                               Name=f"{void_name} Relation",
                               Description="Voiding feature applied to fastener",
                               RelatingBuildingElement=fastener,
                               RelatedOpeningElement=voiding_feature)

        ifc_file.create_entity("IfcRelDefinesByProperties",
                               GlobalId=ifcopenshell.guid.new(),
                               OwnerHistory=owner,
                               RelatedObjects=[voiding_feature],
                               RelatingPropertyDefinition=pset)

    output_path = ifc_file_path.replace("_updated.ifc", "_updated.ifc")
    ifc_file.write(output_path)
    print(f"Modified IFC file saved as: {output_path}")


if __name__ == "__main__":
    add_voiding_feature_to_fasteners(
        r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.
    ifc"
    )
```

LISTING E.3: ifcOpenShell python code for adding *IfcTask* and *IfcTaskTime* and a property set

```python
ifc = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample
      Beam_updated.ifc")

voiding_feature_guid = "3W2O2mdAL4dwQkU01hLgFg"

voiding_feature = next((entity for entity in ifc.by_type("IfcVoidingFeature") if
     entity.GlobalId == voiding_feature_guid), None)

if voiding_feature is None:
    print("IfcVoidingFeature not found.")
else:

    task = ifc.create_entity("IfcTask",
                             GlobalId=ifcopenshell.guid.new(),
                             Name="Plasma Cutting Task",
                             Status="NOTSTARTED")

    task_time = ifc.create_entity("IfcTaskTime",
                                  ScheduleStart="2025-03-27T18:30:00",
```

```
                                  ScheduleFinish="2025-03-27T18:35:00",
                                  DurationType="WORKTIME",
                                  ScheduleDuration="00:05:00",
                                  IsCritical=False,
                                  ActualStart="2025-03-27T18:40:00",
                                  ActualFinish="2025-03-27T18:43:00")

    task.TaskTime = task_time

    property_set = ifc.create_entity("IfcPropertySet",
                                     GlobalId=ifcopenshell.guid.new(),
                                     Name="Pset_PlasmaCuttingProcess")

    robot_command = ifc.create_entity("IfcPropertySingleValue", Name="RobotCommand")
    robot_speed = ifc.create_entity("IfcPropertySingleValue", Name="RobotSpeed")
    property_set.HasProperties = [robot_command, robot_speed]

    rel_defines = ifc.create_entity("IfcRelDefinesByProperties",
                                    GlobalId=ifcopenshell.guid.new(),
                                    RelatedObjects=[task],
                                    RelatingPropertyDefinition=property_set)

    task_relationship = ifc.create_entity("IfcRelAssignsToProcess",
                                          GlobalId=ifcopenshell.guid.new(),
                                          RelatingProcess=task,
                                          RelatedObjects=[voiding_feature])

    for entity in [task, task_time, property_set, robot_command, robot_speed, rel_defines, task_relationship]:
        ifc.add(entity)

    ifc.write(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.ifc")
```

LISTING E.4: ifcOpenShell python code updating the robot speed for the plasma cutting process

```

ifc = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.ifc")

for pset in ifc.by_type("IfcPropertySet"):
    if pset.Name == "Pset_PlasmaCuttingProcess":
        for prop in pset.HasProperties:
            if prop.Name == "RobotSpeed":
                prop.NominalValue = ifc.create_entity("IfcReal", 4.75)
                print("RobotSpeed updated successfully.")
                break
        else:
            print("RobotSpeed property not found in Pset_PlasmaCuttingProcess.")
        break
else:
    print("Pset_PlasmaCuttingProcess not found in the IFC file.")

ifc.write(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.ifc")
```

LISTING E.5: ifcOpenShell python code retrieving all relevant pro

```
ifc_file = ifcopenshell.open(r"C:\Users\VictoriaJung\Desktop\Python WIP\Diss\IFC4_Sample Beam_updated.ifc")

target_name = "Cutout_01"

voiding_feature = next((v for v in ifc_file.by_type("IfcVoidingFeature") if v.Name == target_name), None)

if not voiding_feature:
    print(f"Kein IfcVoidingFeature mit Name '{target_name}' gefunden.")
else:
    processes = []
    for rel in ifc_file.by_type("IfcRelAssignsToProcess"):
        if voiding_feature in rel.RelatedObjects:
            processes.append(rel.RelatingProcess)

    if processes:
        print(f"Gefundene Prozesse f r {target_name}:")
        for p in processes:
            print(f"- {p.Name} (Typ: {p.is_a()}, Status: {getattr(p, 'Status', '-')})")
    else:
        print(f"Keine Prozesse mit {target_name} verkn pft.")
```

Appendix F

SPARQL Queries

In addition to the listings within Chapter 7, this appendix provides the relevant SPARQL queries for uploading triplets to the triplestore.

LISTING F.1: SPARQL update beam width

```
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX schema: <http://schema.org>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX qudt-unit-1-1: http://qudt.org/1.1/vocab/unit#

INSERT {
    inst:IfcIShapeProfileDef_137 a ifc:IfcIShapeProfileDef.
    inst:IfcIShapeProfileDef_137 ifc:hasProperties_IfcProfileDef inst:Width.
    inst:Width a ifc:IfcProfileProperties.
    inst:Width dstv:hasPlannedValue inst:Width_IPE160_P.
    inst:Width dstv:hasMeasuredValue inst:Width_IPE160_M.

    inst:Width_IPE160_P a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Width_IPE160_P qudt-1-1:numericValue 82.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Width_IPE160_P prov:generatedAtTime ?timestamp.

    inst:Width_IPE160_M a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Width_IPE160_M qudt-1-1:numericValue 81.4"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Width_IPE160_M prov:generatedAtTime ?timestamp.

}
WHERE {
    BIND(NOW() AS ?timestamp)
}
```

LISTING F.2: SPARQL update beam height

```
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX schema: <http://schema.org>

INSERT {
    inst:IfcIShapeProfileDef_137 ifc:hasProperties_IfcProfileDef inst:Depth.
    inst:Depth a ifc:IfcProfileProperties.
    inst:Depth dstv:hasPlannedValue inst:Depth_IPE160_P.
    inst:Depth dstv:hasMeasuredValue inst:Depth_IPE160_M.

    inst:Depth_IPE160_P a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Depth_IPE160_P qudt-1-1:numericValue 160.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Depth_IPE160_P prov:generatedAtTime ?timestamp.

    inst:Depth_IPE160_M a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Depth_IPE160_M qudt-1-1:numericValue 162.1"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Depth_IPE160_M prov:generatedAtTime ?timestamp.

WHERE {
    BIND(NOW() AS ?timestamp)
}
```

LISTING F.3: SPARQL update beam length

```
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://http://www.w3.org/ns/prov#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX schema: <http://schema.org>

Insert{
    inst:IfcElementQuantity_175 ifc:hasProperties_IfcProfileDef inst:Length.
    inst:Length a ifc:IfcProfileProperties.
    inst:Length dstv:hasPlannedValue inst:Length_160_P.
    inst:Length dstv:hasMeasuredValue inst:Length_160_M.

    inst:Length_160_P a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Length_160_P qudt-1-1:numericValue 1005.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Length_160_P prov:generatedAtTime ?timestamp.

    inst:Length_160_M a dstv:PropertyState,  qudt-1-1:QuantityValue.
    inst:Length_160_M qudt-1-1:numericValue 1002.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Length_160_M prov:generatedAtTime ?timestamp.

WHere{
Bind(NOW() As ?timestamp)
    }
```

LISTING F.4: SPARQL update for linking resources to processes

```
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ioc: <http://w3id.org/ioc#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

INSERT {
  <http://baufest.org/dstv-test02#BeamFabrication_010_Plasma_292> a ioc:Process ;
    ioc:resource <http://baufest.org/dstv-test02#KR30-3_F> .

  <http://baufest.org/dstv-test02#KR30-3_F> a ioc:Resource ;
    rdfs:label "KR30-3 F" .
}
WHERE {}
```

LISTING F.5: SPARQL update cutout height tolerances

```
PREFIX ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#>
PREFIX inst: <https://baufest.org/inst#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX dstv: <http://w3id.org/dstv#>
PREFIX qudt-1-1: <http://qudt.org/schema/qudt/>
PREFIX qudt-unit-1-1: http://qudt.org/1.1/vocab/unit#
PREFI schema: <http://schema.org/>

INSERT {
    inst:WidthCutout_292  dstv:hasDefinedToleranceProperty
     inst:DefinedToleranceHeightCutout.
    inst:DefinedToleranceHeightCutout dstv:hasNormReference inst:NormHeightCutout.
    inst:DefinedToleranceHeightCutout dstv:hasToleranceLimit
     inst:ToleranceLimitHeightCutout.
    inst:ToleranceLimitHeightCutout dstv:hasUpperToleranceLimit
     inst:Tolerance_WidthCutout_U.
    inst:ToleranceLimitHeightCutout dstv:hasLowerToleranceLimit
     inst:Tolerance_WidthCutout_L.

    inst:Tolerance_WidthCutout_U a dstv:PropertyState, qudt-1-1:QuantityValue.
    inst:Tolerance_WidthCutout_U qudt-1-1:numericValue 3.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Tolerance_WidthCutout_U prov:generatedAtTime ?timestamp.

    inst:Tolerance_WidthCutout_L a dstv:PropertyState, qudt-1-1:QuantityValue.
    inst:Tolerance_WidthCutout_L qudt-1-1:numericValue "-0.0"^^xsd:double;
    qudt-1-1:unit qudt-unit-1-1:MilliM.
    inst:Tolerance_WidthCutout_L prov:generatedAtTime ?timestamp.

    inst:NormHeightCutout schema:Value "DIN EN1090-2".
}
WHERE {
    BIND(NOW() AS ?timestamp)
}
```